Coherence-Aware Transition-Intent Fusion for Co-Safe LTL Planning under Uncertain Semantic Maps

Junyue Huang **Bowen Ye** **Xiang Yin***
School of Automation and Intelligent Sensing
Shanghai Jiao Tong University, Shanghai 200240, China
hjy-564904993@sjtu.edu.cn, yebowen1025@sjtu.edu.cn, yinxiang@sjtu.edu.cn

Abstract

Robots operating in semantic environments often need to satisfy co-safe linear temporal logic (LTL) tasks within a finite execution budget before the semantic map is fully known. Many practical semantic-LTL planners maintain semantic beliefs but execute a single task policy induced by a point semantic interpretation, such as a maximum-a-posteriori (MAP) semantic map, possibly supported by risk checks or active perception. This premature commitment can be brittle when distant semantic observations are weak or correlated. We propose Coherence-Aware Transition-Intent Fusion, an online planner that preserves automaton-relevant semantic uncertainty using a finite top-K intent abstraction. Given a conservative current automaton estimate, the planner propagates plausible semantic assignments through the automaton, converts feasible successor transitions into weighted reach-avoid intents, and fuses their risk-gated Dijkstra progress scores. When competing intents create incoherent motion, a KL-regularized calibration step shifts weight toward intents consistent with recent physical progress. Experiments on random-obstacle and room-like semantic maps show improved satisfaction over single-intent and raw-fusion variants, and over a TOAPP* MAP-plus-active-perception baseline under the same static-view sensor. Ablations attribute the gain primarily to intent-level calibration.

1 Introduction

Linear temporal logic (LTL) is a formal language for specifying temporal constraints over sequences of semantic propositions. For example, a task such as "avoid red until reaching green, then eventually visit blue" constrains both which semantic labels may appear and in what order they may appear along the robot trajectory [Baier and Katoen, 2008, Ding et al., 2014, Lacerda et al., 2019]. In many semantic environments, however, the labels that determine task progress are only partially known. The robot observes nearby cells more reliably than distant cells, and far observations may be too weak to identify the correct proposition [Fu et al., 2016, Ghasemi et al., 2020, Kantaros et al., 2022]. This creates a mismatch between the symbolic automaton state used for planning and the semantic uncertainty maintained by the robot.

Semantic beliefs are used in prior work on task-oriented active perception and perception-based LTL planning [Ghasemi et al., 2020, Kantaros et al., 2022]. A common practical response is to maintain a semantic belief, compute a maximum-a-posteriori (MAP) semantic map, and replan in the product of the physical map and task automaton. The Task-Oriented Active Perception and Planning (TOAPP) framework further decides when to gather information before continuing [Ghasemi et al., 2020]. Such approaches are natural when repeated sensing can substantially reduce uncertainty, but errors caused by the same viewing geometry may persist across frames. We therefore study static-view semantic

*Corresponding author.

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

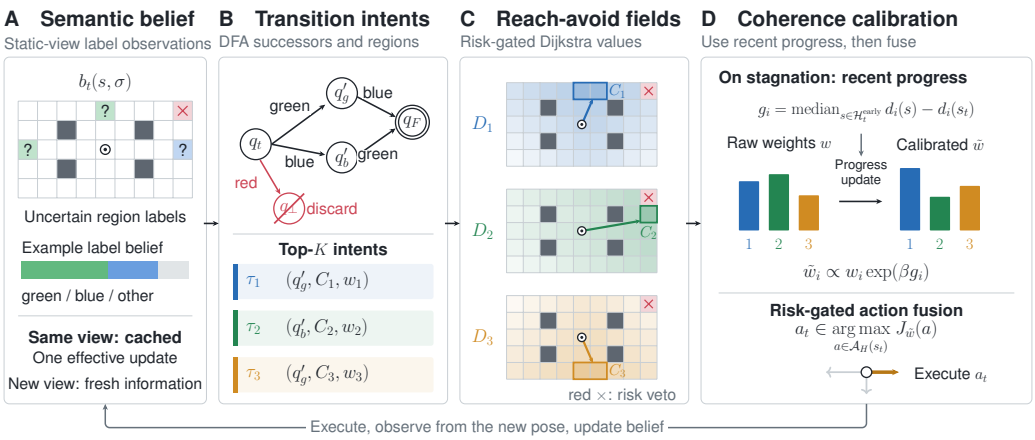


Figure 1: Schematic of Coherence-Aware Transition-Intent Fusion. (A) Static-view semantic belief. (B) Feasible DFA successors paired with connected regions define transition intents; different regions can share a successor. (C) Per-intent risk-gated reach-avoid values; arrows indicate target directions. (D) Recent physical progress calibrates intent weights during stagnation, followed by risk-gated action fusion. Execution supplies new observations and motion history.

observations: one effective observation per viewpoint-target pair, with fresh information available from new viewpoints. There, MAP collapse can commit the planner to a wrong semantic hypothesis long before the robot reaches the relevant region.

This paper studies a complementary design principle: avoid committing to a single semantic hypothesis or automaton transition unless the task structure forces it. At each step, the robot maintains a belief over semantic assignments and a conservative estimate of the current automaton state. For the current physical state and automaton state, we enumerate plausible semantic assignments, propagate them through the automaton, remove transitions that cannot still reach an accepting state, and transform the remaining alternatives into local reach-avoid intents. An action is selected by fusing the induced objectives, rather than by choosing one frontier state and discarding all others. Conceptually, this is a belief-to-task-intent abstraction: semantic posterior mass is retained only where it can change automaton progress, rather than being collapsed into a point semantic map or used only to decide whether to gather more evidence for a selected task policy.

Naive fusion is not enough. Multiple intents can point through different rooms or corridors, and their local action scores may cancel each other, causing loops or excessive probing. Our main contribution is therefore a coherence-aware online calibration layer. Given raw intent weights w_i and recent Dijkstra progress signals g_i, the calibrated weights solve

$$\tilde{w} = \arg\max_{u\in\Delta_K} \left[\sum_{i=1}^{K} u_i g_i - \frac{1}{\beta} D_{\mathrm{KL}}(u\|w) \right], \qquad \tilde{w}_i \propto w_i \exp(\beta g_i). \tag{1}$$

Here Δ_K is the K-simplex, $w = (w_1, \ldots, w_K)$, D_{KL} is Kullback–Leibler divergence, and $\beta > 0$ is a calibration gain distinct from the observation false-positive schedule $\beta(d)$. This keeps the planner within the original multi-intent distribution while shifting mass toward intents that are physically coherent with the recent trajectory.

Contributions. First, we formulate transition-intent fusion, where uncertain semantic assignments induce weighted automaton-transition-conditioned reach-avoid objectives. Second, we introduce coherence-aware intent calibration as a local KL-regularized optimization problem rather than a fallback mixture with a single-intent policy. Third, we evaluate on natural random semantic maps under four observation strengths and show improved satisfaction and reduced timeouts compared with single-intent and raw-fusion baselines. Figure 1 summarizes the resulting online planner. Appendix A provides the extended related-work comparison.

2 Problem Formulation

We follow the semantic-map planning setup of task-oriented active perception [Ghasemi et al., 2020], but instantiate a different temporal correlation model for observations. The goal is to satisfy a co-safe LTL task within a finite horizon in a map whose semantic labeling is not known in advance. The key distinction is that, in our main setting, repeatedly viewing the same target cell from the same robot state does not create independent evidence.

Agent model. The robot moves on a finite physical-state representation, modeled as a Markov decision process (MDP)

$$\mathcal{M} = (\mathcal{S}, s_0, \mathcal{A}, P, H), \tag{2}$$

where H is the episode horizon. For the main grid experiments, we use deterministic transitions to isolate semantic uncertainty. A history-dependent policy π maps past states, observations, and the maintained semantic belief to actions.

Semantic environment. Let AP be a finite set of semantic propositions, such as colors. The true semantic map is a deterministic but unknown labeling

$$L^\star : \mathcal{S} \to 2^{\mathsf{AP}}. \tag{3}$$

Let $\Sigma_{\text{cell}} \subseteq 2^{\mathsf{AP}}$ denote the admissible local assignments, which may contain multiple propositions. In the main grid benchmark, each cell contains at most one proposition, giving

$$\Sigma_{\text{cell}} = \{\emptyset\} \cup \{\{p\} : p \in \mathsf{AP}\}. \tag{4}$$

For the main analysis, we maintain local label beliefs using factorized semantic updates,

$$b_t(s, \sigma) \approx \Pr(L^\star(s) = \sigma \mid h_t), \qquad \sigma \in \Sigma_{\text{cell}}. \tag{5}$$

Here h_t denotes the action-observation history available to the planner. We initialize this belief from a neutral prior rather than an informative map prior. The planner constructs transition intents from these local label distributions and the task automaton.

Correlated distance-dependent observations. At time t, from robot state s_t, the sensor returns semantic observations of cells within a sensing radius. For a target cell s, proposition p, and distance $d = d(s_t, s)$, the marginal observation probabilities are

$$\Pr(Z_t(s_t, s, p) = 1 \mid p \in L^\star(s)) = \alpha(d), \tag{6}$$

$$\Pr(Z_t(s_t, s, p) = 1 \mid p \notin L^\star(s)) = \beta(d), \tag{7}$$

where $\alpha(d)$ decreases with distance and $\beta(d)$ increases toward the uninformative limit. Thus nearby observations are reliable, while far observations approach pure semantic noise.

The main experiments use a static-view correlated version of this model. For each viewpoint cell x and target cell s, a latent vector view

$$Y_{x,s} = (Y_{x,s,p})_{p \in \mathsf{AP}} \sim O(x, s, L^\star(s)) \tag{8}$$

is sampled once per episode, where O denotes the semantic observation kernel induced by $\alpha(d)$ and $\beta(d)$. The cached view contributes at most one informative Bayesian update; repeated requests to the same viewpoint-target pair provide no new likelihood evidence. Therefore standing still and observing the same cells repeatedly does not reduce uncertainty by repeated independent likelihood multiplication. This is the main modeling difference from TOAPP-like repeated-view active perception: active sensing can still help by moving to new viewpoints, but it cannot wash out uncertainty through repeated measurements from the same view.

Static-view sensing in practice. A view is the semantic observation of a target region from a fixed robot pose. In a realistic perception system, this observation may already combine a short sequence of frames from that pose. Averaging or fusing these frames can reduce short-term sensor noise, while the remaining semantic error may still be tied to the pose and target geometry: the target may be partly hidden, too small in the image, seen from an unclear angle, or consistently confused by the segmentation model. More frames from the same pose do not change the viewpoint or necessarily resolve these errors. Our task planner uses this semantic-level information, and the cached view represents one effective observation per pose-target pair. Moving to a new viewpoint can provide fresh semantic information; Table 12 evaluates the complementary iid-view setting.

Temporal-logic task and automaton. A robot trajectory induces a word

$$\sigma_0\sigma_1\cdots, \qquad \sigma_t = L^\star(s_t) \in 2^{\mathsf{AP}}, \tag{9}$$

and the LTL formula is evaluated on this sequence of proposition valuations. We use standard temporal operators such as until U and eventually F; for example, $\neg\mathrm{red}\ \mathsf{U}\ \mathrm{green}$ means that red must be avoided until green is reached. We consider a co-safe LTL formula φ over AP, whose satisfaction can be witnessed by a finite good prefix. We represent its accepting-prefix behavior by a deterministic finite automaton (DFA)

$$\mathcal{A}_\varphi = (Q, q_0, \Sigma, \delta, F), \qquad \Sigma = 2^{\mathsf{AP}}. \tag{10}$$

Here Q is the task-state set, $\delta : Q \times \Sigma \to Q$ is the deterministic transition function, and $F \subseteq Q$ contains accepting states. Reaching F recognizes a satisfying prefix, after which the episode terminates. We restrict transitions to feasible local assignments and complete a rejecting sink; Q_{bad} comprises states from which F is unreachable under this alphabet. The evaluated formulas include reach-avoid tasks, sequential reachability, and finite conjunctions of these objectives. An episode succeeds if the true labeled trajectory reaches an accepting prefix within H actions, violates the task if it reaches Q_{bad}, and times out otherwise. The finite horizon is an execution budget in addition to the co-safe restriction; it is not a definition of that logic fragment. Recurrent acceptance is not part of the evaluated objective.

Planning objective. The ideal objective is accepting-prefix reachability within the execution budget. Let $\xi^\pi(L^\star, O)$ denote the labeled trajectory generated by policy π under the true semantic map and observation process. Let τ_F be the number of executed actions at its first accepting prefix, or infinity if no such prefix occurs. The planning problem is

$$\pi^\star \in \arg\max_\pi \Pr_{\pi, L^\star, O}\left[\tau_F \le H\right]. \tag{11}$$

Solving this Bayes-adaptive problem exactly would require planning over the joint physical state, automaton state, and high-dimensional semantic belief, as in general partially observable Markov decision process (POMDP) planning [Kaelbling et al., 1998, Ross et al., 2008, Barenboim and Indelman, 2023]. The question we address is how to construct a task-oriented online planner that preserves the automaton-relevant parts of semantic uncertainty without collapsing immediately to a single MAP semantic map.

3 Coherence-Aware Transition-Intent Fusion

Our approach replaces the usual

$$b_t \longrightarrow \widehat{L}_{\mathrm{MAP}} \longrightarrow \text{product-MDP policy} \tag{12}$$

interface by a belief-to-automaton-intent interface:

$$b_t,\ q_t \longrightarrow \{(q_i', \mathcal{R}_i, \mathcal{U}_i, w_i)\}_{i=1}^K \longrightarrow \text{risk-gated fused action.} \tag{13}$$

This keeps multiple task-relevant semantic hypotheses alive at the automaton-transition level, while still producing a tractable local control problem.

Top-K surrogate objective. Equation (11) is a Bayes-adaptive satisfaction objective over the full semantic-map posterior. Instead of solving this in the exponentially large belief space, we construct at each time a compact posterior surrogate

$$\mathcal{I}_t^K = \{(q_i', C_i, w_i, D_i)\}_{i=1}^K, \tag{14}$$

where q_i' is a feasible DFA successor, C_i is a connected spatial component that can trigger it, w_i is its posterior intent weight, and D_i is a safe reachability value for that component. The action layer then optimizes

$$a_t \in \arg\max_{a \in \mathcal{A}_H(s_t)} \left[\sum_{i=1}^K \tilde{w}_i Q_i(s_t, a) - \lambda_{\mathrm{safe}} C_{\mathrm{safe}}(a)\right], \tag{15}$$

where $\mathcal{A}_H(s_t)$ removes hard-risk actions, Q_i is the local progress score for intent i, $\tilde{w}_i$ is either the raw posterior intent weight or the coherence-calibrated weight, $C_{\mathrm{safe}}(a)$ summarizes residual safety

Algorithm 1 Coherence-Aware Transition-Intent Fusion

Require: Accepting-prefix DFA $\mathcal{A}_\varphi$, grid MDP $\mathcal{M}$, horizon H, intent budget K
Initialize neutral semantic belief b_0, physical state s_0, and automaton estimate q_0
for $t = 0, 1, \ldots, H-1$ **do**
  Update b_t from static-view observations and update conservative q_t from current-cell evidence
  Compute feasible successor mass $\rho_t(q' \mid s, q_t)$; remove rejecting or accepting-unreachable successors
  Split surviving high-posterior successor cells into connected components C_i
  Form intents $\tau_i = (q'_i, C_i, w_i)$; keep the top K by Eq. (19)
  For each τ_i, compute hard risk H_t, soft risk $S_{t,i}$, and safe Dijkstra value D_i
  if intent entropy, revisit rate, and recent progress indicate incoherent fusion **then**
    Calibrate intent weights by $\tilde{w}_i \propto w_i \exp(\beta g_i)$
  else
    Set $\tilde{w}_i \leftarrow w_i$
  end if
  Select a_t by Eq. (15); execute it and observe the next state
  if the true product reaches acceptance, rejection, or timeout **then**
    return episode status
  end if
end for
return timeout

and motion costs expanded in Eq. (25), and $\lambda_{\text{safe}} \geq 0$ is its aggregate weight. This is the optimization object solved by our online planner: a posterior-weighted local approximation to the temporal-logic satisfaction objective.

The loop never selects a MAP semantic map and never mixes the final action with a single-intent policy; all online corrections act inside the transition-intent abstraction.

3.1 From semantic belief to transition intents

Let s_t be the current physical state and q_t the conservative automaton estimate. For any candidate cell s, the belief induces a distribution over assignments $\sigma \in \Sigma_{\text{cell}}$. For each plausible assignment, the DFA successor is

$$q' = \delta(q_t, \sigma). \tag{16}$$

Equivalently, the belief induces an automaton-successor distribution

$$\rho_t(q' \mid s, q_t) = \sum_{\sigma \in \Sigma_{\text{cell}}} b_t(s, \sigma)\, \mathbf{1}\{\delta(q_t, \sigma) = q'\}. \tag{17}$$

We discard successors in Q_{bad}, which cannot reach an accepting state in the restricted automaton. Each surviving successor q' induces one or more local reach-avoid intents:

$$\tau_i = (q'_i, \mathcal{R}_i, \mathcal{U}_i, w_i). \tag{18}$$

Here $\mathcal{R}_i$ is a set of cells whose semantic belief can progress the automaton toward q'_i or beyond, $\mathcal{U}_i$ is a set of cells that would make the current transition path unsafe, and w_i combines semantic probability mass, distance-to-acceptance in the automaton, and physical distance. If the same automaton successor corresponds to multiple disconnected semantic regions, it creates multiple components rather than one merged target. This matters in room maps: two green regions can be separated by walls, and treating them as one connected reach set can create invalid local policies.

More explicitly, for an intent component C_i associated with successor q'_i, the unnormalized weight is

$$\bar{w}_i = m_i\, \kappa_i\, r_i, \qquad m_i = \sum_{s \in C_i} \rho_t(q'_i \mid s, q_t), \tag{19}$$

where m_i is posterior transition mass. The automaton term

$$\kappa_i = \exp\left(-d_A(q'_i)/T_I\right) \tag{20}$$

uses the shortest automaton-edge distance $d_A(q'_i)$ from q'_i to an accepting state in the restricted DFA, with intent-temperature $T_I > 0$; unreachable successors have already been filtered out. It prioritizes successors closer to acceptance, and the spatial discount

$$r_i = \frac{1}{1 + \eta d_M(s_t, C_i)} \tag{21}$$

penalizes physically distant components using the obstacle-aware graph distance d_M, with spatial scale $\eta > 0$. The planner keeps the top K components by $\bar{w}_i$ and normalizes them. Single-intent planning is therefore the special case $K = 1$, while Fusion uses $K > 1$ and delays posterior collapse to a single downstream intent.

3.2 Fused risk-gated action selection

For each action $a \in \mathcal{A}$, a risk-gated Dijkstra-style rollout estimates how the next state supports the active intents. We distinguish hard risk from intent-specific soft risk. Hard risk is the posterior probability that entering a cell immediately violates the co-safe task or makes acceptance unreachable:

$$H_t(s) = \sum_{\sigma:\delta(q_t,\sigma)\in Q_{\mathrm{bad}}} b_t(s,\sigma). \tag{22}$$

Here Q_{bad} is the set of rejecting or acceptance-unreachable states in the restricted DFA. Actions entering cells with $H_t(s') > \theta_H$ are removed from $\mathcal{A}_H(s_t)$. Soft risk is local to an intent: a cell may not globally violate the task, but may invalidate the progression route represented by intent i. We denote this risk by $S_{t,i}(s)$ and first treat it as a strict constraint; if no strict route exists, it becomes a penalty in a relaxed route.

For intent i, the strict safe graph contains cells $X_i^{\mathrm{strict}} = \{s : H_t(s) \le \theta_H,\ S_{t,i}(s) \le \theta_S\}$. Reverse Dijkstra computes

$$D_i^{\mathrm{strict}}(s) = \min_{\zeta: s\to C_i,\ \zeta\subseteq X_i^{\mathrm{strict}}} |\zeta|. \tag{23}$$

If all strict routes are infeasible, we keep hard-risk filtering and relax soft risk:

$$D_i^{\mathrm{relax}}(s) = \min_{\zeta: s\to C_i} \sum_{u\in\zeta} \left(1 + \lambda_S S_{t,i}(u)\right), \qquad H_t(u) \le \theta_H. \tag{24}$$

Thus each intent internally solves a constrained shortest-path problem before the multi-intent fusion step. We use "safe" below only for this belief-induced, thresholded graph; it is not a formal guarantee of zero violations under the true hidden semantic map. The raw action score has the schematic form

$$\begin{aligned} J(a) = \sum_{i=1}^{K} w_i \left[\lambda_P \Delta D_i(a) + \lambda_R \mathbf{1}\{s' \in C_i\}\right] \\ - \lambda_L L(a) - C_{\mathrm{aux}}(a) - \lambda_{\mathrm{stay}} \mathbf{1}\{a = \mathrm{stay}\}. \end{aligned} \tag{25}$$

Here s' is the next physical cell, $\Delta D_i(a)$ is the safe-Dijkstra distance decrease toward intent i, $L(a) = \sum_{i:D_i(s')=\infty} w_i$ is the posterior mass of intents lost by the action, and $C_{\mathrm{aux}}(a) = \lambda_{\mathrm{cal}} R_{\mathrm{cal}}(s') + \lambda_{\mathrm{loop}} R_{\mathrm{loop}}(a) + \lambda_{\mathrm{visit}} R_{\mathrm{visit}}(s')$ contains only nonnegative residual penalties from calibrated local risk, recent-loop memory, and state-revisit memory. Action-independent step constants are omitted because they do not affect the one-step action ranking. Hard risk H_t is a lexicographic veto: actions entering cells with $H_t(s') > \theta_H$ are removed before scoring. Soft-bad regions, which only invalidate some hypotheses or some future sequences, are first treated as hard constraints inside D_i when a safe route exists and are relaxed only when no such route is available. If no intent has a reachable safe-Dijkstra route, the planner falls back to probe states, which are candidate viewpoints expected to make the current intent set more actionable under the belief. This differs from MAP replanning: the planner does not first select one semantic map or one automaton frontier, and therefore does not erase alternatives that remain task feasible.

A two-intent example. Suppose an accepting transition can be realized by either of two uncertain spatial components C_1, C_2, producing intents (q_1', C_1, w_1, D_1) and (q_2', C_2, w_2, D_2) with weights $0.6, 0.4$. The successors may be the same task state even though the spatial routes differ. To illustrate Eq. (25), consider two non-stay actions with no reach bonus or lost intent and equal auxiliary costs. If action A has distance-progress values $(1, -0.5)$, its relevant score is $0.6(1) + 0.4(-0.5) = 0.4$; if action B has values $(0.5, 0.5)$, its score is 0.5. Fusion therefore prefers B, which makes progress toward both alternatives, although A is better for the highest-weight intent alone. These are illustrative weighted-distance changes, not measured episode results. If incompatible alternatives subsequently cause repeated visits with little progress, the calibration below uses their recent progress to adjust the same two weights.

3.3 Coherence-aware intent calibration

Raw fusion may still fail when several plausible intents pull the action selector in incompatible directions. We track recent physical progress for each intent. Let $d_i(s)$ be the graph distance from state s to the current intent component under the safe Dijkstra metric. For a recent window of states $(s_{t-m}, \ldots, s_t)$, let $\mathcal{H}_t^{\text{early}}$ be the first half of that window and define

$$g_i = \text{median}_{s \in \mathcal{H}_t^{\text{early}}} d_i(s) - d_i(s_t). \tag{26}$$

Positive g_i means the current state is closer to intent i than the recent reference position. When the planner detects high intent entropy, repeated revisits, and little weighted progress, it replaces w by the KL-regularized optimizer in Eq. (1). The KL calibration itself is an intent-level update: it does not mix the final policy with a single-intent policy and does not use the true map. The full implementation also includes a small action-level coherence penalty, ablated in Appendix C; empirically the KL-only variant nearly matches the full method.

Per-intent optimality. For fixed belief b_t, automaton estimate q_t, risk thresholds, and intent component C_i, Eq. (23) is an exact shortest-path problem on the belief-induced safe graph. Reverse Dijkstra therefore returns the optimal safe reach value for intent i under the current abstraction. The novelty of Fusion is not replacing shortest-path planning inside an intent; it is keeping several posterior transition intents alive and optimizing the action layer over their weighted progress values.

No-early-commit condition. Single-intent planning is equivalent to setting $K = 1$ and retaining only the largest raw intent weight. Let a_S be the action chosen by this premature-commit planner, and let

$$J_K(a) = \sum_{i=1}^{K} w_i Q_i(s_t, a) - \lambda_{\text{safe}} C_{\text{safe}}(a) \tag{27}$$

be the top-K surrogate before coherence calibration. If there exists an action a such that

$$J_K(a) > J_K(a_S), \tag{28}$$

then Fusion strictly improves the local top-K surrogate over Single at that state. This condition captures the intended regime: posterior mass is spread over several plausible transitions, and one action can make safe progress for more than the top-1 hypothesis. Let $p_i = \bar{w}_i / \sum_j \bar{w}_j$ be the normalized weights over all intent components, ordered by weight, and let $\epsilon_K = \sum_{i>K} p_i < 1$. After top-$K$ truncation, the retained weights are $w_i = p_i/(1 - \epsilon_K)$ for $i \le K$. With fixed per-intent scores satisfying $|Q_i(s, a)| \le Q_{\max}$, the truncation error is bounded by

$$|J_{\text{all}}(a) - J_K(a)| \le 2 Q_{\max} \epsilon_K. \tag{29}$$

Here $J_{\text{all}}(a) = \sum_i p_i Q_i(s_t, a) - \lambda_{\text{safe}} C_{\text{safe}}(a)$ uses all intents and the same safety term as J_K. The bound accounts for both discarded mass and renormalization of the retained weights. Thus the top-K abstraction is accurate when the retained intents cover most posterior mass; increasing K indefinitely is not necessarily useful because low-mass intents can reintroduce local interference.

Calibration optimality. For fixed progress estimates g_i and raw weights $w_i > 0$, Eq. (1) is strictly concave over the simplex. The calibrated weights are the unique optimizer and satisfy

$$\log \frac{\tilde{w}_i}{w_i} - \log \frac{\tilde{w}_j}{w_j} = \beta(g_i - g_j). \tag{30}$$

Thus coherence calibration monotonically increases the relative weight of intents with higher recent physical progress, while remaining KL-close to the original belief-induced weights. This gives the online layer a precise optimization interpretation: it is a bounded local correction to multi-intent fusion, not an arbitrary rule for escaping loops. In the implementation, a small floor on $\tilde{w}_i$ prevents the update from collapsing into a hard MAP commitment.

These theoretical statements concern local surrogate optimization and shortest paths on a fixed belief-induced graph. They are not guarantees of global task satisfaction or POMDP optimality. The closed-form KL optimizer describes the update before the implementation's numerical weight-floor adjustment.

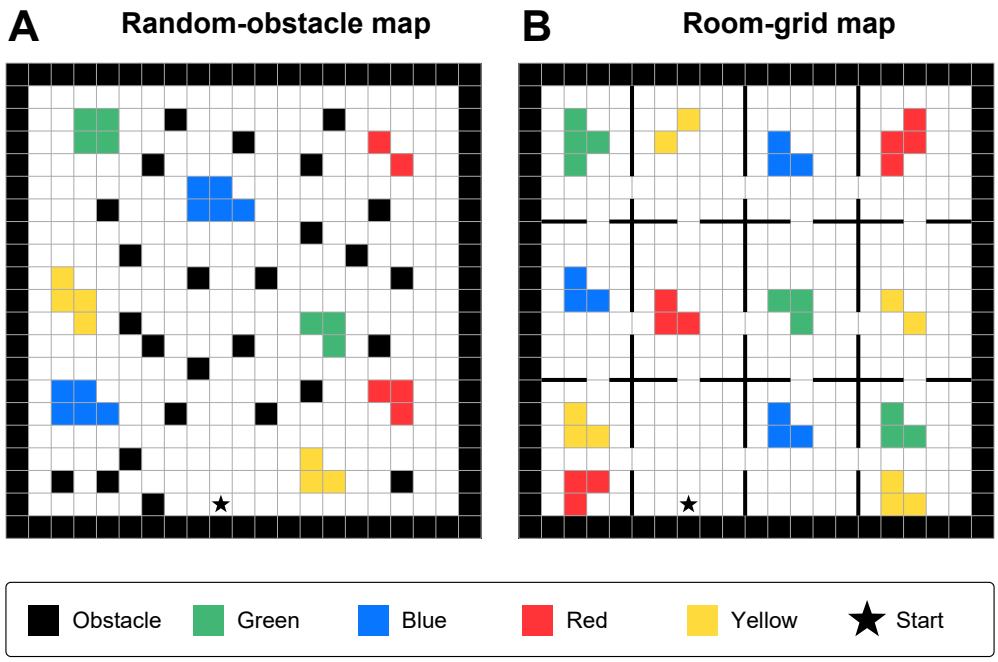


Figure 2: Representative benchmark map families matching the generators used in the experiments. Random-obstacle maps test unstructured semantic navigation, while room-grid maps add topological bottlenecks where premature commitment and incoherent multi-intent fusion are more likely to cause timeouts. Each cell has at most one proposition in the main benchmark.

4 Experiments

Maps and tasks. We evaluate on two map families. The random-obstacle family has unstructured obstacle fields and semantic clusters. The random-room family contains room-like topology and narrow connections, creating harder long-range exploration. Each cell has at most one proposition. We use three LTL task families: reach-avoid formulas such as $\neg \mathrm{red}\ \mathsf{U}\ \mathrm{green}$, mixed formulas that combine reach-avoid and eventual visits, and more complex sequential formulas with multiple ordered goals. Figure 2 illustrates the two map families used in the experiments.

Experimental protocol. Unless otherwise stated, all methods use the same horizon $H = 160$, sensing radius 10, neutral semantic prior 0.06 per proposition, and static-view correlated observations. The currently occupied cell is treated as a contact-level semantic observation shared by all planners; other visible cells follow the distance-dependent observer in Section 2. Table 1 gives the four observer profiles as endpoint tuples $(\alpha_{\mathrm{near}}, \alpha_{\mathrm{far}}, \beta_{\mathrm{near}}, \beta_{\mathrm{far}})$ for true-positive and false-positive schedules. "Near" and "far" are the endpoint parameters of the distance schedule; the contact-level current-cell reading overrides this schedule only at the robot's occupied cell. When reported, success steps are averaged only over successful episodes. This conditional mean can increase when a planner rescues difficult episodes that otherwise time out; it should be interpreted together with satisfaction and timeout rates, not as an isolated efficiency ranking across different success subsets.

Table 1: Observer profiles. TP/FP are true/false positives.

Profile	near TP	far TP	near FP	far FP
strong	0.990	0.620	0.005	0.380
base	0.970	0.550	0.010	0.450
weak	0.950	0.520	0.020	0.480
very weak	0.920	0.505	0.040	0.495

Methods. The main internal comparisons are *Single* ($K = 1$), *Raw Fusion* ($K = 4$ without coherence calibration), and *CoherenceAware Fusion* ($K = 4$). For external context, we compare

Table 2: Large-split internal comparison. S, V, and T denote success, violation, and timeout rates.

Observer	Method	N	S	V	T
strong	Single ($K = 1$)	437	0.943	**0.005**	0.053
strong	Raw Fusion ($K = 4$)	437	0.945	0.007	0.048
strong	CoherenceAware ($K = 4$)	437	**0.966**	0.011	**0.023**
base	Single ($K = 1$)	437	0.927	0.018	0.055
base	Raw Fusion ($K = 4$)	437	0.954	**0.002**	0.043
base	CoherenceAware ($K = 4$)	437	**0.975**	0.007	**0.018**
weak	Single ($K = 1$)	437	0.918	0.018	0.064
weak	Raw Fusion ($K = 4$)	437	0.927	**0.009**	0.064
weak	CoherenceAware ($K = 4$)	437	**0.957**	0.011	**0.032**
very weak	Single ($K = 1$)	437	0.908	0.016	0.076
very weak	Raw Fusion ($K = 4$)	437	0.913	0.007	0.080
very weak	CoherenceAware ($K = 4$)	437	**0.938**	**0.002**	**0.059**

against TOAPP*, an adaptation of the TOAPP MAP policy-validation and active-perception pipeline [Ghasemi et al., 2020]. Plain MAP-replan is reported in the appendix as a diagnostic lower bound because it collapses the neutral semantic prior to a point map and times out throughout the main static-view split.

Large split. Table 2 reports the completed large split over seeds 24–39. Each observer-method pair contains 437 feasible episodes after removing oracle-infeasible instances, meaning true-map instances with no satisfying path within the horizon; Appendix Table 7 gives task-family stratification.

The calibrated planner improves success over both internal variants across all observation strengths, mainly by reducing timeouts under weak observers. The method improves the success-timeout-violation tradeoff rather than guaranteeing zero violations: Raw Fusion has slightly lower violation rates in some profiles, while CoherenceAware mainly increases satisfaction. A medium risk-gate sweep confirms that this gain is not only a looser safety filter: at the default $\theta_H = 0.075$, CoherenceAware fixes 6/0 base-observer and 9/0 weak-observer Raw Fusion failures in matched pairs (Appendix Table 10). Table 3 shows that the improvement is not driven by one formula family; the gap is largest on mixed and complex LTL tasks, where premature commitment and multi-intent interference are more common. On matched instances aggregated over the four observers, CoherenceAware fixes 98 Single failures while breaking 37, and fixes 57 Raw Fusion failures while breaking 15; Appendix Table 6 reports the corresponding sign tests.

Table 3: Success by LTL task family.

Task	Single	Raw	Coherence
reach-avoid	0.948	0.953	**0.957**
mixed	0.905	0.941	**0.977**
complex LTL	0.917	0.926	**0.956**
Macro avg.	0.923	0.940	**0.963**

External baselines. Table 4 compares the base-observer full method against TOAPP*, Single, and Raw Fusion on the same seeds and 437 feasible instances. TOAPP* tests the MAP-plus-active-perception design principle under the same sensor available to all planners, not a line-by-line reproduction of TOAPP. Under the main static-view sensor, TOAPP* obtains $S/V/T = 0.451/0.014/0.535$. Under a TOAPP-like iid repeated-view observer on the same instances, it improves to $S/V/T = 0.865/0.053/0.082$, supporting that the main-table gap is tied to the observation regime rather than to a failed active-perception implementation; Appendix Table 12 reports the corresponding comparison for all four methods under both sensors. Appendix Table 13 also reports a stronger belief-style sanity check on a tractable medium subset.

The asterisk indicates that the MAP task policy, belief-risk gating, and bounded task-oriented active-perception structure follow TOAPP, while the observation update is replaced by the static-view correlated sensor used by all methods in our main experiments.

Table 4: Base-observer main baseline comparison under static-view correlated sensing.

Method	N	S	V	T	Success steps
TOAPP*	437	0.451	0.014	0.535	35.70
Single ($K = 1$)	437	0.927	0.018	0.055	57.48
Raw Fusion ($K = 4$)	437	0.954	**0.002**	0.043	58.97
CoherenceAware Fusion ($K = 4$)	437	**0.975**	0.007	**0.018**	57.15

Ablation. Appendix Table 9 isolates the coherence layer inside the fusion family. *MemoryAware* keeps safe routing and loop-aware physical memory but disables coherence calibration. *Action-only* removes KL intent reweighting, whereas *KL-only* removes the action-level coherence penalty. These variants separate calibrated intent weights from generic loop-escape behavior: on the base observer, MemoryAware and Action-only obtain $S/T = 0.966/0.027$, while KL-only and Full both obtain $0.975/0.018$. Across all observers, Action-only remains close to the memory-aware control, whereas KL-only matches the full method, indicating that the main mechanism is intent-simplex calibration rather than an action-level coherence heuristic. Appendix C reports additional no-physical-memory and no-escape sensitivity checks.

When calibration helps. Calibration helps when similarly weighted intents favor incompatible moves and cause revisits; Raw Fusion may suffice when moves agree or one intent dominates. Across observer profiles in Table 2, calibration gains are comparable to, and often larger than, the initial fusion gains.

Additional robustness and scale checks. Appendix F adds trigger sweeps, an extra seed block, stochastic execution, larger semantic alphabets and automata, and full-split belief-space comparisons. Appendices G and H examine multi-proposition labels and non-grid roadmap planning.

Discussion and limitations. For robot deployment, mapping and localization would supply regions, connectivity, and robot pose; semantic observations projected onto regions would update label beliefs. Our planner would select a region, with motion execution handled by a planner and feedback controller. Missing regions or biased observations can impair this connection, which still needs robot-system validation. Factorized beliefs and a finite intent budget are approximations; coherence can also favor an incorrect hypothesis yielding short-term progress (Appendix B). Recurrent objectives require a different long-term formulation beyond the finite-prefix setting studied here.

5 Conclusion

We presented Coherence-Aware Transition-Intent Fusion for co-safe LTL planning under uncertain semantic maps. The method keeps multiple plausible automaton transitions alive, turns them into local reach-avoid intents, and calibrates their weights online using a KL-regularized physical-coherence objective. Across random-obstacle and random-room maps, calibrated fusion improves satisfaction over single-intent, raw-fusion, and TOAPP* baselines.

Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China (62573291, 62533017). The authors declare no competing interests.

References

S. Arora, E. Hazan, and S. Kale. The multiplicative weights update method: a meta-algorithm and applications. *Theory Comput.*, 8:121–164, 2012.

C. Baier and J.-P. Katoen. *Principles of Model Checking (Representation and Mind Series)*. The MIT Press, 2008. ISBN 026202649X.

R. Bajcsy, Y. Aloimonos, and J. K. Tsotsos. Revisiting active perception. *Autonomous Robots*, 42: 177 – 196, 2016.

M. Barenboim and V. Indelman. Online pomdp planning with anytime deterministic guarantees. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 79886–79902. Curran Associates, Inc., 2023.

A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003. ISSN 0167-6377.

K. Chatterjee, M. Chmelík, and M. Tracol. What is decidable about partially observable markov decision processes with omega-regular objectives. In *Annual Conference for Computer Science Logic*, 2013.

X. Ding, S. L. Smith, C. Belta, and D. Rus. Optimal control of markov decision processes with linear temporal logic constraints. *IEEE Transactions on Automatic Control*, 59(5):1244–1257, 2014.

M. A. Do Carmo Alves, A. Varma, Y. Elkhatib, and L. Soriano Marcolino. Information-guided planning: An online approach for partially observable problems. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 69157–69177. Curran Associates, Inc., 2023.

J. Fu and U. Topcu. Probably approximately correct mdp learning and control with temporal logic constraints. In *Proceedings of Robotics: Science and Systems*, Berkeley, USA, July 2014.

J. Fu, N. Atanasov, U. Topcu, and G. J. Pappas. Optimal temporal logic planning in probabilistic semantic maps. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3690–3697, 2016.

M. Ghasemi, E. Bulgur, and U. Topcu. Task-oriented active perception and planning in environments with partially known semantics. In H. D. III and A. Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 3484–3493. PMLR, 13–18 Jul 2020.

E. M. Hahn, M. Perez, S. Schewe, F. Somenzi, A. Trivedi, and D. Wojtczak. Omega-regular decision processes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(19):21125–21133, Mar. 2024.

Q. H. Ho, T. J. Becker, B. Kraske, Z. Laouar, M. S. Feather, F. Rossi, M. Lahijanian, and Z. Sunberg. Recursively-constrained partially observable markov decision processes. In N. Kiyavash and J. M. Mooij, editors, *Uncertainty in Artificial Intelligence, 15-19 July 2024, Universitat Pompeu Fabra, Barcelona, Spain*, volume 244 of *Proceedings of Machine Learning Research*, pages 1658–1680. PMLR, 2024a.

Q. H. Ho, M. S. Feather, F. Rossi, Z. N. Sunberg, and M. Lahijanian. Sound heuristic search value iteration for undiscounted POMDPs with reachability objectives. In *The 40th Conference on Uncertainty in Artificial Intelligence*, 2024b.

M. Jackermeier and A. Abate. DeepLTL: Learning to efficiently satisfy complex LTL specifications for multi-task RL. In *The Thirteenth International Conference on Learning Representations*, 2025.

L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1):99–134, 1998. ISSN 0004-3702.

S. Kalluraya, G. J. Pappas, and Y. Kantaros. Multi-robot mission planning in dynamic semantic environments. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1630–1637, 2023.

Y. Kantaros, S. Kalluraya, Q. Jin, and G. J. Pappas. Perception-based temporal logic planning in uncertain semantic maps. *IEEE Transactions on Robotics*, 38(4):2536–2556, 2022.

B. Lacerda, F. Faruq, D. Parker, and N. Hawes. Probabilistic planning with formal performance guarantees for mobile service robots. *The International Journal of Robotics Research*, 38(9): 1098–1123, 2019.

M. Lahijanian, S. B. Andersson, and C. Belta. Temporal logic motion planning and control with probabilistic satisfaction guarantees. *IEEE Transactions on Robotics*, 28(2):396–409, 2012.

X.-B. Le, D. Wagner, L. Witzman, A. Rabinovich, and L. Ong. Reinforcement learning with LTL and ω-regular objectives via optimality-preserving translation to average rewards. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS '24, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9798331314385.

A. Linard, A. Gautier, D. Duberg, and J. Tumova. Robust mitl planning under uncertain navigation times. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2498–2504, 2024.

R. J. Moss, A. Jamgochian, J. Fischer, A. Corso, and M. J. Kochenderfer. Constrainedzero: Chance-constrained pomdp planning using learned probabilistic failure surrogates and adaptive safety constraints. In K. Larson, editor, *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pages 6752–6760. International Joint Conferences on Artificial Intelligence Organization, 8 2024. Main Track.

M. Perez, F. Somenzi, and A. Trivedi. A PAC learning algorithm for LTL and Omega-Regular objectives in MDPs. In *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9.

S. Ross, J. Pineau, and S. Paquet. Online planning algorithms for pomdps. *Journal of Artificial Intelligence Research*, 32:663–704, 07 2008.

M. T. J. Spaan, T. Veiga, and P. U. Lima. Decision-theoretic planning under uncertainty with information rewards for active cooperative perception. *Autonomous Agents and Multi-Agent Systems*, 29:1157 – 1185, 2014.

M. Vahs, C. Pek, and J. Tumova. Risk-aware spatio-temporal logic planning in gaussian belief spaces. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7879–7885, 2023.

B. Ye, J. Huang, Y. Liu, X. Qiao, and X. Yin. Bridging perception and planning: Towards end-to-end planning for signal temporal logic tasks. In *2026 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5054–5061. IEEE, 2026. doi: 10.1109/ICRA57385.2026.11696821.

P. Yu, S. Dong, S. Sheng, L. Feng, and M. Kwiatkowska. Trust-aware motion planning for human-robot collaboration under distribution temporal logic specifications. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 12949–12955, 2024.

A Related Work

Temporal-logic planning under semantic and model uncertainty. LTL and automata-based planning have been widely used for robotics and control by forming a product between a physical model and a task automaton and optimizing satisfaction or progress [Baier and Katoen, 2008, Lahijanian et al., 2012, Ding et al., 2014, Lacerda et al., 2019]. Related work has addressed unknown transition models [Fu and Topcu, 2014], probabilistic semantic maps [Fu et al., 2016], and mission planning in dynamic semantic environments [Kalluraya et al., 2023]. Work connecting perception and temporal-logic planning ranges from LTL planning with uncertain semantic maps [Kantaros et al., 2022] to end-to-end trajectory generation from multi-view images for signal temporal logic (STL) tasks [Ye et al., 2026]. Recent variants also consider risk-aware spatio-temporal logic in belief spaces, distribution temporal logic, and robust metric temporal logic under uncertain navigation times [Vahs et al., 2023, Yu et al., 2024, Linard et al., 2024]. These approaches show that semantic and geometric uncertainty are central to formal planning, but they typically optimize a single product model, a selected semantic interpretation, or a task-specific probabilistic predicate. Our focus is narrower but different: the uncertain semantic label of a cell is itself the automaton-edge trigger, so we preserve several feasible successor transitions as weighted intents before choosing the next motion.

POMDP and belief-space planning. POMDP planning gives the most general formulation for partial observability [Kaelbling et al., 1998, Ross et al., 2008], and recent online methods provide stronger guarantees, information-guided exploration, reachability-oriented heuristic search, recursive constraints, or learned chance-constraint surrogates [Barenboim and Indelman, 2023, Do Carmo Alves et al., 2023, Ho et al., 2024b,a, Moss et al., 2024]. However, an exact Bayes-adaptive formulation for our problem would maintain a joint belief over physical state, automaton state, and the semantic label of every cell. This state space grows with the product of map size, automaton size, and semantic-map posterior support, while omega-regular objectives under partial observation also face severe decidability and complexity barriers [Chatterjee et al., 2013]. We therefore do not compete with full belief-space optimality; instead, we approximate the task-relevant posterior by a small set of automaton-transition-conditioned reach-avoid intents and replan online in the physical grid.

Active perception and uncertain semantic maps. Active perception couples control with information acquisition [Bajcsy et al., 2016], often through POMDP or belief-space objectives with explicit information rewards [Spaan et al., 2014]. TOAPP [Ghasemi et al., 2020] specializes this principle to temporal-logic tasks in partially known semantic environments by combining a MAP-like task policy with active perception when the current policy is unreliable. This is closely related to our setting and is the basis of the TOAPP* baseline. The key modeling distinction is that our main experiments use static-view correlated observations: moving to a new viewpoint can be informative, but repeatedly querying the same viewpoint-target pair does not generate independent evidence. Under this regime, validating one selected hypothesis can still be brittle, so our planner converts the current semantic belief into multiple automaton-transition intents instead of asking whether to keep executing a single MAP-induced policy.

Learning and online updates for temporal objectives. Recent work studies learning and optimization with LTL or omega-regular objectives in MDPs and reinforcement learning, including probably approximately correct (PAC) learning, omega-regular decision processes, average-reward translations, and neural multi-task policies for LTL specifications [Perez et al., 2024, Hahn et al., 2024, Le et al., 2024, Jackermeier and Abate, 2025]. Our method is not a neural policy learner; it assumes an online semantic belief and a compiled task automaton. The calibration step is instead a local entropic mirror-descent update on the intent simplex [Beck and Teboulle, 2003, Arora et al., 2012]: raw intent weights serve as a prior, physical progress supplies the reward signal, and the update remains KL-close to the belief-induced weights.

B Limitations

The method is not globally optimal for the full POMDP induced by semantic uncertainty. Its optimality statement is local: given fixed transition intents and progress signals, calibration solves a KL-regularized progress maximization problem. The current planner also relies on a factorized semantic belief and a finite top-K intent set, so posterior correlations across cells are not represented

explicitly. Remaining failures concentrate in room-like maps with weak observers and complex LTL tasks, where timeouts correlate with high revisit and probing fractions. Coherence calibration can also reinforce an incorrect intent if that intent happens to yield short-term physical progress; the KL penalty, weight floor, conservative automaton update, and hard-risk gate mitigate but do not eliminate this failure mode. Finally, TOAPP* adapts the TOAPP planning and active-perception structure to our observation model, but is not a line-by-line reproduction of the original implementation.

Recurrent objectives. The finite-prefix formulation makes information gathering compete with task completion within a bounded horizon. Under an infinite-horizon average-cost objective, a finite initial exploration cost can instead become negligible: subject to maintaining possible future satisfaction, an agent may first reduce semantic uncertainty and then execute the recurrent task. This does not imply that safe exploration is always possible; it identifies a difference in how the objective values information acquisition. Recurrent tasks with evolving semantic labels would also require continued monitoring, motivating a different objective and uncertainty-management formulation.

C Additional Experimental Results

Confidence intervals and paired tests. Table 5 reports Wilson 95% confidence intervals for the large-split success rates. Table 6 reports matched-pair sign tests over identical feasible instances, aggregated across the four observer profiles. The tests compare episode-level success indicators and therefore account for the fact that methods are evaluated on the same maps and tasks.

Table 5: Large-split success rates with Wilson 95% confidence intervals.

Observer	Single	Raw Fusion	CoherenceAware
strong	0.943 [0.917, 0.961]	0.945 [0.920, 0.963]	0.966 [0.944, 0.979]
base	0.927 [0.898, 0.948]	0.954 [0.930, 0.970]	0.975 [0.955, 0.986]
weak	0.918 [0.888, 0.940]	0.927 [0.898, 0.948]	0.957 [0.933, 0.972]
very weak	0.908 [0.878, 0.932]	0.913 [0.883, 0.936]	0.938 [0.912, 0.957]

Table 6: Aggregated matched-pair sign tests over four observer profiles. "Fix" means CoherenceAware succeeds while the baseline fails; "Break" means the reverse.

Comparison	Fix	Break	Both success	Both fail	p
CoherenceAware vs. Single	98	37	1578	35	1.50×10^{-7}
CoherenceAware vs. Raw Fusion	57	15	1619	57	6.54×10^{-7}
CoherenceAware vs. MemoryAware	31	0	1645	72	9.31×10^{-10}

Task-family breakdown. The pooled rates in Table 2 combine reach-avoid, mixed, and complex LTL formulas. Table 7 reports the same large-split results stratified by task family and aggregated across the four observer profiles. The macro average gives equal weight to the three task families, rather than weighting by the number of generated formulas.

Table 7: Large-split success/violation/timeout rates by LTL task family, aggregated over the four observer profiles.

Task family	N	Single	Raw Fusion	CoherenceAware
reach-avoid	464	0.948/0.009/0.043	0.953/0.004/0.043	0.957/0.009/0.034
mixed	220	0.905/0.018/0.077	0.941/0.000/0.059	0.977/0.005/0.018
complex LTL	1064	0.917/0.016/0.067	0.926/0.008/0.066	0.956/0.008/0.036
Macro avg. S	–	0.923	0.940	0.963

Choice of K. Table 8 reports the large-split sweep over the number of retained intents for the full coherence-aware planner. Moderate fusion is best: $K = 4$ is the most stable choice for base, weak, and very weak observers, while $K = 2$ is closer to premature commitment and $K = 8$ can reintroduce multi-intent interference.

Table 8: Appendix K sweep for CoherenceAware Fusion on seeds 24–39.

Observer	K	N	S	T	Success steps
strong	2	437	0.963	0.034	52.93
strong	4	437	0.966	0.023	54.64
strong	8	437	0.968	0.018	54.80
base	2	437	0.959	0.039	53.79
base	4	437	0.975	0.018	57.15
base	8	437	0.963	0.025	56.85
weak	2	437	0.943	0.046	56.61
weak	4	437	0.957	0.032	59.84
weak	8	437	0.938	0.048	60.13
very weak	2	437	0.934	0.064	56.94
very weak	4	437	0.938	0.059	58.09
very weak	8	437	0.934	0.064	59.67

Component ablations. Table 9 reports the full component ablation. The KL-only variant nearly matches the full method, whereas the action-only variant remains close to the memory-aware control. This supports our interpretation that the decisive correction is intent-weight calibration on the simplex.

Table 9: Appendix component ablation on seeds 24–39. Entries show success/timeout rates.

Observer	Memory	Full	No physical	Action only	KL only	No escape
strong	0.945/0.043	0.966/0.023	0.957/0.032	0.945/0.043	0.966/0.023	0.970/0.023
base	0.966/0.027	0.975/0.018	0.970/0.023	0.966/0.027	0.975/0.018	0.982/0.016
weak	0.936/0.050	0.957/0.032	0.950/0.039	0.938/0.048	0.957/0.032	0.959/0.037
very weak	0.918/0.080	0.938/0.059	0.936/0.062	0.918/0.080	0.938/0.059	0.943/0.050

Risk-gate sensitivity. To check whether the gain comes only from a looser risk filter, Table 10 sweeps the hard-risk gate θ_H on a medium split. CoherenceAware improves success and reduces timeouts at all tested gates; at the default $\theta_H = 0.075$, it fixes 6/0 base-observer and 9/0 weak-observer Raw Fusion failures in matched success pairs.

Table 10: Risk-gate sensitivity on seeds 24–31. Entries are $S/V/T$; Fix/Break counts matched success changes of CoherenceAware relative to Raw Fusion.

Observer	θ_H	Raw Fusion	CoherenceAware	Fix/Break
base	0.050	0.903/0.000/0.097	0.938/0.004/0.058	10/2
base	0.075	0.951/0.000/0.049	0.978/0.000/0.022	6/0
base	0.120	0.942/0.000/0.058	0.965/0.000/0.035	6/1
weak	0.050	0.920/0.000/0.080	0.938/0.004/0.058	7/3
weak	0.075	0.903/0.013/0.084	0.942/0.018/0.040	9/0
weak	0.120	0.929/0.018/0.053	0.947/0.009/0.044	5/1

Scaling smoke. As a small stress check on grid size and horizon, Table 11 varies the map size while keeping the base observer and static-view sensor fixed. The calibrated planner retains the highest success rate across the three tested scales, although on the largest map its successful trajectories are longer than Raw Fusion.

External baseline sanity. Sanity checks confirm that TOAPP* performance is observation-regime sensitive rather than intrinsically broken. On the same 437 feasible base-observer instances as Table 4, Table 12 compares our main static-view correlated observer with a TOAPP-like iid repeated-view observer, where repeated measurements are treated as fresh Bayesian evidence. TOAPP* improves from $S = 0.451$ to $S = 0.865$ under this TOAPP-like observer; this indicates that the active-perception component is functional, but repeated-view information is important for TOAPP's

Table 11: Map-size/horizon smoke test on seeds 24–29. Entries show $S/V/T$ and average successful steps.

Size / H	Method	N	$S/V/T$	Steps
17×17 / 140	Single	172	0.977/0.006/0.017	46.64
17×17 / 140	Raw Fusion	172	0.971/0.000/0.029	43.99
17×17 / 140	CoherenceAware	172	0.983/0.006/0.012	43.06
19×19 / 160	Single	170	0.924/0.018/0.059	57.64
19×19 / 160	Raw Fusion	170	0.941/0.000/0.059	58.44
19×19 / 160	CoherenceAware	170	0.971/0.000/0.029	57.02
23×23 / 200	Single	173	0.942/0.006/0.052	90.37
23×23 / 200	Raw Fusion	173	0.913/0.000/0.087	89.58
23×23 / 200	CoherenceAware	173	0.948/0.000/0.052	94.82

advantage. The full method remains strong under both observation regimes, though the gap narrows under iid repeated-view sensing.

Table 12: Observation-model sanity for TOAPP*. The same 437 feasible base-observer instances are evaluated under the main static-view correlated observer and a TOAPP-like iid repeated-view observer.

Setting	Method	N	S	V	T	Success steps
static-view	TOAPP*	437	0.451	0.014	0.535	35.70
static-view	Single	437	0.927	0.018	0.055	57.48
static-view	Raw Fusion	437	0.954	0.002	0.043	58.97
static-view	CoherenceAware	437	0.975	0.007	0.018	57.15
TOAPP-like iid view	TOAPP*	437	0.865	0.053	0.082	44.65
TOAPP-like iid view	Single	437	0.954	0.002	0.043	48.75
TOAPP-like iid view	Raw Fusion	437	0.973	0.007	0.021	48.75
TOAPP-like iid view	CoherenceAware	437	0.979	0.007	0.014	48.59

Additional belief-style baseline sanity. We also tested an automaton-belief fusion variant that keeps a belief over plausible automaton states and builds posterior-weighted transition fields, but does not use coherence calibration. Table 13 places it on the same 226 oracle-feasible base-observer static-view configurations as MAP-replan, TOAPP*, Single, Raw Fusion, and CoherenceAware. A more direct product-distance QMDP-style baseline, using fully observable product-distance values averaged under the current belief, was implemented as well; this is a QMDP-style approximation because it scores actions with fully observable values rather than solving the full belief-space POMDP. It is substantially slower and times out on most of this subset; the full-split extension is reported in Appendix F. This medium subset contains reach-avoid, mixed, and complex-LTL tasks from seeds 24–31 and both map families.

Table 13: Belief-style baseline sanity. All rows use the same 226 oracle-feasible base-observer static-view configurations from seeds 24–31, both map families, and reach-avoid, mixed, and complex-LTL tasks.

Method	N	S	V	T	Success steps
MAP-replan	226	0.000	0.000	1.000	–
TOAPP*	226	0.442	0.009	0.549	38.85
Single ($K = 1$)	226	0.920	0.018	0.062	62.19
Raw Fusion ($K = 4$)	226	0.951	0.000	0.049	61.41
CoherenceAware ($K = 4$)	226	0.978	0.000	0.022	58.97
Automaton-belief fusion	226	0.774	0.022	0.204	54.05
Product-distance QMDP-style[†]	226	0.035	0.000	0.965	26.12

[†] The product-distance QMDP-style run required 337.3 minutes for 226 feasible configurations. The automaton-belief/point-q medium sanity run required 50.2 minutes while covering 226 feasible

configurations per method. Both belief-style variants are additionally evaluated on the full split in Table 20; the subset results above remain as the original matched comparison.

D Leakage Audit

The planner is evaluated in simulation, so we explicitly separate online information from oracle information. During an episode, the planner may access the current physical state, the known obstacle map, the task automaton, the maintained semantic belief, the static-view observation cache for observations already made, and contact-level evidence at the occupied cell. It may not access the true semantic map, future observations, oracle shortest paths, or oracle feasibility labels. Oracle information is used only offline to filter infeasible instances and to evaluate success, violation, timeout, and oracle path length.

The main leakage risks are therefore: using true semantic labels when constructing intents, using oracle feasibility during action selection, or updating the automaton state from the true label rather than the observation-derived belief. The current implementation avoids these in the planner interface: transition intents are built from belief-induced successor masses, hard/soft risks are computed from the belief, and the conservative automaton estimate is updated from current-cell evidence. The anonymous supplement includes a short structural leakage audit pointing to the planner entry points and the oracle-only evaluation routines.

E Planner Details

Table 14 lists the default CoherenceAware planner configuration. These parameters are fixed across observer profiles; the observer profile only changes semantic observation probabilities.

Table 14: Main planner hyperparameters.

Group	Parameter	Value
Intent set	retained intents K	4
Intent set	component threshold fraction	0.60
Risk	hard-risk action gate θ_H	0.075
Risk	soft-as-hard threshold θ_S	0.20
Risk	relaxed soft-risk weight λ_S	3.0
Safe Dijkstra	lost-intent penalty	1.5
Safe Dijkstra	probe progress weight	0.65
Safe Dijkstra	progress / reach / stay weights	1.0 / 0.75 / 0.08
Lookahead	depth / discount	1 / 0.65
Automaton update	progress threshold / margin	0.75 / 0.25
Automaton update	current observation required	true
Memory trigger	window / min steps	14 / 8
Memory trigger	unique / revisit fractions	0.60 / 0.35
Memory escape	horizon / score margin	7 / 0.05
Coherence trigger	revisit fraction / entropy	0.40 / 0.35
Coherence update	gain scale / weight floor	0.75 / 0.03
Coherence update	accept margin / progress epsilon	-0.02 / 0.05
Action coherence	opposition / low-support weights	0.65 / 0.35
Action coherence	support floor	0.45

The action score in Eq. (25) is implemented with lexicographic hard-risk filtering: actions with $H_t(s') > \theta_H$ are removed whenever at least one safe action exists. For each intent, reverse Dijkstra is first run on the strict graph $H_t \le \theta_H, S_{t,i} \le \theta_S$; if no strict route exists, the planner keeps the hard-risk gate and adds the relaxed soft-risk cost. The coherence update is only triggered after enough recent history is available and the trajectory shows high revisit fraction, high intent-weight entropy, and little weighted distance improvement.

Lost-intent score example. Equation (25) also penalizes moves that lose a plausible intent. Consider two intents with weights 0.6 and 0.4, and two non-stay actions with no reach bonus and equal auxiliary costs. Action A makes one unit of distance progress for the first intent but makes the

second intent unreachable. The unreachable intent contributes to the lost-intent penalty, not to the distance-progress term. With the reported progress weight 1.0 and lost-intent penalty 1.5, the relevant score is $1.0 \cdot 0.6 - 1.5 \cdot 0.4 = 0.0$. Action B reduces both intent distances by 0.5 and loses neither, giving $1.0(0.6 \cdot 0.5 + 0.4 \cdot 0.5) = 0.5$. The score therefore prefers B, preserving both plausible transitions.

F Additional Robustness and Scale Experiments

This section collects the additional experiments used in the discussion. It leaves the main benchmark and its reported results unchanged. Unless specified otherwise, the observer is base, the sensing regime is static-view, the semantic prior is neutral, and the main planner configuration is used.

Parameter selection and coherence triggers. The constants in Table 14 were selected as one global configuration through early small-scale tests focused on typical failure cases, then kept fixed for the reported comparisons. The existing K and risk gate sweeps appear in Tables 8 and 10. We additionally vary the revisit threshold over $0.30, 0.40, 0.50$ with entropy fixed at its default 0.35, and the entropy threshold over $0.25, 0.35, 0.45$ with revisit fixed at 0.40. Table 15 uses the full 437-instance protocol in each row; the default matches Table 2. The outcomes remain stable over these choices. Equal aggregate outcomes do not imply identical trigger decisions or trajectories.

Table 15: Coherence trigger sensitivity. The default row is shared by the two one-factor sweeps.

Revisit threshold	Entropy threshold	N	$S/V/T$
0.30	0.35	437	0.977/0.007/0.016
0.40	0.35	437	0.975/0.007/0.018
0.50	0.35	437	0.975/0.007/0.018
0.40	0.25	437	0.975/0.007/0.018
0.40	0.45	437	0.975/0.007/0.018

Additional seed block. We also evaluate seeds 1–15 using the same fixed configuration and all four observer profiles (Table 16). Oracle-feasibility filtering leaves 435 instances per observer-method pair, distinct from the 437 instances in the main split. This is an additional robustness check, not a claim that the original parameter-selection process used a prespecified held-out tuning split.

Table 16: Fixed-configuration evaluation on seeds 1–15; entries are satisfaction/violation/timeout rates.

Observer	N	Single	Raw Fusion	CoherenceAware
strong	435	0.926/0.014/0.060	0.936/0.007/0.057	0.954/0.007/0.039
base	435	0.917/0.023/0.060	0.915/0.014/0.071	0.949/0.016/0.034
weak	435	0.924/0.014/0.062	0.920/0.014/0.067	0.956/0.005/0.039
very weak	435	0.920/0.021/0.060	0.938/0.000/0.062	0.959/0.007/0.034

Stochastic execution. Under the Table 4 protocol, each non-stay action is replaced by another valid action with probability 0.10, and the planners replan from the realized state. Table 17 reports the results with all other benchmark settings unchanged. This is an execution-noise robustness check, not an optimal stochastic-transition planning experiment. It shows that useful task completion persists under disturbed execution, with increased violations and timeouts relative to the deterministic base setting.

Semantic class scale. We expand the local semantic alphabet by adding distractor labels while preserving the submitted task-template structure and planner configuration. The four-class control reproduces the main base result, and the six- and eight-class conditions use the same 437 map-task instance identifiers (Table 18). The experiment measures operation with a larger alphabet; it does not require success to decrease monotonically with alphabet size, since the resulting semantic ambiguities also change.

Table 17: Action replacement probability 0.10 under the base observer.

Method	N	$S/V/T$
Single	437	0.915/0.059/0.025
Raw Fusion	437	0.915/0.032/0.053
CoherenceAware	437	0.920/0.048/0.032

Table 18: Semantic alphabet scale with the same task-template structure; entries are satisfaction/violation/timeout rates.

Classes	N	Single	Raw Fusion	CoherenceAware
4	437	0.927/0.018/0.055	0.954/0.002/0.043	0.975/0.007/0.018
6	437	0.954/0.005/0.041	0.931/0.011/0.057	0.943/0.018/0.039
8	437	0.961/0.000/0.039	0.943/0.000/0.057	0.961/0.000/0.039

Larger task automata. The submitted complex-LTL templates compile to at most 8 states and 21 transitions. The added formula set reaches 11 states and 26 transitions, using $H = 220$ for longer accepting prefixes and retaining the base observer and planner configuration. Across 80 oracle-feasible instances, CoherenceAware obtains $S/V/T = 0.938/0.013/0.050$ (75 successes, one violation, four timeouts). The added formulas, in the notation g, b, r, y for green, blue, red, yellow, are

$$\mathsf{F}(g \wedge \mathsf{F}b) \wedge \mathsf{F}(y \wedge \mathsf{F}r) \wedge (\neg r \mathsf{U} g),$$
$$\mathsf{F}(g \wedge \mathsf{F}(b \wedge \mathsf{F}y)) \wedge \mathsf{F}(r \wedge \mathsf{F}g),$$
$$\neg r \mathsf{U}\big(g \wedge (\neg y \mathsf{U}(b \wedge (\neg r \mathsf{U}(y \wedge \mathsf{F}g))))\big).$$

These form a separate task set, not a resampled subset of the main 437 instances.

Task-template catalogue. Table 19 lists all 16 templates in the three task families used for the main comparison. The two map families and seeds 24–39 produce $16 \times 2 \times 16 = 512$ candidates before feasibility filtering.

Table 19: Complete main-benchmark task catalogue, with g, b, r, y denoting the four color propositions.

Family	Formula
reach avoid	$\neg b \mathsf{U} g$
reach avoid	$\neg g \mathsf{U} b$
reach avoid	$\neg r \mathsf{U} g$
reach avoid	$\neg y \mathsf{U} b$
mixed	$(\neg r \mathsf{U} g) \wedge (\mathsf{F} b)$
mixed	$(\neg y \mathsf{U} b) \wedge (\mathsf{F} g)$
complex ltl	$\neg(b \vee r) \mathsf{U}(y \wedge \mathsf{F} g)$
complex ltl	$\neg(r \vee y) \mathsf{U}(g \wedge \mathsf{F} b)$
complex ltl	$\neg r \mathsf{U}(g \wedge (\neg y \mathsf{U} b))$
complex ltl	$\neg r \mathsf{U}(g \wedge \mathsf{F} b)$
complex ltl	$\neg y \mathsf{U}(b \wedge (\neg r \mathsf{U} g))$
complex ltl	$\neg y \mathsf{U}(b \wedge \mathsf{F} g)$
complex ltl	$(\neg b \mathsf{U}(y \wedge \mathsf{F} g)) \wedge (\neg r \mathsf{U} g)$
complex ltl	$(\neg r \mathsf{U}(g \wedge \mathsf{F} b)) \wedge (\neg y \mathsf{U} b)$
complex ltl	$\mathsf{F}(b \wedge \mathsf{F}(g \wedge \mathsf{F} y))$
complex ltl	$\mathsf{F}(g \wedge \mathsf{F}(b \wedge \mathsf{F} y))$

Full-split belief-space comparisons. We extend the two variants in Table 13 from seeds 24–31 to seeds 24–39, covering all 437 oracle-feasible instances under the base observer, static-view sensing, and the same episode horizon. Table 20 reports automaton-belief fusion and the product-distance QMDP-style approximation used in the original subset comparison. Both run on the full split, although their satisfaction rates remain below the main methods in Table 2. Automaton-belief fusion

preserves uncertainty over future semantic transitions and additionally smooths the current task-progress estimate into a belief over source automaton states. In this benchmark, observations at the occupied cell are relatively reliable, so this smoothing introduces additional task-state uncertainty. A source-state belief is more relevant when the current cell label itself remains uncertain; the diagnostic examines this representation choice under the same observer and budget.

Table 20: Full-split extension of the belief-space comparisons in Table 13.

Method	N	S	V	T
Automaton-belief fusion	437	0.799	0.014	0.188
Product-distance QMDP-style	437	0.030	0.000	0.970

G Tasks with Multi-Proposition Labels

Tasks requiring joint labels. We evaluated CoherenceAware on four co-safe formulas requiring two propositions at the same cell: $\mathsf{F}(g \wedge b)$, $\mathsf{F}(r \wedge y)$, $\neg(r \vee y)\mathsf{U}(g \wedge b)$, and $\neg(b \vee r)\mathsf{U}(y \wedge g)$, where g, b, r, y denote green, blue, red, and yellow. The local alphabet contains all subsets of these four propositions, and the map generator permits joint labels. We used 19×19 maps from both map families, seeds 24–39, the base observer, neutral semantic priors, static-view sensing, and $H = 160$, with the original CoherenceAware planner and parameters unchanged. Of 128 generated candidates, 112 were oracle-feasible. CoherenceAware obtained 110 successes, one violation, and one timeout, or $S/V/T = 0.982/0.009/0.009$. Semantic labels were inferred from observations, not supplied to the planner.

Original complex-LTL templates. We additionally tested the original CoherenceAware planner on all ten complex-LTL formulas listed in Table 19, allowing a cell to carry a set of propositions. The local label alphabet is the power set of the four benchmark propositions, and some cells carry two propositions. The formulas and their temporal semantics are unchanged. We used the original 19×19 random-obstacle and room-grid maps, seeds 24–39, the base observer, neutral semantic priors, static-view sensing, and horizon $H = 160$. The planner and its parameters were unchanged. In particular, semantic labels were hidden from the planner and inferred from observations.

This gives 320 fixed map-task candidates. The oracle found 267 feasible instances; the other 53 were excluded by the same feasibility rule used for the main results, without replacing seeds. On the feasible instances, CoherenceAware achieved 254 successes, 3 violations, and 10 timeouts, or $S/V/T = 0.951/0.011/0.037$. The two nested-avoidance formulas account for nine of the ten timeouts. Joint labels may shorten some tasks while making avoidance constraints harder, so this experiment tests the multi-proposition interface rather than a uniformly harder task distribution.

H Task Planning on a PRM Roadmap

We also evaluated the unchanged CoherenceAware planner in a continuous two-dimensional workspace represented by a probabilistic roadmap (PRM). Each map contains four circular physical obstacles and two circular semantic regions per proposition. The roadmap has 400 nodes placed uniformly over free space, with collision-free edges connecting nearby nodes. Nodes are selected from geometry, not seeded at semantic targets; target and avoid labels remain hidden and are inferred from the same base static-view observer. The planner still selects its next graph move, with $H = 160$ roadmap moves per episode.

The three co-safe tasks are (i) reaching green and blue while avoiding red, (ii) reaching green before blue while avoiding red, and (iii) reaching green, blue, and yellow in order while avoiding red. Each task was run on seeds 0–99. Task progress is evaluated on the sequence of visited nodes: passing a target partway along an edge does not count as reaching it. To prevent a hidden avoid region from being skipped between nodes, red is checked along the entire executed edge for both oracle feasibility and outcome evaluation. The planner is not given the true red regions when building or scoring its roadmap.

All 300 generated instances were oracle-feasible. For the three tasks in the order above, the success/violation/timeout counts were $98/2/0$, $99/1/0$, and $99/1/0$, respectively. Overall, CoherenceAware obtained $S/V/T = 0.987/0.013/0.000$. This tests task-level planning on an irregular graph embedded in continuous geometry, including edge-level avoidance; it does not evaluate robot dynamics or a complete perception-to-control system.

Path to robot integration. A mapping and localization layer would first provide navigable regions or roadmap nodes, their connectivity, and robot pose. Visual semantic observations would then be projected onto these regions to obtain label likelihoods, from which CoherenceAware updates beliefs and selects the next task-relevant region or node. A motion planner and feedback controller would execute the corresponding motion. The first likely difficulty is converting visual observations into reliable region beliefs: an omitted task-relevant region, an incorrect observation-to-region association, or an overconfident wrong likelihood can produce incorrect intents or risk estimates. Addressing these errors requires semantic calibration, localization-aware projection, and online refinement of candidate regions or roadmap nodes; CoherenceAware does not itself repair the perception or localization system. Robot execution additionally requires dynamically feasible paths and feedback control. The roadmap experiment evaluates the task-planning step of this integration path.

I Reproducibility Notes

All main experiments use neutral semantic priors, deterministic grid transitions, horizon $H = 160$, sensing radius 10, map seeds 24–39, two map families, and three LTL task families. The reported large split filters oracle-infeasible instances and leaves 437 feasible episodes per observer-method pair out of 512 generated map-task instances; 75 instances, or 14.6%, are oracle-infeasible. If infeasible instances are counted as timeouts for all methods, the internal ranking is unchanged: unfiltered success rates for Single/Raw Fusion/CoherenceAware are 0.805/0.807/0.824 for strong, 0.791/0.814/0.832 for base, 0.783/0.791/0.816 for weak, and 0.775/0.779/0.801 for very weak. The anonymous supplement contains the planner code, benchmark runners, LTL conversion dependency, exact JSON outputs used for the tables in `adaptive_intent_learning/outputs/` and `analysis/outputs/`, selected wall-clock logs, and an artifact manifest. All main and appendix tables can be regenerated with `paper_neurips_2026/make_paper_tables.py`. The supplement also includes `artifact_verify.py`, which checks required files, scans for obvious non-anonymous local/remote traces, verifies key aggregate numbers, regenerates paper tables, and runs the structural true-map leakage audit. Citation consistency can be checked with `paper_neurips_2026/check_citations.ps1`, and the standalone leakage audit can be run with `paper_neurips_2026/check_no_true_map_leakage.py`. The combined local check script is `paper_neurips_2026/run_submission_checks.ps1`. Observed wall-clock logs are retained under `cache/train_logs/`: the base-observer local ablation blocks took about 1.9–2.2 hours per 1536-job block, the full external base-observer MAP/TOAPP run took about 12.2 hours, and the resumed remote eight-block ablation log records completion at 2175 minutes. The automaton-belief/point-q medium baseline sanity took 50.2 minutes for 226 feasible configurations per method, while the product-distance QMDP-style medium sanity took 337.3 minutes for 226 feasible configurations.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction describe co-safe LTL planning under uncertain semantic maps. Tables 2, 4, and 9 support the main empirical claims. The theory concerns local optimization, not global POMDP optimality or zero-violation safety.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The main-paper Discussion and Limitations paragraph and Appendix B discuss the scope of the task model, factorized beliefs, finite intent budget, local theory, and remaining failure modes.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions.
   - The authors should reflect on the scope of the claims made.
   - The authors should reflect on the factors that influence the performance of the approach.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: Section 3 states the assumptions for the KL-regularized calibration result: fixed intents, positive raw weights, and fixed progress estimates. The closed-form optimizer is derived in the main text; the paper treats this as a local surrogate result, not a global performance guarantee.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.

- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Sections 2–4 and Appendix I specify the map families, task families, observer profiles, seeds, horizon, oracle-feasible filtering, planner constants, JSON output locations, and table-generation scripts.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways, including detailed algorithms, code and data release, hosted models, or other suitable means.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility.

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

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

   Answer: [Yes]

   Justification: The anonymous supplement includes planner code, benchmark runners, Rabinizer dependency, exact JSON table outputs, selected logs, a manifest, README instructions, and `artifact_verify.py`.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines for more details.
   - While release of code and data is encouraged, a No answer is acceptable with justification.
   - The instructions should contain the exact command and environment needed to run to reproduce the results.
   - The authors should provide instructions on data access and preparation, including raw, preprocessed, intermediate, and generated data when applicable.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines, or state which ones are omitted and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions if applicable.

6. **Experimental setting/details**

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

   Answer: [Yes]

Justification: There is no neural-network training in the main method. Section 4 and Appendix I specify evaluation details; Table 1 gives observer profiles and Table 14 gives planner constants.

Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Appendix C reports Wilson confidence intervals and matched-pair sign tests over identical oracle-feasible instances.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated.
   - The method for calculating the error bars should be explained.
   - The assumptions made should be given.
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - For asymmetric distributions, the authors should be careful not to show symmetric error bars that would yield out-of-range values.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix I reports representative wall-clock logs. The anonymous supplement records the local Intel i7-14650HX workstation and 4-CPU remote worker used for checks and longer ablations.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper.

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work is a simulation study of planning algorithms and does not involve human subjects, private data, scraped datasets, or deployed robotic systems.

   Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity.

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: The paper studies a simulated planning algorithm and does not release a deployed decision system or model. We do not claim direct deployment impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses, fairness considerations, privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments.

11. **Safeguards**

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

    Answer: [N/A]

    Justification: No high-risk model, scraped dataset, or dual-use artifact is released; the release contains simulation code, synthetic maps, and numerical outputs.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards.
    - Datasets that have been scraped from the Internet could pose safety risks.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this.

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: The experiments use generated maps and public software dependencies. The supplement includes the repository license, dependency list, and notes on the Rabinizer4/Owl third-party components used for LTL conversion.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license should be included for each asset.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided.

13. **New assets**

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

Answer: [Yes]

Justification: The supplement documents the released synthetic-map experiment outputs, JSON tables, logs, generated figures, README, and artifact manifest.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets if applicable.

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

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

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

Answer: [N/A]

Justification: The paper does not involve crowdsourcing or research with human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [N/A]

Justification: LLM assistance was used only for drafting, editing, and early schematic figure prototyping. It was not part of the core method, experimental pipeline, table generation, or reported results.

Guidelines:

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