# RISK BUDGETS FOR SUPERVISING AGENTS WITH HIDDEN GOALS UNDER TEMPORAL LOGIC CONSTRAINTS

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

## ABSTRACT

Autonomous agents are often deployed with fixed goal-conditioned skills while their operating requirements continue to change. A planetary rover, for example, may face new ordering or deadline requirements after launch, when retraining and recertifying its flight-qualified controller is infeasible. We study an external supervisor for such post-deployment requirements, expressed in linear temporal logic on finite traces (LTLf). It observes physical states, infers the agent's active goal from motion, and may reassign that goal, while the low-level controller remains inaccessible. The central difficulty is that decisions that appear acceptable in isolation can accumulate into an episode-level violation probability above the requested tolerance. We introduce an intent-aware supervisor whose risk account prices autonomy relative to a task-level takeover policy, charging increases in continuation risk and refunding decreases. Goal-conditioned dynamic programming precomputes the violation risks of autonomy until the next task event, one autonomous step, and takeover, while Bayesian filtering combines them under the current intent belief. We prove an exact accounting identity: the bad-prefix violation probability plus the expected terminal balance equals the requested tolerance. Experiments show that the method enforces instance-specific tolerances while retaining sparse high-level intervention across various LTLf specifications.

## 1 INTRODUCTION

Many deployed autonomous agents choose among high-level goals and execute each goal through a fixed low-level skill. New operating requirements may arise after deployment, such as completing one task only after another or within a deadline. Temporal logic (TL) expresses such requirements precisely (Baier et al., 2008; Vasile et al., 2017); we use linear temporal logic on finite traces (LTLf) for finite observation sequences (Camacho et al., 2018). We consider a specification unavailable to an agent whose controller may be proprietary or costly to recertify. Similar settings motivate runtime monitoring in certified flight software (Giannakopoulou et al., 2020; Dutle et al., 2020).

We study an external supervisor that maintains an LTLf monitor and enforces the specification through the agent's existing goal interface without changing its fixed skills. The supervisor observes physical states and task events, but not the active goal or low-level actions, and therefore infers the goal from motion. An intervention assigns a goal that the agent follows until the next task event. Over an $N$-step interval, the objective is to bound the violation probability by a requested tolerance $\alpha$ while keeping intervention sparse, measured by the fraction of time spent under assigned goals.

The main obstacle is the mismatch between decision-local risk and an episode-level guarantee. Consider a simple case in which each autonomous step has an independent probability $\zeta$ of causing a violation, while supervised execution is safe. Allowing autonomy for all $N$ steps then gives an episode-level violation probability $1-(1-\zeta)^N$. Meeting tolerance $\alpha$ therefore requires $\zeta$ to shrink approximately as $\alpha/N$; local admissibility alone does not control cumulative risk.

We address this mismatch by treating the permitted episode-level risk as a resource shared across supervisory decisions. Starting from a goal-assignment strategy with computable remaining risk, the supervisor spends this resource when autonomy increases the predicted risk and refunds it when a new observation lowers that risk. Because the active goal is hidden, each prediction combines intent inference from motion with the fixed skills and the LTLf monitor. We show that these charges and refunds exactly account for changes in episode-level violation risk. A nonnegative balance therefore



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.

guarantees the requested tolerance while permitting autonomy whenever the risk margin allows it. The contributions are as follows.

- We formulate post-deployment supervision of a fixed agent with hidden intent, goal-level control, an episode-level LTLf violation tolerance, and sparse intervention (Section 4).
- We develop an intent-aware supervisor that infers the active goal and allocates the tolerance as a shared risk budget across autonomous and assigned-goal continuations.
- We prove an exact identity between bad-prefix violation probability and unused budget, yielding an episode-level guarantee whenever the balance remains nonnegative (Theorem 1).
- Across five LTLf families, we compare with five baselines and evaluate tolerance adherence, intervention, component ablations, and controlled scaling (Section 7).

## 2 Related Work

**Temporal logic specifications and runtime supervision.** Reward machines and specification-conditioned policies incorporate TL requirements into learning (Icarte et al., 2022; Shao & Kwiatkowska, 2023; Kantaros & Wang, 2024; Voloshin et al., 2023; Jackermeier & Abate, 2025; Jiang et al., 2021; Aksaray et al., 2016), while other methods compose learned skills to execute a TL task supplied at test time (Jothimurugan et al., 2021; Araki et al., 2021; Qiu et al., 2023; Yalcinkaya et al., 2024). Runtime shields and safety filters intervene through an action interface (Alshiekh et al., 2018; Jansen et al., 2020; Carr et al., 2023; Könighofer et al., 2023; Wabersich & Zeilinger, 2021; Aksaray et al., 2021), and runtime assurance switches to a verified backup when a monitor predicts a hazard (Sha, 2001). Supervisory control disables events in a fixed plant (Ramadge & Wonham, 1987). Our supervisor redirects goals selected by a fixed agent.

**Risk prediction and allocation.** Predictive monitors and agent guardrails use rules or risk estimates to flag hazards or trigger intervention (Junges et al., 2021; Cairoli et al., 2023; Wang et al., 2026; Kamath et al., 2025; Wang et al., 2025). Chance-constrained planning and stochastic barrier certificates allocate or bound trajectory risk (Ono & Williams, 2008; Santana et al., 2016; Moss et al., 2024; Bouchard et al., 2010; Huynh et al., 2014; Sootla et al., 2022; Chow et al., 2018; Prajna et al., 2007; Steinhardt & Tedrake, 2012). Recent probabilistic shields augment an MDP with a safety level (Hamel-De le Court et al., 2025), account for worst-case LTL satisfaction in robust MDPs (Hamel-De le Court et al., 2026), or use history-dependent safety–permissiveness guarantees (Heck et al., 2026). Since a constraint enforced separately at each planning step need not bound mission risk (Köhler et al., 2023; Wang & Gros, 2021), our supervisor charges exact conditional risk changes to one episode-level account while acting through the goal interface of a fixed agent with hidden intent.

**Hidden intent and oversight.** Intent inference estimates goals from behavior, shared autonomy uses those estimates to provide assistance (Baker et al., 2009; Ziebart et al., 2009; Javdani et al., 2018; Jain & Argall, 2019; Fu & Topcu, 2015), predictive monitors infer temporal-logic intents to forecast violations without intervening (Yoon & Sankaranarayanan, 2021), and reachability filters have been applied to a requested goal in place of an action (Jeong et al., 2025). Budgeted oversight and risk-controlled post-processing preserve a fixed policy while limiting harmful outcomes (Greenblatt et al., 2023; Griffin et al., 2026; Bhatt et al., 2025; Joshi et al., 2026; Lekeufack et al., 2024). Here, inferred intent determines autonomous-continuation risk, and the budget coordinates interventions across the episode. Appendix A.1 compares these lines of work.

## 3 Preliminaries: LTLf and Bad-Prefix Automata

To formalize requirements over a finite $N$-step observation window, we use LTLf (De Giacomo & Vardi, 2013). Let $\mathsf{AP}$ be a finite set of atomic propositions, let $\Gamma = 2^{\mathsf{AP}}$ be the alphabet of propositional valuations, and let $\Gamma^* := \bigcup_{k\geq 0} \Gamma^k$ be the set of all finite words over $\Gamma$, including the empty word $\lambda$. LTLf formulas are evaluated on nonempty finite words $w \in \Gamma^+ := \Gamma^* \setminus \{\lambda\}$; $w \models \varphi$ means that $w$ satisfies $\varphi$. We use the next, until, eventually, always, and weak-until operators as $\mathbf{X}$, $\mathbf{U}$, $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{W}$, respectively.

A standard approach is to translate the formula into a deterministic finite automaton (DFA) $\mathcal{D}_\varphi = (Q, \Gamma, \delta, q_{\text{init}}, F)$, with states $Q$, initial state $q_{\text{init}} \in Q$, and accepting states $F \subseteq Q$ (Fuggitti, 2019).

The transition function $\delta$ reads one symbol. If a generated DFA omits a transition, we complete it by adding a rejecting trap $q_\perp$ to $Q$, directing each missing transition to $q_\perp$, and setting $\delta(q_\perp, \sigma) = q_\perp$ for every $\sigma \in \Gamma$; thus $q_\perp \notin F$ and $\delta : Q \times \Gamma \to Q$ is total. For $u \in \Gamma^*$ and $\sigma \in \Gamma$, its extension to words is defined by $\delta^*(q, \lambda) = q$ and $\delta^*(q, u\sigma) = \delta(\delta^*(q, u), \sigma)$. For a nonempty word $w$, the DFA accepts $w$ exactly when $\delta^*(q_{\text{init}}, w) \in F$, equivalently when $w \models \varphi$. Acceptance is evaluated when the finite word ends, so an accepting state need not be absorbing. A finite word is a bad prefix when no finite continuation is accepted:

$$\text{Bad}(\varphi) := \{u \in \Gamma^* : \ \delta^*(q_{\text{init}}, uv) \notin F \text{ for every } v \in \Gamma^*\}. \tag{1}$$

Equivalently, $Q^{\varphi}_{\text{bad}} := \{q \in Q : \delta^*(q, v) \notin F \text{ for every } v \in \Gamma^*\}$ contains the states from which acceptance is unreachable, and $u \in \text{Bad}(\varphi)$ exactly when $\delta^*(q_{\text{init}}, u) \in Q^{\varphi}_{\text{bad}}$. A nonaccepting state outside this set represents a pending obligation; for example, a bounded response becomes bad only after its deadline passes without a response. We use this bad-prefix interpretation for runtime supervision (Kupferman & Vardi, 2001; Bauer et al., 2011); Appendix A.2 gives the finite-trace semantics and the computation of $Q^{\varphi}_{\text{bad}}$.

## 4 Problem Formulation

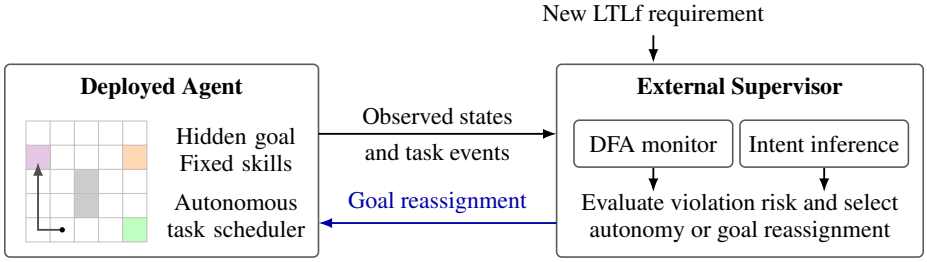



Figure 1: An external supervisor observes the deployed agent and monitors a new LTLf requirement. It intervenes through goal reassignment; the agent's fixed skills execute low-level actions.

Figure 1 summarizes the setting formalized below: a supervisor observes a fixed agent with hidden goals and may reassign its goal to enforce a new LTLf requirement with limited intervention.

### 4.1 Agent, Intent, and Task Events

The agent starts at $s_0$ with goals from a finite set $\mathcal{G}$. Let $(\mathcal{S}, \mathcal{A}, P)$ be a finite controlled Markov process with state $s \in \mathcal{S}$, action $a \in \mathcal{A}$, and kernel $P(s' \mid s, a)$. Its fixed stochastic skill library is $\Pi = \{\pi_g\}_{g \in \mathcal{G}}$, where an active goal $g$ induces actions $a \sim \pi_g(\cdot \mid s)$ and the marginalized dynamics $P_g(s' \mid s) := \sum_{a \in \mathcal{A}} P(s' \mid s, a)\pi_g(a \mid s)$. Task events represent meaningful outcomes of physical transitions, such as entering an inspection zone. Let $\Sigma = 2^{\mathsf{AP}} \setminus \{\varnothing\}$ be the alphabet of nonempty proposition sets, and let $L : \mathcal{S} \times \mathcal{S} \to \Sigma \cup \{\varepsilon\}$ assign each transition its event label or $\varepsilon \notin \Sigma$ when no event occurs. We write $\ell_{t+1} = L(s_t, s_{t+1})$ and call an $\varepsilon$-labeled transition silent; removing silent labels from $\ell_{1:t}$ yields the event word observed so far.

**Autonomous task scheduling.** At time $t$, the agent holds a single goal $g_t \in \mathcal{G}$. We call this goal its intent. The goal is hidden from the supervisor, whose initial uncertainty is represented by the distribution $b_0$, with $g_0 \sim b_0$. During autonomous execution, $s_{t+1} \sim P_{g_t}(\cdot \mid s_t)$. The fixed task scheduler $\Lambda = (\mathcal{M}, m_0, \tau, \kappa)$ has finite state space $\mathcal{M}$, initial state $m_0$, state update $\tau$, and next-goal kernel $\kappa$. Its state $m_t$ summarizes the observed event history, and after $\ell_{t+1}$ it updates as

$$m_{t+1} = \tau(m_t, \ell_{t+1}), \quad g_{t+1} \sim \kappa(\cdot \mid g_t, m_{t+1}, \ell_{t+1}). \tag{2}$$

Silence preserves the task process, $\tau(m, \varepsilon) = m$ and $\kappa(g' \mid g, m, \varepsilon) = \mathbf{1}\{g' = g\}$, so the goal remains fixed between events. The supervisor knows $m_t$ from the event word; $g_t$ remains hidden.

### 4.2 Specification, Supervision, and Objective

The labels form a physical-step word $\ell_{1:N}$ over $\bar{\Sigma} := \Sigma \cup \{\varepsilon\}$, with $\varepsilon$ identified with the empty valuation in $\Gamma$. Event-relative operators skip silent transitions, whereas physical-step operators advance on every transition; Appendix E.1 gives their translation into standard LTLf. We compile the LTLf specification $\varphi$ into one completed DFA and compute its bad set as in Section 3. This yields the

bad-prefix monitor $\mathcal{A}_\varphi = (\mathcal{Q}, \bar{\Sigma}, \bar{\delta}, q_0, Q_{\mathrm{bad}})$, where $(\mathcal{Q}, q_0, Q_{\mathrm{bad}}) := (Q, q_{\mathrm{init}}, Q^\varphi_{\mathrm{bad}})$, $\bar{\delta}$ maps $\varepsilon$ to the empty valuation before applying the DFA transition, and $Q_{\mathrm{bad}}$ is absorbing. Along a physical trajectory, the monitor evolves as

$$q_{t+1} = \bar{\delta}(q_t, \ell_{t+1}). \tag{3}$$

The episode violates the post-deployment requirement when the monitored prefix reaches the bad set, $\mathcal{V}_{0:N} := \{\exists t \in \{0, \dots, N\} : q_t \in Q_{\mathrm{bad}}\}$. Evaluation ends after the $N$th transition, and an obligation becomes a violation during this interval when its deadline expires before completion.

**Information and intervention.** The supervisor observes physical states and task events and records its past inputs. Its history before acting at time $t$ is $h_t = (s_{0:t}, \ell_{1:t}, u_{0:t-1})$. Because the active goal is hidden, the supervisor maintains a belief $b_t$ over $g_t$, whose Bayesian update is defined in Section 5.1. When no assignment is in force, its action set is $\mathcal{U}(s_t) := \{\mathsf{allow}\} \cup \mathcal{G}_{\mathrm{cmd}}(s_t)$: it can leave the current goal unchanged with allow or assign a goal from the interface set $\mathcal{G}_{\mathrm{cmd}}(s_t) \subseteq \mathcal{G}$.

A supervision policy $\mu = (\mu_t)_{t=0}^{N-1}$ generates $u_t \sim \mu_t(\cdot \mid h_t)$ subject to the holding constraint: once a goal is assigned, the assignment remains in force until the next event. Let $\widetilde{g}_t$ denote the goal executed during step $t$: $\widetilde{g}_t = g_t$ when $u_t = \mathsf{allow}$, and $\widetilde{g}_t = u_t$ when the supervisor assigns $u_t \in \mathcal{G}_{\mathrm{cmd}}(s_t)$. The physical transition follows $P_{\widetilde{g}_t}$, so the selected skill continues to choose low-level actions. The scheduler uses $\widetilde{g}_t$ in place of $g_t$ in equation 2; at the next event, the assignment ends and the scheduler selects the agent's next goal.

**Risk and intervention objectives.** Given a tolerance $\alpha \in [0, 1]$ chosen by the operator, the supervisor must satisfy

$$\mathbb{P}^\mu(\mathcal{V}_{0:N}) \le \alpha \tag{4}$$

under its closed-loop policy $\mu$. Intervention is measured by the expected fraction of steps during which an assigned goal is in force, $J_{\mathrm{int}}(\mu) := \frac{1}{N}\mathbb{E}^\mu\left[\sum_{t=0}^{N-1} \mathbf{1}\{u_t \neq \mathsf{allow}\}\right]$. The supervisor therefore uses the deployed process and monitor to satisfy equation 4 while keeping $J_{\mathrm{int}}(\mu)$ small, assigning a goal when the predicted reduction in violation risk warrants intervention.

# 5 Supervision with a Risk Budget

## 5.1 Bayesian intent filtering

The supervisor represents uncertainty about the active goal by $b_t(g) := \mathbb{P}(g_t = g \mid h_t)$. After choosing $u_t$, the executed-goal distribution is

$$b^u_t(g) := \mathbf{1}\{u_t = \mathsf{allow}\} b_t(g) + \mathbf{1}\{u_t \in \mathcal{G}\}\mathbf{1}\{g = u_t\}. \tag{5}$$

The observed transition supplies the likelihood $P_g(s_{t+1} \mid s_t)$. Bayes' rule followed by the scheduler update gives

$$b_{t+1}(g') = \frac{\sum_{g \in \mathcal{G}} b^u_t(g) P_g(s_{t+1} \mid s_t) \kappa(g' \mid g, m_{t+1}, \ell_{t+1})}{\sum_{g \in \mathcal{G}} b^u_t(g) P_g(s_{t+1} \mid s_t)}, \quad m_{t+1} = \tau(m_t, \ell_{t+1}). \tag{6}$$

The controlled information state is $x_t = (s_t, q_t, m_t, b_t, n_t)$, where $n_t = N - t$ is the number of remaining steps. Appendix B.1 derives equation 6. Proposition 1, proved in Appendix C.1, establishes that $b_t$ is the exact posterior and $x_t$ is a controlled information state.

## 5.2 Violation risks of supervisory choices

At any time with no assignment in force, including within an autonomous window, the supervisor may assign a goal or leave the current goal unchanged. We evaluate three continuations. A TAKEOVER assigns the minimum-risk available goal and holds it until the next event. A WINDOW leaves the current goal unchanged until that event, while a STEP leaves it unchanged for one transition and then reassesses. To compute a remaining-episode violation probability for a window or step, we use takeover as the planned behavior after its autonomous period ends. When that boundary is reached, the supervisor may keep the planned takeover or select another autonomous period; the risk account charges the resulting change.

Let $D_n(g, s, q)$ be the violation probability with $n$ steps remaining when goal $g$ is held until the next event and optimal held takeover is used thereafter. A window and an assigned goal share these

held-goal dynamics. At an event boundary, minimizing $D_n$ over assignable goals gives the takeover risk and command,

$$\rho_n^*(s,q) := \min_{g \in \mathcal{G}_{\mathrm{cmd}}(s)} D_n(g,s,q), \quad g_n^*(s,q) \in \arg\min_{g \in \mathcal{G}_{\mathrm{cmd}}(s)} D_n(g,s,q). \tag{7}$$

Let $A_n(g,s,q)$ be the corresponding risk when $g$ is retained for one step and takeover begins at the next state. For an information state $x$ with physical, monitor, and belief components $(s,q,b)$, the posterior-averaged risks and their differences from immediate takeover are

$$\begin{aligned} \overline{D}_n(x) &:= \sum_g b(g) D_n(g,s,q), \quad \Delta_n^{\mathrm{win}}(x) := \overline{D}_n(x) - \rho_n^*(s,q), \\ \overline{A}_n(x) &:= \sum_g b(g) A_n(g,s,q), \quad \Delta_n^{\mathrm{step}}(x) := \overline{A}_n(x) - \rho_n^*(s,q). \end{aligned} \tag{8}$$

The step premium may be negative because the next observation can improve the takeover choice. Accordingly, $\rho_n^*$ is the optimum within the held-takeover class, not a lower bound over every feasible supervision policy. The Bellman recursions in Appendix B.2 form a finite-horizon dynamic program that builds the risk tables offline from $n = 0$ through $N$; online supervision then uses table lookups and posterior averages. Proposition 2 establishes exactness and complexity, while Lemmas 1 and 2 establish the required ordering.

### 5.3 The risk budget

A reference continuation specifies how the agent will be controlled for the rest of the episode. The supervisor may revise it after each observed autonomous transition, while an assigned goal remains in force until the next event. At a revision, an increase in conditional violation probability is charged to the balance and a decrease is credited back; maintaining a nonnegative balance yields the episode-level bound in Theorem 1.

**Committed risk.** Let $z_t^{\mathrm{in}} \in \{\textsc{boundary}, \textsc{window}, \textsc{takeover}\}$ identify the reference continuation carried into time $t$. Its risk $C_t^-$ equals $\rho_{n_t}^*(s_t,q_t)$ at a fresh boundary before the next mode is chosen, $\overline{D}_{n_t}(x_t)$ inside a window, and $D_{n_t}(g_t^{\mathrm{cmd}},s_t,q_t)$ during a held assignment, where $g_t^{\mathrm{cmd}}$ is the assigned goal. When a new selection is available, define the mode-indexed outgoing risks

$$C_t(\textsc{window}) := \overline{D}_{n_t}(x_t), \quad C_t(\textsc{step}) := \overline{A}_{n_t}(x_t), \quad C_t(\textsc{takeover}) := \rho_{n_t}^*(s_t,q_t). \tag{9}$$

After the supervisor selects mode $z_t$ and before the next physical transition, set $C_t^+ = C_t(z_t)$; an inherited takeover instead has $C_t^+ = C_t^-$. Thus the argument of $C_t(z)$ selects a candidate, while $C_t^-$ and $C_t^+$ are the risks before and after the decision.

**Admission.** At time zero the committed continuation is optimal held takeover, $C_0^- = \rho_N^*(s_0,q_0)$. This initialization admits tolerances $\alpha \geq C_0^-$; initialize $B_0 = B_{\mathrm{init}} := \alpha - C_0^-$. The candidate transfer for selecting mode $z$, the realized transfer, and the balance update are

$$\chi_t(z) := C_t(z) - C_t^-, \quad \chi_t := C_t^+ - C_t^-, \quad B_{t+1} := B_t - \chi_t, \tag{10}$$

where a new selection sets $\chi_t = \chi_t(z_t)$ and an inherited takeover has $\chi_t = 0$. Write $\Delta_t(z) := C_t(z) - \rho_{n_t}^*(s_t,q_t)$, so $\Delta_t(\textsc{window}) = \Delta_{n_t}^{\mathrm{win}}(x_t)$ and $\Delta_t(\textsc{step}) = \Delta_{n_t}^{\mathrm{step}}(x_t)$. With a spending cap $\eta > 0$, mode $z$ is admitted when

$$\mathrm{adm}_t(z) := \mathbf{1}\{\chi_t(z) \leq B_t\}\mathbf{1}\{\Delta_t(z) \leq \eta B_{\mathrm{init}}\} = 1. \tag{11}$$

The balance test provides the risk guarantee, while the second test uses $\eta$ to control how quickly that balance may be spent. Preferring the window keeps an autonomous task contiguous, so the supervisor admits a window when possible, then one step, and assigns the optimal takeover goal if neither is admitted.

**Re-evaluation and refunds.** At a boundary, the charge equals the added risk $\Delta_t(z)$. Inside a window, continuing costs zero and switching to a step or takeover returns nonnegative credit by Lemmas 1 and 2; Appendix C gives the mode-by-mode calculation. Each decision preserves $C_t^- + B_t = C_t^+ + B_{t+1}$, admission keeps $B_{t+1} \geq 0$, and the physical transition determines the next incoming reference. The terminal balance is $B_N$.

Algorithm 1 assembles these definitions into the online procedure. The monitor and continuation tables are constructed offline, so each online step requires one filter update and table lookups followed by posterior averages over the goal set.

**Algorithm 1** Risk-budgeted supervisor

Precompute $\{D_n, A_n, \rho_n^*, g_n^*\}_{n=0}^{N}$ by backward induction (Appendix B.2)
**if** $\alpha < \rho_N^*(s_0, q_0)$ **then** stop: $\alpha$ is below the takeover risk
$B_0 \leftarrow B_{\text{init}}$; choose $\eta > 0$; $z_0^{\text{in}} \leftarrow$ BOUNDARY
**for** $t = 0, 1, \ldots, N-1$ **do**
  $n \leftarrow N - t$; set $C_t^-$ from $z_t^{\text{in}}$; $z_t \leftarrow z_t^{\text{in}}$; $\chi_t \leftarrow 0$
  **if** $z_t^{\text{in}} \neq$ TAKEOVER **then**
    $z_t \leftarrow$ the first mode in (WINDOW, STEP) admitted by equation 11, or TAKEOVER if neither is admitted; $\chi_t \leftarrow \chi_t(z_t)$
    **if** $z_t =$ TAKEOVER **then** $g^{\text{cmd}} \leftarrow g_n^*(s_t, q_t)$
  **end if**
  $B_{t+1} \leftarrow B_t - \chi_t$
  Set $u_t \leftarrow$ allow for $z_t \in \{\text{WINDOW}, \text{STEP}\}$ and $u_t \leftarrow g^{\text{cmd}}$ for takeover; execute one step and observe $(s_{t+1}, \ell_{t+1})$
  Update $q_{t+1}$, $m_{t+1}$, and $b_{t+1}$ by equations 3, 2, and 6; record a violation if $q_{t+1} \in Q_{\text{bad}}$
  $z_{t+1}^{\text{in}} \leftarrow$ BOUNDARY if $z_t =$ STEP or $\ell_{t+1} \in \Sigma$; otherwise $z_{t+1}^{\text{in}} \leftarrow z_t$
**end for**

## 6 Theoretical Analysis

We first establish the exactness of the quantities used by Algorithm 1, then prove its risk-account identity and guarantee. Appendices B–D give the computations, proofs, and additional results. The analysis uses the following conditions:

**Assumption 1** (Known deployed process)**.** The supervisor knows $\{P_g\}_{g \in \mathcal{G}}$, $L$, $\Lambda$, $\mathcal{A}_\varphi$, and the initial goal distribution $b_0$.

**Assumption 2** (Goal interface)**.** (i) $\mathcal{G}_{\text{cmd}}(s)$ is nonempty for every $s \in \mathcal{S}$. (ii) If $g \in \mathcal{G}_{\text{cmd}}(s)$, $P_g(s' \mid s) > 0$, and $L(s, s') = \varepsilon$, then $g \in \mathcal{G}_{\text{cmd}}(s')$. (iii) Every $g$ with $b_0(g) > 0$ belongs to $\mathcal{G}_{\text{cmd}}(s_0)$, and every $g'$ with $\kappa(g' \mid g, \tau(m, \ell), \ell) > 0$ belongs to $\mathcal{G}_{\text{cmd}}(s')$ whenever $P_g(s' \mid s) > 0$ and $\ell = L(s, s') \in \Sigma$.

Assumption 2 keeps every goal possible under autonomy commandable after silent transitions and scheduler updates.

### 6.1 Exact information state and continuation risks

**Proposition 1** (Exact posterior and information state)**.** *Under Assumption 1, suppose* $b_0(g) = \mathbb{P}(g_0 = g \mid s_0)$. *Then every realized transition has positive predictive likelihood almost surely, and equation 6 gives* $b_t(g) = \mathbb{P}(g_t = g \mid h_t)$ *for every* $t$. *Moreover,* $x_t = (s_t, q_t, m_t, b_t, n_t)$ *is a controlled information state: conditioned on* $(x_t, u_t)$, *the distribution of* $(x_{t+1}, \mathbf{1}\{q_{t+1} \in Q_{\text{bad}}\})$ *is independent of the earlier history.*

**Proposition 2** (Exactness and computation of the continuation values)**.** *Under Assumption 2, for every remaining horizon* $n$ *and safe pair* $(s, q)$, *that is,* $q \notin Q_{\text{bad}}$*: (i) backward induction using the Bellman recursions in Appendix B.2 computes the window and step risks of Section 5.2 exactly; (ii)* $\rho_n^*(s, q)$ *is the minimum violation probability over policies that assign a goal at each event boundary and hold it until the next event, attained by* $g_n^*(s, q)$ *in equation 7; and (iii) constructing the* $D$, $\rho^*$, *and* $A$ *tables through horizon* $N$ *takes* $O(N|\mathcal{G}||\mathcal{S}||\mathcal{Q}|d_P)$ *operations and* $O(N|\mathcal{G}||\mathcal{S}||\mathcal{Q}|)$ *storage, independent of the scheduler state, where* $d_P := \max_{g,s} |\operatorname{supp} P_g(\cdot \mid s)|$ *is the largest number of one-step successors under any skill, and each online posterior average costs* $O(|\mathcal{G}|)$.

The proofs and lemmas in Appendix C give $\rho_n^*(s, q) \leq \overline{D}_n(x)$ and $\overline{A}_n(x) \leq \overline{D}_n(x)$ under Assumptions 1 and 2, because Lemma 1 keeps the posterior supported on assignable goals. Hence a window has nonnegative added risk, and one autonomous step never adds more risk than a whole window.

### 6.2 Risk guarantee for the budgeted supervisor

The values $C_t^-$ and $C_t^+$ in Section 5.3 are the exact conditional risks before and after a decision. If $\mathcal{F}_t$ denotes the information available at that decision, first-step conditioning gives

$$\mathbb{E}[C_{t+1}^- \mid \mathcal{F}_t] = C_t^+. \tag{12}$$

After one transition, a window continues after silence and becomes optimal held takeover after an event; a step becomes takeover, while a held assignment retains its goal after silence and selects a new one after an event. Appendix C.4 formalizes this closure property.

**Theorem 1** (Risk account). *Suppose the supervisor makes history-measurable choices among feasible members of a family of reference continuations closed under continuation (Appendix C.4), whose exact risks satisfy equation 12, initializes $B_0 = \alpha - C_0^-$, and updates the balance by equation 10. Then*

$$\mathbb{P}(\mathcal{V}_{0:N}) = C_0^- + \mathbb{E}\Big[\sum_{t=0}^{N-1} \chi_t\Big] \quad \textit{and} \quad \mathbb{P}(\mathcal{V}_{0:N}) + \mathbb{E}[B_N] = \alpha. \tag{13}$$

*If in addition $\alpha \geq C_0^-$ and $\chi_t \leq B_t$ for every $t$, then $B_N \geq 0$ and $\mathbb{P}(\mathcal{V}_{0:N}) \leq \alpha$.*

The proof tracks $M_t := C_t^- + B_t$: decisions preserve it exactly and physical transitions preserve it in conditional expectation, so its terminal value $\mathbf{1}\{\mathcal{V}_{0:N}\} + B_N$ gives the identity and risk bound. Appendix C.4 gives the full argument. Risk tolerance as a martingale state is classical (Bouchard et al., 2010; Huynh et al., 2014); here the increments are exact posterior-conditional risks obtained from the offline continuation tables.

**Corollary 1** (Tolerance guarantee). *Under Assumptions 1 and 2, for every $\alpha \in [\rho_N^*(s_0, q_0), 1]$, Algorithm 1 maintains $B_t \geq 0$ and satisfies $\mathbb{P}(\mathcal{V}_{0:N}) + \mathbb{E}[B_N] = \alpha$ and hence $\mathbb{P}(\mathcal{V}_{0:N}) \leq \alpha$.*

Theorem 1 places no restriction on how admitted modes are chosen, so the balance test in equation 11 guarantees the tolerance under every history-measurable preference over modes, of which the window-first order in Algorithm 1 is one.

**Relation to local risk thresholds.** Repeated local checks can admit fresh risk throughout an episode. Propositions 3 and 4 in Appendix D.1 show that a uniformly valid depth-$H$ threshold must shrink on the order of $\alpha H/N$ as restarts accumulate. The risk account instead charges every change to one episode-level balance. In Appendix D, Theorem 2 lower-bounds the intervention that goal ambiguity forces on any supervisor, Proposition 5 identifies when takeover carries no risk, and Proposition 6 transfers the guarantee to a deployed process whose kernels are close in total variation.

## 7 Experiments

**Environments and specifications.** We generate randomized grid worlds with side lengths from 9 to 11 and four or five task zones: 12 development maps for parameter selection and 20 held-out maps for evaluation. Each episode runs over an observation window of $N = 120$ steps. The agent holds one zone as its goal and moves toward it under a fixed skill; arriving there emits a task event carrying that zone's propositions. Layouts, zone positions, initial states, intent priors, and slip probabilities vary across maps. We test five constraint families: bounded separation, timing composite, precedence chain, bounded response, and allowance composite. Appendix E.1 gives their LTLf formulas and event/step semantics. We evaluate map–formula pairs: an evaluation pair $j$ couples one held-out map with one family, instantiated with that map's bounds.

**Methods and metrics.** We compare the budgeted supervisor with five baselines: completion-label, first-event, and depth-$H$ risk thresholds in the style of predictive monitoring (Junges et al., 2021; Cairoli et al., 2023), and bounded-reachability and multiplicative-shield rules that adapt action-level shielding to assignable goals (Alshiekh et al., 2018; Jansen et al., 2020; Heck et al., 2026). Every method receives the same observations and intent posterior and acts through the same goal interface. Table 5 in Appendix E.2 summarizes their decision rules. We report violation risk and intervention, the fraction of physical steps executed under an assigned goal. A method succeeds when its risk respects the requested tolerance and its intervention is low at a matched risk level.

For map–formula pair $j$, let $p_{\mathrm{aut},j}$ be the exact no-supervision risk, computed by the dynamic program in Appendix B.3, and let $\rho_{N,j}^* := \rho_N^*(s_0, q_0)$ be its initial takeover risk. We keep the 97 of 100 pairs with $p_{\mathrm{aut},j} - \rho_{N,j}^* \geq .15$, so that intervention can lower the violation risk by a meaningful margin. These endpoints vary widely across pairs: a common raw tolerance would lie below $\rho_{N,j}^*$ on some pairs, outside the scope of Corollary 1, and above $p_{\mathrm{aut},j}$ on others, where no intervention is needed. We therefore normalize requested and achieved risks between them:

$$\alpha_j = \rho_{N,j}^* + \beta\big(p_{\mathrm{aut},j} - \rho_{N,j}^*\big), \quad \widetilde{p}_j = \big(\mathbb{P}(\mathcal{V}_{0:N}^{(j)}) - \rho_{N,j}^*\big) / \big(p_{\mathrm{aut},j} - \rho_{N,j}^*\big). \tag{14}$$

Here $\beta$ is the requested normalized tolerance, $\mathcal{V}_{0:N}^{(j)}$ is the violation event on pair $j$, and $\widetilde{p}_j$ is the achieved normalized risk. Held takeover anchors zero and unsupervised execution anchors one, so one $\beta$ sets every pair's tolerance at the same relative position and normalized risks can be averaged across pairs. Unless stated otherwise, reported risk in tables and figures refers to $\widetilde{p}_j$.

**Evaluation protocol.** All methods share 300 episode seeds on each held-out pair, and dynamic programming computes $p_{\mathrm{aut},j}$ and $\rho^*_{N,j}$ exactly. For Section 7.1, each baseline frontier is the lower convex hull of its held-out parameter sweep. Our frontier sweeps $\beta$, with its spending cap selected on the 12 development maps at each value, and the family-level frontiers are interpolated at normalized risk .3. Sections 7.2 and 7.3 instead select parameters on the development maps at $\beta = .25$ and freeze them for held-out evaluation; the scaling study uses separate development maps (Appendix E.4). Macro averages weight the five families equally. Brackets are 95% intervals from 5,000 paired bootstrap draws, and $\pm$ is one standard error over pairs. Appendix E.3 lists all parameter grids, seeds, and sample counts.

### 7.1 Intervention at matched risk

Table 1 reads these frontiers at achieved normalized risk .3. Besides the five baselines, it lists a baseline envelope, the best baseline in each specification family, which takes the lowest intervention that any of the five reaches at that risk. Because different baselines win in different families, the envelope's macro average is lower than that of any single baseline.

Table 1: Intervention at achieved normalized risk .3; lower is better. Entries are estimates $\pm$ one standard error from resampling map–formula pairs. Reduction is relative to the baseline envelope, with a 95% interval that resamples the same pairs for every method.

| Constraint | Ours | C-label | First | $H$-risk | Mult. shield | Reach. | Envelope | Reduction |
|---|---|---|---|---|---|---|---|---|
| Bounded separation | **.029±.005** | .099±.024 | .065±.014 | .065±.013 | .166±.014 | .216±.014 | .065 | 56% [44, 62] |
| Timing composite | **.133±.008** | .417±.042 | .356±.032 | .217±.023 | .177±.009 | .391±.029 | .177 | 25% [18, 32] |
| Precedence chain | **.082±.006** | .253±.048 | .129±.019 | .119±.016 | .128±.014 | .418±.058 | .119 | 31% [4, 42] |
| Bounded response | **.054±.008** | .700±.001 | .194±.021 | .085±.011 | .139±.009 | .204±.029 | .085 | 37% [25, 45] |
| Allowance composite | **.088±.010** | .183±.044 | .120±.019 | .105±.013 | .229±.012 | .361±.019 | .105 | 17% [3, 22] |
| Macro average | **.077±.003** | .330±.016 | .173±.010 | .118±.007 | .168±.005 | .318±.015 | .110 | 30% [23, 34] |

Our method has the lowest estimated intervention in all five families. Averaged across families, intervention decreases from .110 for the envelope to .077, a relative reduction of $30.1\%$ with a 95% bootstrap interval of $[22.9\%, 33.6\%]$. The macro-average reduction stays between $24\%$ and $35\%$ at all six achieved risks we compare, .05, .1, .2, .3, .5, and .75 (Figure 3 in Appendix F.3).

### 7.2 Tolerance adherence and horizon transfer

Deployment requires meeting the requested tolerance on every pair with parameters fixed in advance, which Corollary 1 guarantees for our method under the model. We therefore freeze the development-map settings at $\beta = .25$ and record an exceedance when the 95% Wilson lower bound on a pair's violation probability, from its 300 episodes, is above $\alpha_j$. We then lengthen the horizon from $N = 120$ to $N = 200$ without retuning, rebuilding only the horizon-dependent risk tables and tolerances for the same maps, formulas, and episode seeds. This tests whether our method still meets every requested tolerance when the observation window grows.

Table 2: Frozen-parameter transfer from $N = 120$ to $N = 200$. Risk and intervention are macro means $\pm$ one standard error; Excess counts pairs with a detected tolerance violation.

| | $N = 120$ | | | $N = 200$ | | |
|---|---|---|---|---|---|---|
| Method | Risk | Intervention | Excess | Risk | Intervention | Excess |
| Ours | .110±.005 | **.107±.005** | **0/97** | .128±.005 | **.103±.005** | **0/97** |
| Completion label | .119±.018 | .455±.014 | 12/97 | .145±.020 | .462±.016 | 17/97 |
| First event | .162±.019 | .219±.012 | 15/97 | .203±.023 | .244±.014 | 23/97 |
| $H$-risk | .110±.011 | .154±.010 | 5/97 | .134±.013 | .163±.012 | 13/97 |
| Multiplicative shield | .065±.017 | .224±.008 | 7/97 | .068±.018 | .213±.008 | 8/97 |
| Reachability | .154±.021 | .402±.015 | 19/97 | .185±.023 | .421±.017 | 23/97 |

At $N = 120$, all six methods have macro-average normalized risk below .25, yet ours is the only method with no detected pairwise exceedance (Table 2). It matches $H$-risk in mean risk, .110, with

lower intervention, .107 versus .154; $H$-risk exceeds five tolerances. The multiplicative shield is more conservative on average, with risk .065 and intervention .224, but still exceeds seven tolerances. After extending the horizon to $N = 200$ without retuning, ours again has zero exceedances, whereas the $H$-risk count rises from 5 to 13. This agrees with the repeated admission of local risk across $H$-step checks described in Section 6.2. Appendix F.3 reports the full horizon-transfer plot and numerical agreement with the risk-account identity.

### 7.3 Mechanism studies

**Intent inference.** We replace the exact posterior from Proposition 1 with its mode, a uniform belief, or the true goal, leaving the supervision rule unchanged (Table 3). The posterior mode lowers intervention to .084 but raises normalized risk to .267 and exceeds 26 of 97 tolerances. The uniform belief raises intervention to .160 and exceeds 9 tolerances. The exact posterior has no detected exceedance and stays within .006 intervention of the true-goal oracle. Appendix D.2 derives an information-theoretic lower bound on the probability of violation plus the probability of any goal assignment under ambiguous intent (Theorem 2).

Table 3: Intent-representation ablation.

| Intent representation | Risk | Intervention | Excess |
|---|---|---|---|
| Exact posterior (ours) | .110±.005 | .107±.005 | 0/97 |
| Posterior mode | .267±.016 | .084±.006 | 26/97 |
| Uniform belief | .142±.009 | .160±.005 | 9/97 |
| *Oracle (true goal)* | .082±.004 | .101±.005 | 0/97 |

**Risk allocation.** These ablations keep the risk account and change how the supervisor spends it, removing the single-step fallback, the re-evaluation inside a window, or refunds, to identify which of these reduces intervention. Every variant has no detected tolerance exceedance, and the full algorithm has the lowest intervention (Table 11 in Appendix F.5).

### 7.4 Scalability

We use ten configurations, five grid sizes from 10×10 to 50×50 in steps of 10, each with 4 or 8 goals, while keeping the response monitor fixed. The supervision horizon scales with grid width; a separate computation study holds it fixed. Parameters are selected only on independent $10 \times 10$ development maps at $\beta = .25$. We compare with $H$-risk and the multiplicative shield, whose family-wise minima reproduce the envelope in Table 1.

In all configurations, our method has the lowest intervention, reducing aggregate intervention by $34\%$ relative to $H$-risk and by $81\%$ relative to the multiplicative shield, and it has no detected tolerance exceedance, while the shield exceeds the tolerance on 2 of 50 map–configuration pairs. Build time and table storage grow roughly linearly with $|\mathcal{S}|$, and the online update stays below .02 ms per step; the largest case, a 50×50 grid with eight goals and $N = 600$, builds in 3.101 seconds and uses 427.7 MB. Appendix E.4 gives the protocol, Appendices F.1 and F.2 report every configuration, and Table 9 reports the main-benchmark costs.

## 8 Conclusion and Future Work

We presented a supervisor that limits irrecoverable violations of a post-deployment LTLf requirement on a fixed agent through sparse goal assignments. It infers the hidden goal from motion, evaluates autonomous and takeover continuations by finite-horizon dynamic programming, and charges their risk differences to an episode-level balance. Under the stated assumptions, the risk account is exact and Algorithm 1 keeps its balance nonnegative, so the supervisor meets every tolerance no smaller than the risk of optimal held takeover (Corollary 1). Across five constraint families, it reduces intervention relative to the best baseline at matched risk and is the only evaluated method with no detected tolerance exceedance on held-out maps.

The current risk calculations use known skill and scheduler dynamics and assume that assigned goals are followed. Future work will account for model uncertainty and imperfect compliance, extending the error analysis in Appendix D.4. Our guarantee counts irrecoverable violations within $N$ steps; obligations still pending at the horizon could be carried into a later window or included in a terminal-satisfaction objective. Certified approximations of the dynamic program could support larger state and monitor spaces.

### AI USE STATEMENT

The authors conceived the research idea, designed the algorithm and initial theoretical framework, and built the core codebase. Generative AI tools assisted with language polishing, related-work discovery, code editing and debugging, proof and data consistency checks, and refinement of mathematical notation and exposition. The authors reviewed and validated all AI-assisted material and take responsibility for the final content of this work.

### ETHICS STATEMENT

All experiments use synthetic grid worlds and involve no human-subject or personal data. The method controls the modeled probability that a deployed agent violates an externally imposed TL requirement. Safety-critical deployment requires validation of the dynamics, scheduler, intent prior, and specification. Appendix D.4 gives a total-variation bound for model error. Goal reassignment redirects agent behavior, so deployed specifications and supervision policies require appropriate governance.

### REPRODUCIBILITY STATEMENT

Appendices B and C provide the dynamic-programming derivations and complete proofs. Appendix E specifies the constraint formulas, baseline rules, parameter grids, random seeds, sample counts, and experimental protocols. Appendix F reports supplementary checks, ablations, and scaling results. A standalone demo provided as supplementary material implements Algorithm 1 on one held-out map with its fixed skills, hidden-goal process, and bounded-separation LTLf monitor.

## REFERENCES

Derya Aksaray, Austin Jones, Zhaodan Kong, Mac Schwager, and Calin Belta. Q-learning for robust satisfaction of signal temporal logic specifications. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pp. 6565–6570. IEEE, 2016.

Derya Aksaray, Yasin Yazıcıoğlu, and Ahmet Semi Asarkaya. Probabilistically guaranteed satisfaction of temporal logic constraints during reinforcement learning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6531–6537. IEEE, 2021.

Mohammed Alshiekh, Roderick Bloem, Rüdiger Ehlers, Bettina Könighofer, Scott Niekum, and Ufuk Topcu. Safe reinforcement learning via shielding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Ashwani Anand, Satya Prakash Nayak, Ritam Raha, and Anne-Kathrin Schmuck. Follow the stars: Dynamic $\omega$-regular shielding of learned probabilistic policies. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems*, pp. 2978–2986, 2026.

Brandon Araki, Xiao Li, Kiran Vodrahalli, Jonathan DeCastro, Micah Fry, and Daniela Rus. The logical options framework. In *International Conference on Machine Learning*, pp. 307–317. PMLR, 2021.

Christel Baier, Joost-Pieter Katoen, and Kim Guldstrand Larsen. *Principles of model checking*, volume 735. MIT press Cambridge, 2008.

Chris L Baker, Rebecca Saxe, and Joshua B Tenenbaum. Action understanding as inverse planning. *Cognition*, 113(3):329–349, 2009.

Andreas Bauer, Martin Leucker, and Christian Schallhart. Runtime verification for ltl and tltl. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 20(4):1–64, 2011.

Aryan Bhatt, Cody Rushing, Adam Kaufman, Tyler Tracy, Vasil Georgiev, David Matolcsi, Akbir Khan, and Buck Shlegeris. Ctrl-z: Controlling ai agents via resampling. *arXiv preprint arXiv:2504.10374*, 2025.

Bruno Bouchard, Romuald Elie, and Nizar Touzi. Stochastic target problems with controlled loss. *SIAM Journal on Control and Optimization*, 48(5):3123–3150, 2010.

Francesca Cairoli, Nicola Paoletti, and Luca Bortolussi. Conformal quantitative predictive monitoring of stl requirements for stochastic processes. In *Proceedings of the 26th ACM international conference on hybrid systems: computation and control*, pp. 1–11, 2023.

Alberto Camacho, Jorge Baier, Christian Muise, and Sheila McIlraith. Finite ltl synthesis as planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 28, pp. 29–38, 2018.

Steven Carr, Nils Jansen, Sebastian Junges, and Ufuk Topcu. Safe reinforcement learning via shielding under partial observability. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 14748–14756, 2023.

Yinlam Chow, Mohammad Ghavamzadeh, Lucas Janson, and Marco Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167): 1–51, 2018.

Giuseppe De Giacomo and Moshe Y Vardi. Linear temporal logic and linear dynamic logic on finite traces. Association for Computing Machinery, 2013.

Aaron Dutle, César Munoz, Esther Conrad, Alwyn Goodloe, Ivan Perez, Swee Balachandran, Dimitra Giannakopoulou, Anastasia Mavridou, Thomas Pressburger, et al. From requirements to autonomous flight: an overview of the monitoring icarous project. 2020.

Jie Fu and Ufuk Topcu. Synthesis of shared autonomy policies with temporal logic specifications. *IEEE Transactions on Automation Science and Engineering*, 13(1):7–17, 2015.

Francesco Fuggitti. Ltlf2dfa, 2019.

Dimitra Giannakopoulou, Anastasia Mavridou, Julian Rhein, Thomas Pressburger, Johann Schumann, and Nija Shi. Formal requirements elicitation with fret. In *International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ-2020)*, number ARC-E-DAA-TN77785, 2020.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. Ai control: Improving safety despite intentional subversion. 2023.

Charlie Griffin, Louis Thomson, Buck Shlegeris, and Alessandro Abate. Games for ai control: Models of safety evaluations of ai deployment protocols. In *Proceedings of IASEAI Conference*, volume 2, pp. 204–216, 2026.

Edwin Hamel-De le Court, Francesco Belardinelli, and Alexander W. Goodall. Probabilistic shielding for safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16091–16099, 2025.

Edwin Hamel-De le Court, Thom Badings, Alessandro Abate, Francesco Belardinelli, and Francesco Fabiano. Robust shielding for safe reinforcement learning. *arXiv preprint arXiv:2606.00270*, 2026.

Linus Heck, Filip Macák, Roman Andriushchenko, Milan Češka, and Sebastian Junges. Shields to guarantee probabilistic safety in mdps. In *International Conference on Computer Aided Verification*, pp. 576–600. Springer, 2026.

Vu Anh Huynh, Leonid Kogan, and Emilio Frazzoli. A martingale approach and time-consistent sampling-based algorithms for risk management in stochastic optimal control. In *53rd IEEE Conference on Decision and Control*, pp. 1858–1865. IEEE, 2014.

Rodrigo Toro Icarte, Toryn Q Klassen, Richard Valenzano, and Sheila A McIlraith. Reward machines: Exploiting reward function structure in reinforcement learning. *Journal of Artificial Intelligence Research*, 73:173–208, 2022.

Mathias Jackermeier and Alessandro Abate. Deepltl: Learning to efficiently satisfy complex ltl specifications for multi-task rl. In *International Conference on Learning Representations*, volume 2025, pp. 14000–14028, 2025.

Siddarth Jain and Brenna Argall. Probabilistic human intent recognition for shared autonomy in assistive robotics. *ACM Transactions on Human-Robot Interaction (THRI)*, 9(1):1–23, 2019.

Nils Jansen, Bettina Könighofer, Sebastian Junges, Alex Serban, and Roderick Bloem. Safe reinforcement learning using probabilistic shields. In *31st International Conference on Concurrency Theory (CONCUR 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

Shervin Javdani, Henny Admoni, Stefania Pellegrinelli, Siddhartha S Srinivasa, and J Andrew Bagnell. Shared autonomy via hindsight optimization for teleoperation and teaming. *The International Journal of Robotics Research*, 37(7):717–742, 2018.

Hyun Joe Jeong, Rosy Chen, and Andrea Bajcsy. Robots that suggest safe alternatives. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 179–186. IEEE, 2025.

Yuqian Jiang, Suda Bharadwaj, Bo Wu, Rishi Shah, Ufuk Topcu, and Peter Stone. Temporal-logic-based reward shaping for continuing reinforcement learning tasks. In *Proceedings of the AAAI Conference on artificial Intelligence*, volume 35, pp. 7995–8003, 2021.

Sunay Joshi, Tao Wang, Hamed Hassani, and Edgar Dobriban. Risk-controlled post-processing of decision policies. *arXiv preprint arXiv:2605.06479*, 2026.

Kishor Jothimurugan, Suguman Bansal, Osbert Bastani, and Rajeev Alur. Compositional reinforcement learning from logical specifications. *Advances in Neural Information Processing Systems*, 34:10026–10039, 2021.

Sebastian Junges, Hazem Torfah, and Sanjit A Seshia. Runtime monitors for markov decision processes. In *International Conference on Computer Aided Verification*, pp. 553–576. Springer, 2021.

Adharsh Kamath, Sishen Zhang, Calvin Xu, Shubham Ugare, Gagandeep Singh, and Sasa Misailovic. Enforcing temporal constraints for llm agents. *arXiv preprint arXiv:2512.23738*, 2025.

Yiannis Kantaros and Jun Wang. Sample-efficient reinforcement learning with temporal logic objectives: Leveraging the task specification to guide exploration. *IEEE Transactions on Automatic Control*, 70(5):2873–2888, 2024.

Johannes Köhler, Ferdinand Geuss, and Melanie N Zeilinger. On stochastic mpc formulations with closed-loop guarantees: Analysis and a unifying framework. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 6692–6699. IEEE, 2023.

Bettina Könighofer, Julian Rudolf, Alexander Palmisano, Martin Tappler, and Roderick Bloem. Online shielding for reinforcement learning. *Innovations in Systems and Software Engineering*, 19(4):379–394, 2023.

Orna Kupferman and Moshe Y Vardi. Model checking of safety properties. *Formal methods in system design*, 19(3):291–314, 2001.

Jordan Lekeufack, Anastasios N Angelopoulos, Andrea Bajcsy, Michael I Jordan, and Jitendra Malik. Conformal decision theory: Safe autonomous decisions from imperfect predictions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11668–11675. IEEE, 2024.

Robert J Moss, Arec Jamgochian, Johannes Fischer, Anthony Corso, and Mykel J Kochenderfer. Constrainedzero: Chance-constrained pomdp planning using learned probabilistic failure surrogates and adaptive safety constraints. 2024.

Masahiro Ono and Brian C Williams. Iterative risk allocation: A new approach to robust model predictive control with a joint chance constraint. In *2008 47th IEEE Conference on Decision and Control*, pp. 3427–3432. IEEE, 2008.

Stephen Prajna, Ali Jadbabaie, and George J Pappas. A framework for worst-case and stochastic safety verification using barrier certificates. *IEEE Transactions on Automatic Control*, 52(8): 1415–1428, 2007.

Wenjie Qiu, Wensen Mao, and He Zhu. Instructing goal-conditioned reinforcement learning agents with temporal logic objectives. *Advances in Neural Information Processing Systems*, 36:39147–39175, 2023.

Peter J Ramadge and W Murray Wonham. Supervisory control of a class of discrete event processes. *SIAM journal on control and optimization*, 25(1):206–230, 1987.

Pedro Santana, Sylvie Thiébaux, and Brian Williams. Rao*: An algorithm for chance-constrained pomdp's. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

Lui Sha. Using simplicity to control complexity. *IEEE Software*, 18(4):20, 2001.

Daqian Shao and Marta Kwiatkowska. Sample efficient model-free reinforcement learning from ltl specifications with optimality guarantees. 2023.

Aivar Sootla, Alexander I Cowen-Rivers, Taher Jafferjee, Ziyan Wang, David H Mguni, Jun Wang, and Haitham Ammar. Sauté rl: Almost surely safe reinforcement learning using state augmentation. In *International Conference on Machine Learning*, pp. 20423–20443. PMLR, 2022.

Jacob Steinhardt and Russ Tedrake. Finite-time regional verification of stochastic non-linear systems. *The International Journal of Robotics Research*, 31(7):901–923, 2012.

Alexandre B Tsybakov and Vladimir Zaiats. *Introduction to nonparametric estimation*, volume 11. Springer, 2009.

Cristian-Ioan Vasile, Derya Aksaray, and Calin Belta. Time window temporal logic. *Theoretical Computer Science*, 691:27–54, 2017.

Cameron Voloshin, Abhinav Verma, and Yisong Yue. Eventual discounting temporal logic counterfactual experience replay. In *International Conference on Machine Learning*, pp. 35137–35150. PMLR, 2023.

Kim Peter Wabersich and Melanie N Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021.

Haoyu Wang, Chris M Poskitt, Jun Sun, and Jiali Wei. Pro2guard: Proactive runtime enforcement of llm agent safety via probabilistic model checking. *arXiv e-prints*, pp. arXiv–2508, 2025.

Haoyu Wang, Christopher M Poskitt, and Jun Sun. Agentspec: Customizable runtime enforcement for safe and reliable llm agents. In *Proceedings of the 2026 IEEE/ACM 48th International Conference on Software Engineering*, pp. 2938–2950, 2026.

Kai Wang and Sébastien Gros. Recursive feasibility of stochastic model predictive control with mission-wide probabilistic constraints. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 2312–2317. IEEE, 2021.

Yifeng Xiao and Pierluigi Nuzzo. Symbolic temporal supervision of llm agents using contracts. *arXiv preprint arXiv:2609.18128*, 2026.

Beyazit Yalcinkaya, Niklas Lauffer, Marcell Vazquez-Chanlatte, and Sanjit Seshia. Compositional automata embeddings for goal-conditioned reinforcement learning. *Advances in Neural Information Processing Systems*, 37:72933–72963, 2024.

Hansol Yoon and Sriram Sankaranarayanan. Predictive runtime monitoring for mobile robots using logic-based bayesian intent inference. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8565–8571. IEEE, 2021.

Brian D Ziebart, Nathan Ratliff, Garratt Gallagher, Christoph Mertz, Kevin Peterson, J Andrew Bagnell, Martial Hebert, Anind K Dey, and Siddhartha Srinivasa. Planning-based prediction for pedestrians. In *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3931–3936. IEEE, 2009.

# A Supplementary Background

## A.1 Positioning Relative to Prior Work

Table 4 compares the combination of properties addressed by the related work in Section 2.

Table 4: Positioning relative to prior work.

| Line of work | Fixed agent + latent goal | Goal-level control | New TL at runtime | Episode risk bound | Exact risk account | Intervention cost |
|---|---|---|---|---|---|---|
| Action-level shields and safety filters | × | × | ○ | ○ | × | ○ |
| Budget-carrying probabilistic shields | × | × | ○ | ✓ | ○ | ○ |
| Predictive runtime monitoring | ○ | × | ○ | × | × | × |
| Chance-constrained planning with a risk state | × | × | × | ✓ | ○ | × |
| Temporal-logic goal and skill composition | × | ✓ | ○ | × | × | × |
| Shared autonomy with intent inference | ○ | ○ | ○ | × | × | ○ |
| Budgeted oversight of untrusted agents | ○ | × | × | ○ | × | ✓ |
| Runtime guardrails for LLM agents | ○ | × | ✓ | × | × | ○ |
| This paper | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

The rows draw on the following references:

- Action-level shields and safety filters: Alshiekh et al. (2018), Jansen et al. (2020), Carr et al. (2023), Könighofer et al. (2023), and Anand et al. (2026).
- Budget-carrying probabilistic shields: Hamel-De le Court et al. (2025), Hamel-De le Court et al. (2026), and Heck et al. (2026).
- Predictive runtime monitoring: Yoon & Sankaranarayanan (2021), Junges et al. (2021), and Cairoli et al. (2023).
- Chance-constrained planning with a risk state: Bouchard et al. (2010), Huynh et al. (2014), Chow et al. (2018), Sootla et al. (2022), and Moss et al. (2024).
- Temporal-logic goal and skill composition: Jothimurugan et al. (2021), Qiu et al. (2023), Yalcinkaya et al. (2024), and Jackermeier & Abate (2025).
- Shared autonomy with intent inference: Javdani et al. (2018), Jain & Argall (2019), and Fu & Topcu (2015).
- Budgeted oversight of untrusted agents: Greenblatt et al. (2023), Griffin et al. (2026), Bhatt et al. (2025), and Joshi et al. (2026).
- Runtime guardrails for LLM agents: Wang et al. (2026), Kamath et al. (2025), Wang et al. (2025), and Xiao & Nuzzo (2026).

A check denotes a defining feature, a circle a related or partial treatment, and a cross a property outside the line's scope.

## A.2 LTLf Semantics and Bad-State Computation

This subsection gives the temporal-logic definitions used by the bad-prefix monitor in Section 3. Formulas over $p \in \mathsf{AP}$ have the grammar

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}\varphi \mid \varphi_1 \,\mathbf{U}\, \varphi_2. \tag{15}$$

For a nonempty finite word $w = \sigma_1 \cdots \sigma_T \in \Gamma^T$ with $T \geq 1$ and a position $j \in \{1, \ldots, T\}$, $\top$ always holds, $p$ holds when $p \in \sigma_j$, and $\neg$ and $\wedge$ have their Boolean meaning. The formula $\mathbf{X}\varphi$ holds at position $j$ when $j < T$ and $\varphi$ holds at $j + 1$. The formula $\varphi_1 \,\mathbf{U}\, \varphi_2$ holds when $\varphi_2$ holds at some position $k \in \{j, \ldots, T\}$ and $\varphi_1$ holds from $j$ through $k - 1$. Satisfaction $w \models \varphi$ is evaluated

at the first position. We use the Boolean abbreviations $\vee$ and $\rightarrow$, together with $\mathbf{F}\varphi := \top \mathbf{U}\, \varphi$, $\mathbf{G}\varphi := \neg\mathbf{F}\neg\varphi$, and $\varphi_1 \,\mathbf{W}\, \varphi_2 := (\varphi_1 \,\mathbf{U}\, \varphi_2) \vee \mathbf{G}\varphi_1$.

The states from which $F$ is reachable are found by backward graph search; the remaining states form $Q^{\varphi}_{\mathrm{bad}}$. This set is closed under transitions, so its states may be merged into one rejecting trap without changing either the accepted language or the bad-prefix language in equation 1.

# B Filtering and Dynamic Programming

This section collects the computations used by the supervisor. The Bayesian filter is updated online after each observed transition. The continuation risks and the no-supervision risk are finite-horizon dynamic programs computed once offline and then queried by Algorithm 1.

## B.1 Bayesian filter update

Conditional on $(h_t, u_t)$, the executed goal has distribution $b^u_t$ from equation 5. For an observed successor $s'$, the joint conditional probability of the executed goal $g$, successor state $s'$, and next goal $g'$ is

$$b^u_t(g) P_g(s' \mid s_t) \kappa(g' \mid g, \tau(m_t, L(s_t, s')), L(s_t, s'))\,.$$

Summing over the previous goal and normalizing by the predictive probability $\sum_g b^u_t(g) P_g(s' \mid s_t)$ gives equation 6. Thus one online update consists of a likelihood multiplication, a sum over the previous goal, and propagation through the scheduler kernel.

## B.2 Continuation-risk dynamic program

Let $n$ be the number of remaining physical transitions. For a successor $s'$, write $\ell = L(s, s')$ and $q' = \bar{\delta}(q, \ell)$. Set $D_n = A_n = 1$ when $q \in Q_{\mathrm{bad}}$ and $D_0 = A_0 = 0$ when $q \notin Q_{\mathrm{bad}}$. For a safe state and $n \geq 1$, first-step conditioning gives

$$\begin{aligned} D_n(g, s, q) = \sum_{s' \in \mathcal{S}} P_g(s' \mid s) \Big[ & \mathbf{1}\{q' \in Q_{\mathrm{bad}}\} + \mathbf{1}\{\ell = \varepsilon,\ q' \notin Q_{\mathrm{bad}}\} D_{n-1}(g, s', q') \\ & + \mathbf{1}\{\ell \in \Sigma,\ q' \notin Q_{\mathrm{bad}}\} \rho^*_{n-1}(s', q') \Big], \end{aligned} \tag{16}$$

and

$$A_n(g, s, q) := \sum_{s' \in \mathcal{S}} P_g(s' \mid s) \left[ \mathbf{1}\{q' \in Q_{\mathrm{bad}}\} + \mathbf{1}\{q' \notin Q_{\mathrm{bad}}\} \rho^*_{n-1}(s', q') \right]. \tag{17}$$

The tables are filled in increasing order of $n$. Given the layer at $n-1$, equation 16 computes $D_n$; equation 7 then gives $\rho^*_n$ and $g^*_n$; and equation 17 computes $A_n$. This recursive order constructs all values needed by the supervisor without enumerating histories. Each layer sums over at most $d_P$ successors for every $(g, s, q)$, giving the time and storage bounds in Proposition 2.

## B.3 Autonomous-risk dynamic program

For the no-supervision endpoint and the depth-$H$ baseline, let $\mu^0$ select allow at every remaining step and define

$$W_n(g, s, m, q) := \mathbb{P}^{\mu^0}(\exists j \leq n : q_{t+j} \in Q_{\mathrm{bad}} \mid \widetilde{g}_t = g, s_t = s, m_t = m, q_t = q)\,. \tag{18}$$

It equals one when $q \in Q_{\mathrm{bad}}$ and zero at a safe state with $n = 0$. For a safe state and $n \geq 1$, let $\ell = L(s, s')$, $q' = \bar{\delta}(q, \ell)$, and $m' = \tau(m, \ell)$. First-step conditioning gives

$$\begin{aligned} W_n(g, s, m, q) = \sum_{s'} P_g(s' \mid s) \Big[ & \mathbf{1}\{q' \in Q_{\mathrm{bad}}\} + \mathbf{1}\{q' \notin Q_{\mathrm{bad}}\} \\ & \cdot \sum_{g'} \kappa(g' \mid g, m', \ell) W_{n-1}(g', s', m', q') \Big]. \end{aligned} \tag{19}$$

Starting from the boundary layer and applying equation 19 recursively yields the exact autonomous endpoint $p_{\mathrm{aut}} := \sum_g b_0(g) W_N(g, s_0, m_0, q_0)$. The full table takes $O(N|\mathcal{G}|^2|\mathcal{S}||\mathcal{M}||\mathcal{Q}|d_P)$ arithmetic operations and $O(N|\mathcal{G}||\mathcal{S}||\mathcal{M}||\mathcal{Q}|)$ storage because it indexes the scheduler state and averages over its next-goal distribution at event successors.

# C Proofs of the Main Results

This section follows the order of the theoretical analysis. It first proves the filtering and continuation-value propositions, then establishes the support and ordering properties used by Algorithm 1, and finally proves the risk-account theorem and its algorithmic corollary.

## C.1 Exact posterior and information state

*Proof of Proposition 1.* The claim holds at $t = 0$ by the condition on $b_0$. Assume it holds at time $t$. The derivation in Appendix B.1 shows that equation 6 is Bayes' rule for the realized transition followed by propagation through $\kappa$. Hence $b_{t+1}$ is the posterior distribution of $g_{t+1}$ given $h_{t+1}$.

Given $(x_t, u_t)$, the predictive distribution of $s_{t+1}$ is $\sum_g b_t^u(g) P_g(\cdot \mid s_t)$. Once $s_{t+1}$ is observed, the label is $\ell_{t+1} = L(s_t, s_{t+1})$, and equations 3, 2, and 6 determine the next monitor state, scheduler state, posterior, and remaining horizon. The controlled transition of $x_t$ therefore depends on the earlier history only through $(x_t, u_t)$. □

## C.2 Exact continuation values and computation

*Proof of Proposition 2.* We use induction on the remaining horizon, with the boundary values in Appendix B.2. For $n \geq 1$, condition on the successor $s'$ produced by the goal $g$ in force. A bad successor contributes one. Under a one-step segment every safe successor continues with $\rho^*_{n-1}(s', q')$, giving equation 17. Under a segment held until the next event, a safe silent successor retains $g$ and continues with $D_{n-1}(g, s', q')$, while a safe event ends the segment and continues with $\rho^*_{n-1}(s', q')$, giving equation 16. The same recursion applies to an autonomous window because the agent's goal persists across silent transitions. Minimizing $D_n$ over assignable goals at each event boundary gives equation 7, and backward induction proves optimality within the held-takeover policy class.

Each layer evaluates $A_n$ and $D_n$ for every $(g, s, q)$ and sums over at most $d_P$ successors. Computing the minimum in equation 7 has no larger order. Neither table indexes the scheduler state because a held goal is unchanged by it. This proves the stated time and storage bounds, and each online posterior average requires one sum over $\mathcal{G}$. □

## C.3 Posterior support and continuation ordering

For a distribution $b$ over $\mathcal{G}$, write $\operatorname{supp}(b) := \{g \in \mathcal{G} : b(g) > 0\}$.

**Lemma 1** (Posterior support)**.** *Under Assumptions 1 and 2 and the conditions of Proposition 1,* $\operatorname{supp}(b_t) \subseteq \mathcal{G}_{\mathrm{cmd}}(s_t)$ *for every* $t$.

*Proof of Lemma 1.* We argue by induction on $t$. The claim holds at $t = 0$ by Assumption 2(iii). Suppose $\operatorname{supp}(b_t) \subseteq \mathcal{G}_{\mathrm{cmd}}(s_t)$. The executed-goal distribution $b_t^u$ is supported on $\operatorname{supp}(b_t)$ under allow and on $\{u_t\} \subseteq \mathcal{G}_{\mathrm{cmd}}(s_t)$ under an assignment. If $\ell_{t+1} = \varepsilon$, the scheduler kernel is the identity, so by equation 6 every $g \in \operatorname{supp}(b_{t+1})$ satisfies $g \in \operatorname{supp}(b_t^u)$ and $P_g(s_{t+1} \mid s_t) > 0$, and Assumption 2(ii) gives $g \in \mathcal{G}_{\mathrm{cmd}}(s_{t+1})$. If $\ell_{t+1} \in \Sigma$, every $g' \in \operatorname{supp}(b_{t+1})$ lies in $\operatorname{supp} \kappa(\cdot \mid g, m_{t+1}, \ell_{t+1})$ for some $g$ with $P_g(s_{t+1} \mid s_t) > 0$, and Assumption 2(iii) gives $g' \in \mathcal{G}_{\mathrm{cmd}}(s_{t+1})$. □

**Lemma 2** (Ordering of the continuation values)**.** *Let Assumption 2 hold. For every* $n \geq 1$*, every safe pair* $(s, q)$*, and every* $g \in \mathcal{G}_{\mathrm{cmd}}(s)$*,*

$$A_n(g, s, q) \leq D_n(g, s, q) \leq W_n(g, s, m, q) \quad \text{for every } m \in \mathcal{M}. \tag{20}$$

*Consequently, for every information state* $x$ *with* $\operatorname{supp}(b) \subseteq \mathcal{G}_{\mathrm{cmd}}(s)$*,*

$$\rho^*_n(s, q) \leq \overline{D}_n(x), \quad \overline{A}_n(x) \leq \overline{D}_n(x), \quad \Delta^{\mathrm{step}}_n(x) \leq \Delta^{\mathrm{win}}_n(x), \tag{21}$$

*and* $A_n$*,* $D_n$*, and* $\rho^*_n$ *are nondecreasing in* $n$*.*

*Proof of Lemma 2.* Fix $n \geq 1$, a safe pair $(s, q)$, and $g \in \mathcal{G}_{\mathrm{cmd}}(s)$. Subtracting equation 17 from equation 16 gives

$$D_n(g, s, q) - A_n(g, s, q) = \sum_{s' \in \mathcal{S}} P_g(s' \mid s)\, \mathbf{1}\{\ell = \varepsilon,\ q' \notin Q_{\mathrm{bad}}\} \big[ D_{n-1}(g, s', q') - \rho^*_{n-1}(s', q') \big].$$

For every $s'$ in this sum, Assumption 2(ii) gives $g \in \mathcal{G}_{\mathrm{cmd}}(s')$, so $D_{n-1}(g, s', q') \geq \min_{g' \in \mathcal{G}_{\mathrm{cmd}}(s')} D_{n-1}(g', s', q') = \rho^*_{n-1}(s', q')$ and the difference is nonnegative.

For the second inequality we show $W_n(g, s, m, q) \geq D_n(g, s, q)$ for every safe $(s, q)$, $m \in \mathcal{M}$, and $g \in \mathcal{G}_{\mathrm{cmd}}(s)$ by induction on $n$. Both sides vanish at $n = 0$. Comparing equation 19 with equation 16 term by term, a bad successor contributes one to both. A safe silent successor contributes $W_{n-1}(g, s', m, q')$ to the former and $D_{n-1}(g, s', q')$ to the latter. Here $\tau(m, \varepsilon) = m$, and $\kappa$ is the identity on silence. Assumption 2(ii) keeps $g$ commandable at $s'$, so the induction hypothesis orders the two terms. A safe successor with $\ell \in \Sigma$ contributes $\sum_{g'} \kappa(g' \mid g, m', \ell) W_{n-1}(g', s', m', q')$ to the former and $\rho^*_{n-1}(s', q')$ to the latter. By the induction hypothesis and Assumption 2(iii), every $g'$ in the support of $\kappa$ satisfies $W_{n-1}(g', s', m', q') \geq D_{n-1}(g', s', q') \geq \rho^*_{n-1}(s', q')$.

Averaging equation 20 under a posterior supported in $\mathcal{G}_{\mathrm{cmd}}(s)$ gives $\overline{A}_n(x) \leq \overline{D}_n(x)$, and $\overline{D}_n(x) \geq \rho^*_n(s, q)$ because each $D_n(g, s, q)$ with $g \in \mathcal{G}_{\mathrm{cmd}}(s)$ is at least the minimum in equation 7. Subtracting $\rho^*_n$ orders the added risks. Monotonicity in $n$ follows by induction from the recursions: $D_0 = \rho^*_0 = 0$ on safe pairs, and if $D_{n-1} \leq D_n$ and $\rho^*_{n-1} \leq \rho^*_n$ pointwise, then every term of equation 16, equation 17, and equation 7 at horizon $n$ is bounded by the corresponding term at horizon $n + 1$. □

### C.4 General risk-account construction

The argument extends to a finite family $\mathcal{Z}$ of reference continuations. A family is closed under continuation if, after one transition under any member $z$, its remaining policy is another member, denoted $\mathrm{next}(z, h_{t+1})$. References are restricted to histories at which their prescribed inputs satisfy the interface and holding constraints. For the exact conditional risk $C(z; h_t) := \mathbb{P}^z(\mathcal{V}_{0:N} \mid h_t)$, first-step conditioning gives

$$C(z; h_t) = \mathbb{E}^z[C(\mathrm{next}(z, h_{t+1}); h_{t+1}) \,|\, h_t]\,. \tag{22}$$

The supervisor chooses a feasible member $z_t$ using its observed history and executes its next input. Starting from a reference $z_0^-$, set $z_{t+1}^- := \mathrm{next}(z_t, h_{t+1})$. The account is

$$\begin{aligned} C_t^- &:= C(z_t^-; h_t), \qquad C_t^+ := C(z_t; h_t), \\ \chi_t &:= C_t^+ - C_t^-, \quad B_{t+1} := B_t - \chi_t, \quad B_0 := \alpha - C_0^-. \end{aligned} \tag{23}$$

Equation 22 implies equation 12. The window, step, and held-takeover continuations of Algorithm 1 form one such family.

### C.5 Risk guarantee for the supervisor

*Proof of Theorem 1.* Let $\mathcal{F}_t$ be the sigma-field generated by $h_t$ and the supervisor's internal randomization up to the decision at time $t$, so that $z_t$, $C_t^-$, $C_t^+$, and $B_t$ are $\mathcal{F}_t$-measurable. Define $M_t := C_t^- + B_t$. By equation 23,

$$M_{t+1} - M_t = C_{t+1}^- - C_t^- - \chi_t = C(\mathrm{next}(z_t, h_{t+1}); h_{t+1}) - C(z_t; h_t).$$

The transition from $h_t$ to $h_{t+1}$ is generated by the input that $z_t$ prescribes at $t$, so equation 22 applied to $z = z_t$ gives $\mathbb{E}[M_{t+1} - M_t \mid \mathcal{F}_t] = 0$. Hence $M$ is a bounded martingale and $\mathbb{E}[M_N] = M_0 = C_0^- + B_0 = \alpha$. At the horizon no step remains, so $C(z; h_N) = \mathbf{1}\{q_N \in Q_{\mathrm{bad}}\}$ for every $z$, and absorption of $Q_{\mathrm{bad}}$ gives $\mathbf{1}\{q_N \in Q_{\mathrm{bad}}\} = \mathbf{1}\{\mathcal{V}_{0:N}\}$. Therefore $\mathbb{P}(\mathcal{V}_{0:N}) + \mathbb{E}[B_N] = \alpha$, and $B_N = B_0 - \sum_t \chi_t$ gives the first identity in equation 13. If $\alpha \geq C_0^-$ and $\chi_t \leq B_t$ for every $t$, then $B_0 \geq 0$ and $B_{t+1} = B_t - \chi_t \geq 0$ by induction, and $\mathbb{P}(\mathcal{V}_{0:N}) = \alpha - \mathbb{E}[B_N] \leq \alpha$.

The martingale property also gives the online identity

$$\mathbb{P}(\mathcal{V}_{0:N} \mid \mathcal{F}_t) + \mathbb{E}[B_N \mid \mathcal{F}_t] = C_t^- + B_t,$$

whose right-hand side bounds the conditional closed-loop violation probability when the remaining transfers are admissible. □

*Proof of Corollary 1.* Use the references WINDOW, STEP, and TAKEOVER$(g)$ for each assignable or already-held goal $g$. After a safe silent transition, a window retains autonomy. After a safe event,

it begins optimal held takeover. A step begins optimal held takeover after its transition. A takeover retains $g$ after silence and selects a new optimal goal after an event. After violation, all values are one. Assumption 2 keeps these continuations feasible, so the reference family is closed.

Proposition 2 gives $C(\text{TAKEOVER}(g); h_t) = D_{n_t}(g, s_t, q_t)$. Averaging under the exact posterior gives $C(\text{WINDOW}; h_t) = \overline{D}_{n_t}(x_t)$ and $C(\text{STEP}; h_t) = \overline{A}_{n_t}(x_t)$. After a silent autonomous transition, Bayes' rule supplies the posterior weights for the remaining window, establishing equation 12 for that mode.

The algorithm starts from optimal held takeover, so $C_0^- = \rho_N^*(s_0, q_0)$. An inherited takeover and a continuing window have zero transfer. At a fresh boundary, selecting takeover also costs zero and the autonomous transfers equal their added risks. Inside a window, switching to a step transfers $\overline{A}_{n_t}(x_t) - \overline{D}_{n_t}(x_t) \leq 0$, while switching to takeover transfers $\rho_{n_t}^*(s_t, q_t) - \overline{D}_{n_t}(x_t) \leq 0$ by Lemma 2. Every positive autonomous transfer admitted by the algorithm satisfies $\chi_t \leq B_t$; the other cases preserve or increase the balance. Theorem 1 therefore applies. □

Together, the exact filter, continuation-value dynamic program, support and ordering lemmas, and reference-family construction establish all claims used in the main risk guarantee.

# D Additional Theoretical Results

This section collects results that characterize the comparison with local thresholds, the intervention cost of hidden intent, structural cases in which takeover becomes risk-free, and the effect of model error.

## D.1 Local-Threshold Guarantees

Repeated local risk checks accumulate exposure over an episode. The depth-$H$ rule in Appendix E.2 permits autonomy when its predicted violation probability over the next $H$ steps is at most $\zeta$. The results below give the horizon scaling required for a uniform guarantee and account for restarted checks.

**Proposition 3** (The guaranteed level must shrink with the horizon)**.** *For every $\zeta \in (0, 1)$, every $H \geq 1$, and every $N \geq H$ there is an instance satisfying Assumptions 1 and 2 with $\rho_N^*(s_0, q_0) = 0$ on which the rule equation 37 with lookahead $H$ and level $\zeta$ never intervenes before the first violation and*

$$\mathbb{P}(\mathcal{V}_{0:N}) = 1 - (1 - \zeta)^{\lfloor N/H \rfloor} \geq (1 - e^{-1}) \min\{1, \zeta \lfloor N/H \rfloor\}. \tag{24}$$

*Consequently, for $0 < \alpha < 1$, any level $\zeta$ that meets tolerance $\alpha$ on every instance must satisfy*

$$\zeta \leq 1 - (1 - \alpha)^{1/\lfloor N/H \rfloor} \leq \frac{\alpha}{(1 - \alpha)\lfloor N/H \rfloor}.$$

*For $\alpha$ bounded away from one, the required level therefore scales as $O(\alpha H/N)$.*

**Proposition 4** (Episode bound for exact-lookahead thresholds)**.** *Partition an episode into segments of at most $H$ steps. At each segment start $\tau_j$, select and follow a reference continuation whose conditional violation probability over the segment is $r_j$; any earlier continuation change starts a new segment. Let $R$ count reached segments, $K$ count early restarts, and $\mathcal{E}_\zeta := \{r_j > \zeta \text{ for some reached segment } j\}$. Then*

$$\mathbb{P}(\mathcal{V}_{0:N}) \leq \zeta\, \mathbb{E}[R] + \mathbb{P}(\mathcal{E}_\zeta) \leq \zeta\, (\lceil N/H \rceil + \mathbb{E}[K]) + \mathbb{P}(\mathcal{E}_\zeta). \tag{25}$$

*If $\mathbb{P}(\mathcal{E}_\zeta) = 0$, $K \leq K_{\max}$ almost surely, and $\zeta \leq \alpha/(\lceil N/H \rceil + K_{\max})$, then $\mathbb{P}(\mathcal{V}_{0:N}) \leq \alpha$.*

*Proof.* Let $V_j$ be the event that the first violation occurs in segment $j$, and let $J_j$ indicate that the segment is reached. Whenever segment $j$ is reached on $\mathcal{E}_\zeta^c$, its conditional risk is at most $\zeta$, so $\mathbb{P}(V_j \cap \mathcal{E}_\zeta^c) \leq \zeta \mathbb{E}[J_j]$. Disjointness of the $V_j$ gives $\mathbb{P}(\mathcal{V}_{0:N}) \leq \mathbb{P}(\mathcal{E}_\zeta) + \zeta \mathbb{E}[R]$. Each early restart splits one length-$H$ segment, so $R \leq \lceil N/H \rceil + K$ pathwise. □

*Proof of Proposition 3.* Take goals $\mathcal{G} = \{g_r, g_s\}$ and the specification $\mathbf{G}\neg p$. Under $g_r$, the process traverses an $H$-state silent cycle and, at the end of each cycle, emits $p$ with probability $\zeta$ or starts

another cycle with probability $1-\zeta$. Under $g_s$, it enters a silent safe state. Let $s_0$ start the cycle, $b_0(g_r)=1$, and make both goals assignable everywhere, so $\rho^*_N(s_0,q_0)=0$.

At every safe state of the cycle with at least $H$ remaining steps, the exact $H$-step autonomous risk is $\zeta$, because the lookahead window from any such state covers exactly one emission opportunity; with fewer steps it is at most $\zeta$. The threshold rule therefore remains autonomous until violation. Surviving all $\lfloor N/H \rfloor$ cycles has probability $(1-\zeta)^{\lfloor N/H \rfloor}$, proving the equality. The lower bound follows from $1-(1-\zeta)^k \geq 1-e^{-\zeta k}$ and $1-e^{-x} \geq (1-e^{-1})\min\{1,x\}$. Solving for $\zeta$ and using $1-(1-\alpha)^{1/k} \leq -\ln(1-\alpha)/k \leq \alpha/((1-\alpha)k)$ gives the necessary level. □

### D.2 Cost of Intent Ambiguity

Intervention also depends on how quickly the observations reveal the agent's goal. Consider two goals with $b_0(S)=b_0(R)=\frac{1}{2}$ (for a general prior, the leading factor $\frac{1}{2}$ in equation 26 becomes $\min\{b_0(S),b_0(R)\}$ and the term $2\alpha$ in equation 27 becomes $\alpha/b_0(R)$). Under a safe goal $S$, autonomous execution stays outside $Q_{\mathrm{bad}}$ through the horizon. Under a risky goal $R$, failure to assign a goal by step $n$ causes a violation almost surely, for some $1 \leq n < N$. Let $\tau_{\mathrm{cmd}} := \inf\{t : u_t \in \mathcal{G}\}$ be the first assignment time and let $\mathcal{I}^{\mathrm{cmd}}_N := \{\tau_{\mathrm{cmd}} < N\}$ be the event that the supervisor intervenes. Let $\mathbb{P}^{1:n}_S$ and $\mathbb{P}^{1:n}_R$ denote the laws of the first $n$ observations under autonomous execution, conditional on the initial goals $S$ and $R$, including the scheduler's goal updates, and let $\mathbb{P}_S$ and $\mathbb{P}_R$ denote the closed-loop laws under the supervisor conditional on $g_0=S$ and $g_0=R$. Assume that, for some $\bar{d} \geq 0$,

$$D_{\mathrm{KL}}\big(\mathbb{P}^{1:n}_R \,\|\, \mathbb{P}^{1:n}_S\big) \leq n\bar{d}.$$

**Theorem 2** (Intent ambiguity forces intervention)**.** *In the two-goal instance above, every supervisor satisfies*

$$\mathbb{P}(\mathcal{V}_{0:N}) + \mathbb{P}(\mathcal{I}^{\mathrm{cmd}}_N) \geq \frac{1}{2}\max\Big\{1-\sqrt{n\bar{d}/2},\ \tfrac{1}{2}e^{-n\bar{d}}\Big\}. \tag{26}$$

*Consequently, if $\mathbb{P}(\mathcal{V}_{0:N}) \leq \alpha$, then under the safe goal the supervisor issues an assignment within the first $n$ steps with probability*

$$\mathbb{P}_S(\tau_{\mathrm{cmd}} \leq n) \geq \max\Big\{1-\sqrt{n\bar{d}/2},\ \tfrac{1}{2}e^{-n\bar{d}}\Big\} - 2\alpha. \tag{27}$$

Before its first assignment, the supervisor applies one decision rule to a prefix generated by the autonomous process. Assigning by step $n$ is therefore a test between the two goal hypotheses. The proof bounds its error probabilities using the divergence between the prefix laws (Tsybakov & Zaiats, 2009, Section 2.4). The theorem gives the required intervention rate under a safe goal when the safe and risky observation laws are close. The intent ablation in Section 7 evaluates how the goal estimate affects violation risk and intervention.

*Proof.* Before the first assignment, the autonomous process generates the observation prefix. The event $T := \{\tau_{\mathrm{cmd}} \leq n\}$ is therefore one measurable decision rule on the assignment-free prefix and the supervisor's internal randomization under both goal laws. Le Cam's two-point inequality gives

$$\mathbb{P}_S(T) + \mathbb{P}_R(T^c) \geq 1 - \mathrm{TV}(\mathbb{P}^{1:n}_S, \mathbb{P}^{1:n}_R),$$

and Pinsker's inequality together with the Bretagnolle–Huber inequality (Tsybakov & Zaiats, 2009, Section 2.4) bounds the distance by

$$\mathrm{TV}(\mathbb{P}^{1:n}_S, \mathbb{P}^{1:n}_R) \leq \min\Big\{\sqrt{n\bar{d}/2},\ 1-\tfrac{1}{2}e^{-n\bar{d}}\Big\}.$$

Under the equal prior, $T$ under the safe goal contributes at least $\mathbb{P}_S(T)/2$ to $\mathbb{P}(\mathcal{I}^{\mathrm{cmd}}_N)$ because $n < N$. The event $T^c$ under the risky goal contributes at least $\mathbb{P}_R(T^c)/2$ to $\mathbb{P}(\mathcal{V}_{0:N})$. Combining the displays proves equation 26.

If $\mathbb{P}(\mathcal{V}_{0:N}) \leq \alpha$, then $\mathbb{P}_R(T^c) \leq 2\alpha$, and substitution into the testing inequality gives equation 27. □

### D.3 Risk-Free Takeover and Reduced Risk Scores

This result identifies when held takeover removes all remaining risk and the window value reduces to a first-event or immediate-label score.

For a fixed goal $g$, call $\mathcal{Z}_g \subseteq \mathcal{S} \times (\mathcal{Q} \setminus Q_{\mathrm{bad}})$ silently invariant under $g$ if, for every $(s,q) \in \mathcal{Z}_g$,

$$P_g(s' \mid s) > 0 \quad \Longrightarrow \quad L(s,s') = \varepsilon \text{ and } (s', \bar{\delta}(q,\varepsilon)) \in \mathcal{Z}_g. \tag{28}$$

**Proposition 5** (Risk-free takeover and first-event reduction)**.** *If $(s,q) \in \mathcal{Z}_g$ and $g \in \mathcal{G}_{\mathrm{cmd}}(s)$, then $D_n(g,s,q) = \rho_n^*(s,q) = 0$ for every $n$. The same conclusion holds from any pair $(s,q)$ for which some $g \in \mathcal{G}_{\mathrm{cmd}}(s)$, held without reassignment, reaches $\mathcal{Z}_g$ almost surely through safe silent transitions.*

*Fix a goal $g$ held by the agent, and let $\tau_\Sigma := \inf\{j \geq 1 : \ell_{t+j} \in \Sigma\}$ be the first event in the corresponding autonomous window. Suppose every preceding silent prefix remains outside $Q_{\mathrm{bad}}$ and every safe outcome of the first event at time $j \leq n$ reaches a pair with $\rho_{n-j}^* = 0$. Then*

$$D_n(g,s,q) = \mathbb{P}^{\mathrm{win}}_{g,s,q}\left(\tau_\Sigma \leq n,\ q_{t+\tau_\Sigma} \in Q_{\mathrm{bad}}\right). \tag{29}$$

*Here $\mathbb{P}^{\mathrm{win}}_{g,s,q}$ denotes the trajectory law of this autonomous window. For an event-order monitor with $\bar{\delta}(\cdot,\varepsilon)$ equal to the identity, if $\tau_\Sigma \leq n$ almost surely and its label is almost surely a fixed $\sigma_g \in \Sigma$, then*

$$D_n(g,s,q) = \mathbf{1}\{\bar{\delta}(q,\sigma_g) \in Q_{\mathrm{bad}}\}. \tag{30}$$

*If these first-event conditions hold for every goal in the support of $b$, with corresponding labels $\{\sigma_g\}$, and the current takeover value is zero, then the added risk of the window becomes*

$$\Delta_n^{\mathrm{win}}(x) = \sum_{g : \bar{\delta}(q,\sigma_g) \in Q_{\mathrm{bad}}} b(g), \tag{31}$$

*the posterior mass on immediately forbidden goals.*

*Proof.* For $(s,q) \in \mathcal{Z}_g$, hold $g$. Every successor remains in $\mathcal{Z}_g$ and outside $Q_{\mathrm{bad}}$, so induction on $n$ gives $D_n(g,s,q) = 0$, and minimizing over assignable goals gives $\rho_n^*(s,q) = 0$. The reachability extension follows from holding the same goal along the safe silent prefix and inside $\mathcal{Z}_g$.

For the autonomous window, decompose trajectories at $\tau_\Sigma$. A silent prefix contributes no violation by assumption, a safe first event contributes the takeover value after that event, which is zero, and a first event that enters $Q_{\mathrm{bad}}$ contributes one, which proves equation 29. A deterministic first label gives equation 30, and averaging under the posterior and subtracting a zero takeover value gives equation 31. □

In the evaluation pairs of Section 7, the initial takeover risk $\rho_N^*(s_0,q_0)$ is zero on 34 of the 97 pairs and below $10^{-3}$ on 76 of them, while every retained pair has an autonomous-risk gap of at least .15.

### D.4 Robustness to Model Error

The following result transfers the model-based guarantee to a deployed process whose kernels are close in total variation.

Suppose the supervisor runs Algorithm 1 with a model $(\widehat{P}_g, \widehat{\kappa}, \widehat{b}_0)$ of the deployed process $(P_g, \kappa, b_0)$, with the labeling map and the monitor known exactly, and let

$$\begin{aligned} \varsigma_P &:= \max_{g \in \mathcal{G},\, s \in \mathcal{S}} \mathrm{TV}\big(\widehat{P}_g(\cdot \mid s), P_g(\cdot \mid s)\big), \\ \varsigma_\kappa &:= \max_{g,m,\ell} \mathrm{TV}\big(\widehat{\kappa}(\cdot \mid g,m,\ell), \kappa(\cdot \mid g,m,\ell)\big), \\ \varsigma_0 &:= \mathrm{TV}(\widehat{b}_0, b_0). \end{aligned} \tag{32}$$

**Proposition 6** (Tolerance transfer under model error)**.** *Let $\widehat{\mathbb{P}}$ be the law of the closed loop when the deployed process follows the model and $\mathbb{P}$ its law under the deployed process, with the same supervisor in both. Then every event $E$ determined by the trajectory satisfies*

$$\left|\widehat{\mathbb{P}}(E) - \mathbb{P}(E)\right| \leq \varsigma_0 + N(\varsigma_P + \varsigma_\kappa). \tag{33}$$

*In particular, if the supervisor is run with tolerance* $\alpha_{\mathrm{model}} \geq \widehat{\rho}^{*}_{N}(s_0, q_0)$*, then* $\mathbb{P}(\mathcal{V}_{0:N}) \leq \alpha_{\mathrm{model}} + \varsigma_0 + N(\varsigma_P + \varsigma_\kappa)$*, so running it at* $\alpha_{\mathrm{model}} = \alpha - \varsigma_0 - N(\varsigma_P + \varsigma_\kappa)$ *meets the deployed tolerance* $\alpha$ *whenever this value is at least* $\widehat{\rho}^{*}_{N}(s_0, q_0)$*.*

*Proof.* Construct the two closed loops on one probability space, including the latent goals. Draw the initial goals from a maximal coupling of $\widehat{b}_0$ and $b_0$, which differ with probability $\varsigma_0$. Conditional on equal histories, the supervisor sees the same observations and issues the same input in both processes. At each step, draw $(s_{t+1}, g_{t+1})$ from a maximal coupling of the joint kernels $\widehat{P}_{\widetilde{g}_t}(s' \mid s_t)\widehat{\kappa}(g' \mid \widetilde{g}_t, m', \ell)$ and $P_{\widetilde{g}_t}(s' \mid s_t)\kappa(g' \mid \widetilde{g}_t, m', \ell)$, where $\ell = L(s_t, s')$ and $m' = \tau(m_t, \ell)$. Their total variation distance is at most $\varsigma_P + \varsigma_\kappa$. The two trajectories coincide until the first disagreement, so each trajectory event has the same indicator up to that time. A union bound over the initial draw and the $N$ transitions bounds the disagreement probability by $\varsigma_0 + N(\varsigma_P + \varsigma_\kappa)$. This also bounds the difference between the two probabilities of $E$. The guarantee under the model follows from Theorem 1 applied by Algorithm 1. □

The margin grows linearly with the horizon because the coupling is renewed at every step.

# E Supplementary Experimental Setup

This section specifies the constraint families, baseline decision rules, main evaluation protocol, and controlled scaling protocol used in Section 7. The formulas and decision rules come first, followed by the sampling, tuning, and hardware settings needed to reproduce the reported results.

## E.1 Constraint Families

The experiments use one deterministic bad-prefix monitor for each LTLf specification. The generated domains assign disjoint role propositions to task zones, and a transition enters at most one zone. Each event therefore carries one goal identity and at most one of the role propositions $p$, $i$, $r$, and $c$; additional zones are role-neutral. Let $E$ denote the disjunction of all atomic propositions, which holds exactly at a non-silent event. We write $\mathbf{X}_s\psi := \mathbf{X}\psi$ for the next physical transition and $\mathbf{X}_e\psi := \mathbf{X}((\neg E)\,\mathbf{U}\,(E \wedge \psi))$ for the next non-silent event. These abbreviations express both clocks in standard LTLf over the physical-step word before the specification is compiled as described in Sections 3 and 4.2. For an integer $h \geq 1$, $\mathbf{F}^{s}_{[1,h]}\psi := \bigvee_{j=1}^{h} \mathbf{X}_s^{j}\psi$ denotes satisfaction within the next $h$ physical transitions. Section 4.2 defines violation as reaching $Q_{\mathrm{bad}}$ during the $N$-step evaluation.

The response bound $d \in [8, 20]$ is 1.02–1.18 times the length of a detour from the $r$ zone through a third zone to the $c$ zone, and it exceeds the direct $r$–$c$ distance by more than one step, so the direct route fits within the deadline while the detour is sensitive to transition noise. The separation bound is $k = \max\{4, \mathrm{round}(2\bar{\ell})\}$, where $\bar{\ell}$ is the mean shortest-path length between distinct goal regions. On the development and held-out maps, $d \in [10, 20]$ and $k \in [10, 22]$.

**Bounded separation.** $\varphi^{k}_{\mathrm{sep}} := \mathbf{G}\big(p \rightarrow \bigwedge_{j=1}^{k} \mathbf{X}_s^{j}\neg p\big)$. After a $p$ event, another $p$ is forbidden during the next $k$ physical transitions.

**Bounded response.** $\varphi^{d}_{\mathrm{resp}} := \mathbf{G}\big(r \rightarrow \mathbf{F}^{s}_{[1,d+1]}c\big)$. After an $r$ event, at most $d$ subsequent transitions without $c$ may elapse, so $c$ must occur by the following $(d+1)$st transition.

**Precedence chain.** $\varphi_{\mathrm{prec}} := (\neg p)\,\mathbf{W}\,\big(i \wedge \mathbf{X}_e((\neg p)\,\mathbf{W}\,r)\big)$. Events containing $p$ are forbidden before an $i$ event and between that event and a later $r$ event.

**Timing composite.** $\varphi_{\mathrm{time}} := (\neg p\,\mathbf{W}\,i) \wedge \mathbf{G}(r \rightarrow \mathbf{X}_e\neg p) \wedge \mathbf{G}(c \rightarrow \mathbf{X}_e\neg r) \wedge \varphi^{d}_{\mathrm{resp}}$, using the same map-specific response bound $d$.

**Allowance composite.** The formula is

$$\varphi_{\mathrm{allow}} := \mathbf{G}\big(p \rightarrow \mathbf{X}_e(\neg p\,\mathbf{W}\,c)\big) \wedge \mathbf{G}\big(r \rightarrow \mathbf{X}_e(\neg p\,\mathbf{W}\,c)\big) \wedge \mathbf{G}\big(r \rightarrow \mathbf{X}_e[\neg r\,\mathbf{W}\,(i \vee (r \wedge \mathbf{X}_e(\neg r\,\mathbf{W}\,i)))]\big).$$

The three clauses use fixed allowances: at most one $p$ between consecutive $c$ events, no $p$ after $r$ until $c$, and at most two $r$ events between consecutive $i$ events.

### E.2 Baseline Definitions

A score-based supervisor permits autonomy below a threshold and otherwise assigns the safest goal. All baselines receive the same posterior and observations and use the same event-held goal interface as our method. Table 5 summarizes their decisions; the exact scores follow below. Here $\zeta$ is a tunable score threshold, $H$ is a positive lookahead horizon, and $\delta_{\mathrm{sh}} \in [0,1]$ is the multiplicative-shield parameter. The reachability and multiplicative rules adapt action-level shielding to assignable goals (Alshiekh et al., 2018; Jansen et al., 2020; Heck et al., 2026).

Table 5: Baseline decision rules.

| Method | Risk summary | Autonomy condition |
|---|---|---|
| Completion label | Next goal-completion label | Immediate score at most $\zeta$ |
| First event | Trajectory through the next event | First-event risk at most $\zeta$ |
| $H$-risk | Next $\min\{H, n_t\}$ steps | Depth-$H$ risk at most $\zeta$ |
| Reachability shield | Positive-probability paths over $H$ steps | No path reaches $Q_{\mathrm{bad}}$ |
| Multiplicative shield | One autonomous step, then takeover | $\delta_{\mathrm{sh}} Q_t \le V_t$ |

The autonomous value $W_n$ and its recursion are defined in Appendix B.3, and the depth-$H$ score truncates that recursion to $\min\{H, n_t\}$. With $\ell = L(s, s')$ and $q' = \bar{\delta}(q, \ell)$, the first-event score terminates after a safe first event. Set $W_n^{\mathrm{fe}}(g, s, q) = 1$ when $q \in Q_{\mathrm{bad}}$ and $W_0^{\mathrm{fe}}(g, s, q) = 0$ when $q \notin Q_{\mathrm{bad}}$; for $n \ge 1$ and $q \notin Q_{\mathrm{bad}}$,

$$W_n^{\mathrm{fe}}(g, s, q) = \sum_{s'} P_g(s' \mid s) \left[ \mathbf{1}\{q' \in Q_{\mathrm{bad}}\} + \mathbf{1}\{\ell = \varepsilon, q' \notin Q_{\mathrm{bad}}\} W_{n-1}^{\mathrm{fe}}(g, s', q') \right]. \tag{34}$$

For the completion-label map $\gamma : \mathcal{G} \to \Sigma$, where $\gamma(g)$ is the event emitted when goal $g$ is completed, the immediate score is

$$W^{\mathrm{imm}}(g, q) := \mathbf{1}\{q \in Q_{\mathrm{bad}} \text{ or } \bar{\delta}(q, \gamma(g)) \in Q_{\mathrm{bad}}\}. \tag{35}$$

Write

$$\begin{aligned} \widehat{W}^{H}(g, x_t) &:= W_{\min\{H, n_t\}}(g, s_t, m_t, q_t), \quad \widehat{W}^{\mathrm{fe}}(g, x_t) := W_{n_t}^{\mathrm{fe}}(g, s_t, q_t), \\ \widehat{W}^{\mathrm{imm}}(g, x_t) &:= W^{\mathrm{imm}}(g, q_t). \end{aligned} \tag{36}$$

Given one of these scores, define $\widehat{\rho}_t(u) := \sum_g b_t^u(g) \widehat{W}(g, x_t)$. Completion label, First event, and $H$-risk use

$$u_t = \begin{cases} \mathsf{allow}, & \widehat{\rho}_t(\mathsf{allow}) \le \zeta, \\ g_{n_t}^*(s_t, q_t), & \widehat{\rho}_t(\mathsf{allow}) > \zeta, \end{cases} \tag{37}$$

and hold an assigned goal until the next event. The reachability shield blocks autonomy if, within $\min\{H, n_t\}$ steps, some positive-probability path of a goal in the posterior support reaches $Q_{\mathrm{bad}}$ during the autonomous window, or reaches an event after which no sequence of goal assignments avoids $Q_{\mathrm{bad}}$ within the remaining lookahead.

For the multiplicative shield, let $Q_t := \overline{A}_{n_t}(x_t)$ and $V_t := \min\{Q_t, \rho_{n_t}^*(s_t, q_t)\}$. Adapting probabilistic shielding to goal-level choices (Jansen et al., 2020; Heck et al., 2026), it permits autonomy when

$$\delta_{\mathrm{sh}} Q_t \le V_t, \tag{38}$$

and otherwise assigns a goal attaining $\rho_{n_t}^*$ until the next event.

### E.3 Reproducibility Protocol

Table 6 collects the main-benchmark choices needed for reproduction. All tunable parameters are selected using development maps and frozen before evaluation. Section E.4 gives the separate controlled-scaling protocol.

Table 6: Main-benchmark reproducibility protocol. Counts are per method and parameter setting unless stated otherwise.

| Component | Setting |
|---|---|
| Domain | Randomized grids with side lengths from 9 to 11 and four or five task zones; randomized walls, zone locations, initial states, intent priors, and slip probability in $[.05, .15]$ |
| Task process | The scheduler retains its goal through silence and events at other zones. When the active goal is completed, it samples uniformly from the other goals, which may be revisited later. The command interface excludes only the goal of the currently occupied zone, so Assumption 2 holds by construction |
| Constraints and horizon | Five families in Appendix E.1; $N = 120$, with transfer evaluation at $N = 200$ |
| Development | 12 maps; parameters selected separately for each requested risk level |
| Evaluation | 20 maps paired with five formulas; 300 shared episode seeds per pair |
| Random seeds | Development maps 0–11; held-out maps 500–519; tuning episodes 1200–1239; held-out episodes 5000–5299 |
| Risk settings | $\beta \in \{.05, .10, .20, .30, .50, .75\}$ for frontiers; frozen-tolerance evaluation at $\beta = .25$ |
| Score thresholds | $\zeta \in \{-10^{-9}, 0, .01, .025, .05, .1, .2, .3, .4, .5\}$ $\cup\{.6, .7, .75, .8, .85, .9, .95, .975, 1\}$; the negative sentinel never admits a zero score |
| Lookahead and shield grids | $H \in \{4, 8, 16, 32\}$ for $H$-risk, $H \in \{2, 4, 8, 16, 32\}$ for reachability; $\delta_{\mathrm{sh}} \in \{0, .1, .2, .3, .4, .5, .6, .7, .8, .9, .95, 1\}$ |
| Spending caps | $\eta \in \{.1, .2, .3, .5, .75, 1, 2, 4\}$ for our method; allocation ablations also include 1.5 and 3 |
| Retention | $p_{\mathrm{aut}} - \rho^*_N \geq .15$; 97 pairs: 19 each for separation, timing, and precedence, and 20 each for response and allowance |
| Uncertainty | 5,000 paired bootstrap draws for frontiers; standard errors across pairs; Wilson lower bounds for detected tolerance exceedance |
| Timing | Apple M5 Pro; online time includes filtering, risk-table lookup, and the supervision decision |

### E.4 Controlled Scaling Protocol

This study varies the physical state space and number of task areas at a fixed monitor size. For $m \in \{4, 8\}$, let $e_i \in \mathsf{AP}$ be the proposition emitted when goal $g_i$ is completed. Define request and service propositions by

$$R_m := \bigvee_{i=1}^{m/2} e_i, \quad C_m := \bigvee_{i=m/2+1}^{3m/4} e_i, \quad \varphi_m := \mathbf{G}\left(R_m \rightarrow \bigvee_{j=1}^{3} \mathbf{X}_e^j C_m\right). \tag{39}$$

Thus $(R_4, C_4) = (e_1 \vee e_2, e_3)$ and $(R_8, C_8) = (e_1 \vee \cdots \vee e_4, e_5 \vee e_6)$; the remaining goals are neutral. A request is violated when the next three events contain no service. An obligation remains pending when fewer than three later events occur before the observation window ends. Both complete bad-prefix DFAs have five states, so the comparison varies the number of goals at a fixed monitor size.

We generate square grids of side length $L \in \{10, 20, 30, 40, 50\}$ with random walls, one-cell task areas, initial states, and intent priors. Nominal skill paths avoid non-target task areas, and every transition has slip probability .1. We use five maps and 100 common episode seeds per configuration. Parameters are selected on four independent $10 \times 10$ development maps with 60 episodes and then fixed. The requested normalized risk is $\beta = .25$. The supervision study uses $N = 12L$ to preserve a comparable number of possible inter-area trips as travel distance grows. The computation study fixes $N = 120$ to isolate growth in $|\mathcal{S}|$. Figure 2 shows retained instances at the two ends of the scale range.

## F Supplementary Experimental Results

This section reports controlled scaling, benchmark diagnostics, allocation ablations, and an initial-prior mismatch stress test.

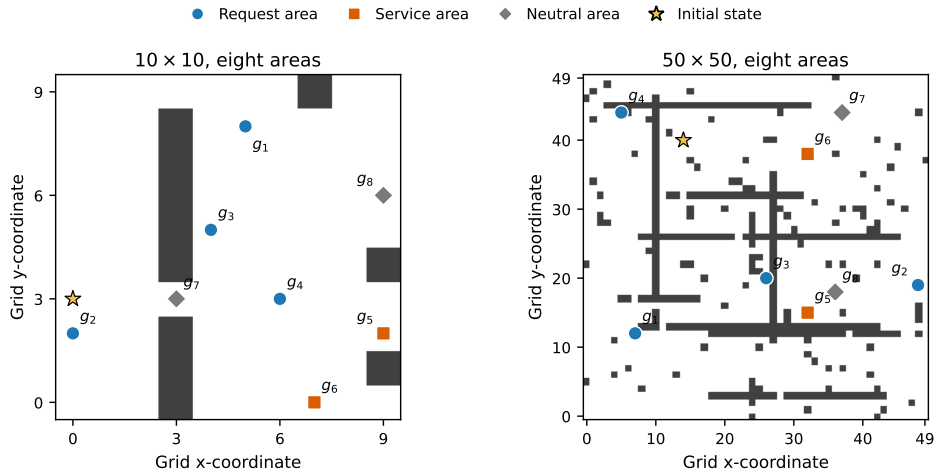



Figure 2: Retained $10 \times 10$ and $50 \times 50$ scaling instances with eight task areas (seeds 830000 and 870000). The specification treats $g_1, \ldots, g_4$ as requests, $g_5, g_6$ as services, and $g_7, g_8$ as neutral events.

### F.1 Supervision Across Grid and Goal Scale

Table 7 reports normalized risk and intervention at the common requested tolerance $\beta = .25$. Our method has no detected exceedance and the lowest intervention in all ten configurations. It reduces aggregate intervention by $34\%$ relative to $H$-risk, which also has no detected exceedance, and by $81\%$ relative to the multiplicative shield, which exceeds the tolerance on 2 of 50 map–configuration pairs.

Table 7: Supervision under controlled grid and goal scaling. Entries are mean ± one standard error across five maps; lower intervention is better. Parameters are selected on independent $10 \times 10$ maps and frozen.

| | | Ours | | $H$-risk | | Multiplicative shield | |
|---|---|---|---|---|---|---|---|
| Areas | Side | Risk | Intervention | Risk | Intervention | Risk | Intervention |
| 4 | 10 | .143±.034 | **.053±.010** | .022±.009 | .099±.022 | .253±.166 | .213±.063 |
| 4 | 20 | .058±.026 | **.053±.007** | .006±.004 | .085±.015 | .000±.000 | .331±.029 |
| 4 | 30 | .106±.018 | **.070±.010** | .028±.014 | .110±.016 | .000±.000 | .290±.021 |
| 4 | 40 | .103±.024 | **.050±.003** | .076±.020 | .068±.016 | .000±.000 | .354±.017 |
| 4 | 50 | .069±.017 | **.047±.004** | .029±.006 | .057±.006 | .000±.000 | .352±.020 |
| 8 | 10 | .187±.026 | **.074±.012** | .037±.010 | .106±.014 | .000±.000 | .393±.016 |
| 8 | 20 | .124±.017 | **.083±.010** | .023±.007 | .119±.019 | .029±.026 | .341±.015 |
| 8 | 30 | .126±.021 | **.066±.008** | .048±.009 | .098±.015 | .000±.000 | .343±.019 |
| 8 | 40 | .131±.025 | **.067±.004** | .071±.014 | .112±.016 | .000±.000 | .348±.017 |
| 8 | 50 | .085±.029 | **.055±.005** | .089±.026 | .076±.011 | .000±.000 | .336±.019 |

### F.2 Computational Scaling

Table 8 fixes $N = 120$ and disables risk-gap screening. Against realized $|\mathcal{S}|$, the log-log construction-time slopes are .95 and .97 for four and eight goals, and both storage slopes are 1.00, consistent with Proposition 2. The complete online update stays below .02 ms per step. At the longest supervision horizon, the $50 \times 50$ grid with eight goals and $N = 600$ builds in 3.101 seconds and occupies 427.7 MB.

**Main-benchmark computation.** Table 9 reports the family-level cost of constructing and using the risk tables. Monitor size drives most of the variation in offline storage and construction time. The online update takes at most .014 ms per physical step.

Table 8: Computation under controlled scaling. Build time and storage are median [25th, 75th percentiles] over five maps at $N = 120$; online time is mean $\pm$ one standard error over the $N = 12L$ rollouts.

| Areas | Side | $\lvert\mathcal{S}\rvert$ | $\lvert\mathcal{Q}\rvert$ | Build (s) | Table (MB) | Online (ms/step) |
|---|---|---|---|---|---|---|
| 4 | 10 | 92 | 5 | .015 [.015, .015] | 1.9 | .0134±.0001 |
| 4 | 20 | 357 | 5 | .052 [.052, .054] | 7.4 | .0134±.0001 |
| 4 | 30 | 794 | 5 | .113 [.112, .114] | 16.5 | .0134±.0001 |
| 4 | 40 | 1434 | 5 | .200 [.198, .209] | 29.8 | .0133±.0000 |
| 4 | 50 | 2235 | 5 | .315 [.314, .318] | 46.4 | .0135±.0001 |
| 8 | 10 | 92 | 5 | .028 [.027, .028] | 3.6 | .0187±.0001 |
| 8 | 20 | 347 | 5 | .103 [.098, .110] | 13.6 | .0182±.0001 |
| 8 | 30 | 841 | 5 | .236 [.235, .237] | 33.0 | .0184±.0001 |
| 8 | 40 | 1438 | 5 | .398 [.398, .423] | 56.4 | .0182±.0001 |
| 8 | 50 | 2195 | 5 | .614 [.606, .632] | 86.1 | .0186±.0001 |

Table 9: Computation over retained pairs, reported as median [25th, 75th percentiles]. Online time is per physical step.

| Family | $\lvert\mathcal{Q}\rvert$ | Build (s) | Table (MB) | Online (ms) |
|---|---|---|---|---|
| Bounded separation | 17 | .046 [.039, .049] | 7.2 [6.0, 7.7] | .013 [.012, .013] |
| Timing composite | 486 | 1.196 [1.005, 1.381] | 203.2 [185.6, 243.1] | .014 [.013, .014] |
| Precedence chain | 4 | .015 [.014, .016] | 1.7 [1.6, 2.0] | .013 [.012, .013] |
| Bounded response | 18 | .050 [.044, .058] | 7.6 [6.9, 8.9] | .013 [.012, .014] |
| Allowance composite | 36 | .092 [.086, .109] | 15.0 [14.0, 17.8] | .014 [.013, .014] |

### F.3 Additional Diagnostics

Figure 3 expands the matched-risk comparison in Table 1: our intervention remains below the baseline envelope at all six comparison points. Figure 4 expands Table 2 by showing every method before and after the horizon is extended without retuning. Our method retains zero detected pairwise exceedances at both horizons, while every comparator has a nonzero excess-pair fraction.

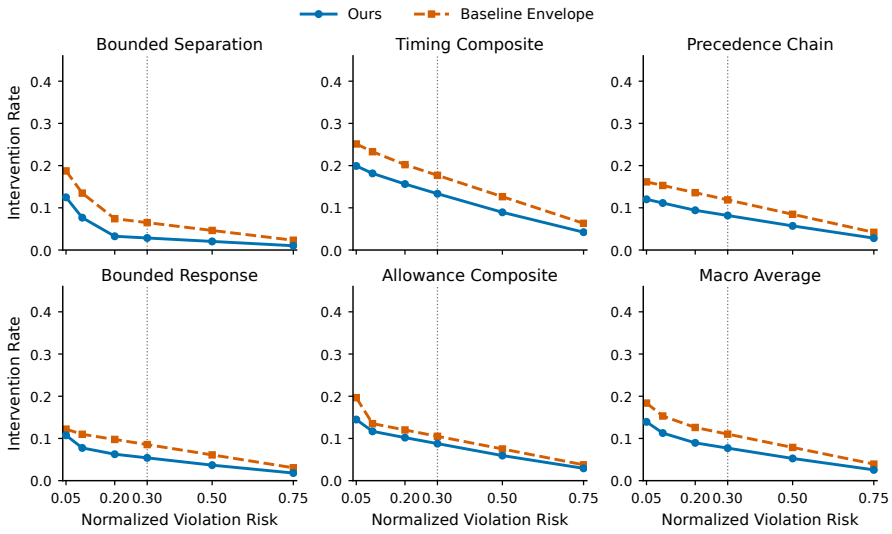



Figure 3: Risk–intervention frontiers from rollouts and reference endpoints. The baseline envelope pools all comparator settings; dotted lines mark the comparison point in Table 1.

Numerical checks give Bellman residuals at most $1.22 \times 10^{-7}$, decision-transfer and pathwise-budget residuals at machine precision, and an accounting residual within $2.2 \times 10^{-3}$ of zero across

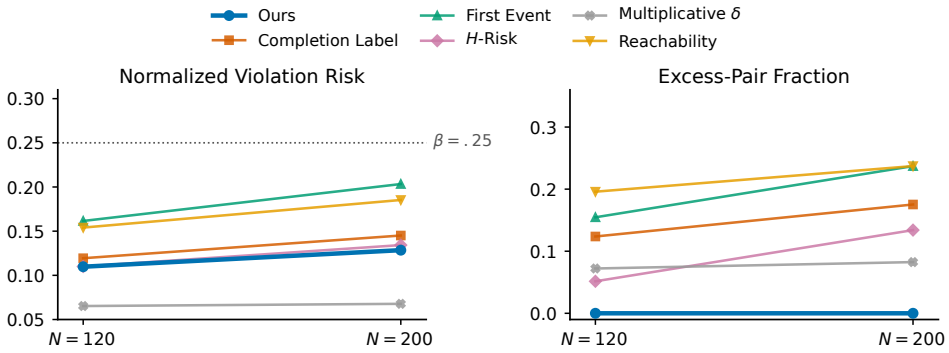



Figure 4: Horizon transfer with parameters frozen at $N = 120$. Left: normalized risk. Right: fraction of pairs with a detected tolerance exceedance.

eight spending-cap settings. The residual is the macro average, over pairs within each family and then over families, of $\widehat{\mathbb{P}}(\mathcal{V}_{0:N}^{(j)}) + \overline{B}_{N,j} - \alpha_j$, where $\widehat{\mathbb{P}}(\mathcal{V}_{0:N}^{(j)})$ is the empirical violation frequency for pair $j$ and $\overline{B}_{N,j}$ is its mean terminal balance.

### F.4 Adversarial Prior-Mismatch Stress Test

We test whether a known total-variation bound on initial-prior error can be converted into a tolerance correction. The supervisor uses the nominal prior $\widehat{b}_0$, while the deployed prior moves probability mass toward the initial goal with the largest held-window risk $D_N(g, s_0, q_0)$. For each pair, the moved mass is an exact total-variation radius $\varsigma_0 = f(\alpha - \rho_N^*)$, with $f \in \{0, .25, .5, .75, .9\}$. The skill kernels, scheduler, labeling map, monitor, spending cap, and episode seeds remain fixed. This stress test uses independent episode seeds 7000–7299. Table 10 compares supervision at the original tolerance $\alpha$ and the corrected model tolerance $\alpha_{\text{model}} = \alpha - \varsigma_0$. The full-model tolerance-transfer rule is derived in Appendix D.4 (Proposition 6).

Table 10: Initial-prior mismatch over 97 held-out pairs. Violation is the unnormalized empirical violation probability; violation and intervention are macro means over 300 episodes per pair. Excess counts pairs whose 95% Wilson lower bound exceeds the deployed tolerance.

| | | Nominal tolerance | | | TV-corrected tolerance | | |
|---|---|---|---|---|---|---|---|
| $f$ | Mean $\varsigma_0$ | Violation | Intervention | Excess | Violation | Intervention | Excess |
| 0 | .000 | .091 | .107 | 0/97 | .091 | .107 | 0/97 |
| .25 | .047 | .097 | .107 | 3/97 | .066 | .118 | 0/97 |
| .50 | .093 | .102 | .107 | 6/97 | .043 | .139 | 0/97 |
| .75 | .140 | .107 | .107 | 6/97 | .024 | .164 | 0/97 |
| .90 | .168 | .111 | .107 | 7/97 | .015 | .184 | 0/97 |

The nominal supervisor does not add intervention as the mismatch grows, and detected exceedances appear. The corrected tolerance restores zero detected exceedances at every tested radius with a controlled increase in intervention.

### F.5 Allocation and Spending-Cap Ablations

Table 11 compares the full rule with three variants. Whole-window allocation keeps within-window re-evaluation but removes the one-step fallback; window-boundary allocation also removes within-window re-evaluation and decides only after events. The no-refund variant keeps the full admission choices but does not credit decreases in risk. Each uses its development-selected minimum-intervention setting at the same requested tolerance. The full rule and window-boundary allocation both attain normalized risk .110, while the full rule reduces intervention by 65%. The no-refund variant intervenes at .109 and replaces the accounting identity with an upper bound.

Table 11: Allocation ablations over 120 episodes per pair. Entries are macro averages ± one standard error. Terminal budget is in unnormalized probability units.

| Rule | Risk | Intervention | Goal changes | Terminal budget |
|---|---|---|---|---|
| Ours | .110±.006 | .105±.005 | 1.92±.10 | .101±.004 |
| Whole window | .047±.005 | .308±.018 | 5.04±.46 | .157±.004 |
| No refund | .097±.006 | .109±.005 | 2.02±.12 | .065±.003 |
| Window boundary | .110±.008 | .296±.017 | 4.46±.41 | .108±.006 |

None of the four rules has a detected tolerance exceedance on the 97 held-out pairs under the 95% Wilson criterion.

**Spending-cap sensitivity.** The parameter $\eta$ in equation 11 limits the added risk of each autonomous choice; the remaining-balance test still applies. At the fixed requested tolerance $\beta = .25$, increasing $\eta$ from .1 to 4 raises achieved normalized risk from .023 to .155 and lowers intervention from .152 to .103. None of the eight tested settings has a detected tolerance exceedance.