Zero-Violation Regret for Cooperative Markov Games with Coupled Instantaneous Hard Constraints

Jialei Liu **Ming Shi**
University at Buffalo
{jialeili,mshi24}@buffalo.edu

Abstract

Learning in cooperative multi-agent systems must preserve safety constraints that couple agents' actions. We study episodic cooperative Markov games with unknown rewards, transitions, and instantaneous constraint functions, where agents execute actions using local observations and a prescribed communication graph. Our algorithm, SAFE-CMG-UCB, composes known and learned local safety certificates and optimizes an optimistic value estimate while retaining a known safe fallback. We prove high-probability zero violation and a regret bound separating estimation error from episodes with uncertified optimal actions. Sublinear regret follows under exact implementable planning, a positive margin for initially uncertified tuples of nonfallback comparator actions, and coverage by safely obtained observations. Local certification reuses information across joint actions, and a constructive distributed protocol quantifies computation, state acquisition, and routed communication. Bounded width of the complete planning graph permits efficient action optimization, but insufficient communication can reduce optimal safe value even at width one. Synthetic experiments examine these benefits and the limitations imposed by uninformative safety feedback and dense coupling.

1 Introduction

Robots sharing a workspace and transmitters sharing a channel must coordinate their actions to avoid collisions or excessive interference [1, 2]. Reinforcement learning (RL) can improve their decisions from interaction [3], but collecting data must also be safe. An expected cumulative-cost budget can permit an unsafe step even when the total cost satisfies the budget in expectation. This distinction matters when a single collision or interference event causes harm. We study *instantaneous hard constraints*, which require every executed joint action to satisfy every constraint throughout learning. The objective is high-probability safety and return approaching that of the best admissible safe policy.

Coupled constraints make both learning and execution difficult. Two robot motions that are safe in isolation can collide when executed together. Likewise, several individually permissible transmissions can jointly exceed a receiver's interference limit. Certifying each agent's action independently is therefore insufficient. An agent may also participate in several constraints, and its action must satisfy all of them simultaneously. Which joint actions can be certified from safely collected observations, and how can agents agree on one before execution? Answering these questions requires both informative safety observations and enough communication to coordinate the agents' actions.

We study this problem in a cooperative Markov game with unknown reward, transition, and constraint models. Each constraint depends on the states and actions of a known group of agents. Training estimates the models centrally, and execution uses local states and messages exchange over a communication graph. Agents coordinate before acting simultaneously. The comparator is the optimal safe protocol under the same communication restrictions.

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

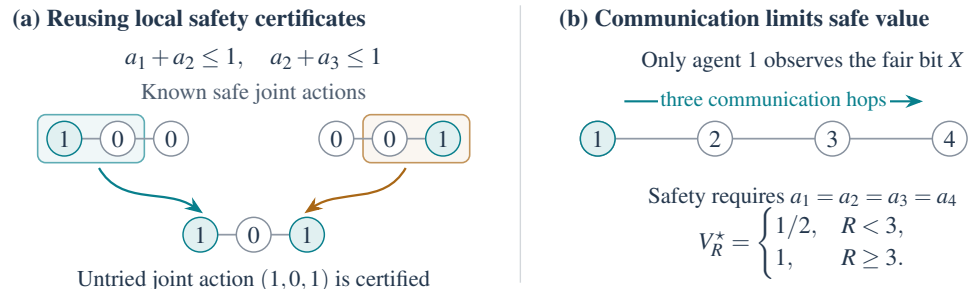


Figure 1: Two structural effects. (a) At a fixed state, known safe actions $(1,0,0)$ and $(0,0,1)$ certify the highlighted pairs. Combining these certificates gives $(1,0,1)$ without executing it. (b) On the four-agent path, safety requires equal actions and the reward is $\mathbf{1}\{a_1 = X\}$. With R communication rounds, the optimal safe reward $V_R^\star$ increases when X can reach agent 4 (Proposition H.1).

Figure 1 reveals why training and communication are not interchangeable. Local certificates can establish an untried joint action's safety from previously collected evidence. Yet even with known models, agents cannot use observations that have not reached them before execution. Conversely, communication alone cannot resolve safety uncertainty left unidentified by the available data. The resulting design question is which safe observations and messages are sufficient to improve team performance without an unsafe trial. Answering it requires separate guarantees for learning which actions are safe and coordinating their execution. It also motivates jointly choosing safe measurements and communication schedules, beyond the fixed-resource conditions analyzed here.

Single-agent instantaneous-safe RL provides the statistical starting point [4–6]. The common reward and joint dynamics also admit a centralized MDP over joint actions. Factored MDP methods already exploit local structure and recover consistent actions through message passing [7, 8]. An unstructured joint-action table estimates repeated local configurations separately, while a full-state optimizer may use information unavailable to local agents. Local immediate rewards and constraints also do not guarantee efficient planning because future-value terms and exploration bonuses can enlarge the action scopes. Even a factor involving one action may require a distant agent's observation. Statistical complexity, planning complexity, and communication cost therefore require separate bounds.

SAFE-CMG-UCB combines local safety certificates with optimistic planning. A local tuple is admitted when either initial knowledge certifies it or a statistical upper bound establishes its safety. The same certificate can then be reused in every compatible joint action. The planner combines these certificates across overlapping scopes, retaining one shared action variable per agent, and selects an action using optimistic reward and transition estimates [9–11]. A known protocol safe at every modeled state remains available as a whole fallback, ensuring safe continuation after every action. A separate extension certifies continuation when the fallback is available only on a restricted domain.

Our first result establishes high-probability zero violation and a regret bound that separates reward and transition estimation from failure to certify optimal actions. Sublinear regret follows under exact communication-compatible planning, a positive margin for initially uncertified tuples of nonfallback comparator actions, and safely acquired observations that reduce uncertainty in the relevant constraint directions. To show why this information condition is essential, we construct two models that give identical observations under the fallback but disagree on whether a better action is safe. A learner that must remain safe in both models can then suffer linear regret. Thus a known safe policy need not provide the observations required to learn a better one.

Our structural results explain when coordination improves safe learning. We characterize the joint actions certified by combining local parts of known safe actions and bound tabular certification failures by local configuration counts. One example changes linear regret to constant regret when the specified learner reuses these local certificates. A second example has treewidth one, yet insufficient communication prevents the team from attaining the full-state optimum. We then construct a junction-tree protocol that enforces one consistent action across overlapping scopes, including state acquisition and physical message routing. We bound communication rounds and the regret cost of finite-precision score messages. Finally, team regret bounds feasible unilateral-improvement gaps, clarifying the relation to constrained equilibrium objectives.

Synthetic experiments complement the theory (Appendix K). On fixed-width chains with rewards normalized by team size, regret varies little across the tested agent counts, with no observed violations.

2 Related Work

Cumulative and instantaneous constraints. Constrained MDPs optimize reward subject to expected cumulative-cost budgets [12]. Regret analyses control budget violation [13–15], and some methods guarantee that every selected policy satisfies its budget with high probability [16, 17]. These guarantees do not impose per-action thresholds. Expected instantaneous constraints on randomized action distributions likewise differ from actionwise safety [18]. Hard ramp and covering constraints also arise in online optimization with revealed feasibility information [19, 20]. Instantaneous-safe bandit and MDP methods are the closest statistical foundations [4–6, 21, 22]. Appendix B.1 compares their assumptions, safety geometry, and observation models with our $N = 1$ specialization.

Structured cooperation and safety. Factored planning and coordinated RL preserve local interactions and use variable elimination to recover consistent joint actions [7, 8]. We use this established planning machinery to combine initial and learned safety certificates, with explicit costs for certification and distributed implementation. Computational efficiency requires small induced width of the complete score-and-constraint graph. Shield decentralization starts from a supplied shield and sound environment abstraction [23]. Other approaches use constrained policy optimization or adaptive safety mechanisms [24, 25]. Differential-game approximations [26] and multiplayer reach–avoid analysis [27] address related control problems for specified models. Our analysis concerns learning unknown models while preserving instantaneous feasibility and controlling regret.

Equilibria and information structure. Constrained Markov games study feasible unilateral deviations and constrained equilibria [28–33]. Shared constraints determine which unilateral policy changes are feasible, so an equilibrium criterion requires a deviation class as well as a reward objective. In our common-reward protocol class, team regret bounds feasible unilateral-improvement gaps, while unilateral stability can coexist with a suboptimal team return (Corollary 5.7). Hierarchical-information cooperation [34] and Stackelberg learning [35, 36] impose distinct information or strategic response structures. Our elimination order controls computation and message organization while preserving the objective and feasible set. It introduces no strategic leader (Appendix B).

3 Problem Formulation

Notation. Write $[n] = \{1, \dots, n\}$, $[x]_+ = \max\{x, 0\}$, and $\Delta(\mathcal{X})$ for the probability simplex on a finite set $\mathcal{X}$. For $I \subseteq [N]$, $s_I = (s_i)_{i \in I}$ and $\mathbf{a}_I = (a_i)_{i \in I}$. For a positive definite matrix M, $\|x\|_M = (x^\top M x)^{1/2}$ is the induced norm. The symbol I_d denotes the d-dimensional identity. Matrix inequalities use the positive semidefinite order. Unmarked logarithms are natural.

3.1 Cooperative dynamics and hard safety

There are $N \geq 1$ agents with finite local state spaces $\mathcal{S}_i$, nonempty global state space $\mathcal{S} \subseteq \prod_i \mathcal{S}_i$, finite nonempty action sets $\mathcal{A}_i$ and their joint-action space $\mathcal{A} = \prod_i \mathcal{A}_i$, horizon $H \geq 1$, initial distribution $\rho \in \Delta(\mathcal{S})$, transition kernels $\{P_h\}_{h=1}^H$, and mean team rewards $r_h : \mathcal{S} \times \mathcal{A} \to [0, 1]$. Over $K \geq 1$ episodes, $s_1^k \sim \rho$ independently of history, and joint action $\mathbf{a}_h^k$ yields $s_{h+1}^k \sim P_h(\cdot \mid s_h^k, \mathbf{a}_h^k)$. Total interactions are $T = KH$, and maximum local action size is $A_{\max} = \max_i |\mathcal{A}_i|$.

There are $m \geq 1$ constraints. For each $\ell \in [m]$, a known scope $I_\ell \subseteq [N]$ determines an unknown deterministic constraint function $g_{\ell,h}(s_{I_\ell}, \mathbf{a}_{I_\ell})$. The safe joint-action set is

$$\mathcal{A}_h^{\mathrm{safe}}(s) = \{\mathbf{a} \in \mathcal{A} : g_{\ell,h}(s_{I_\ell}, \mathbf{a}_{I_\ell}) \leq 0 \text{ for every } \ell \in [m]\}. \tag{1}$$

An agent in overlapping scopes must use the same action in every constraint involving it. Consequently, the intersection in (1) need not be a product of individual safe-action sets. Safety concerns the latent constraint values, so measurement noise does not itself constitute a violation. Every executed action, including those used to collect training data, must satisfy these constraints.

3.2 Communication-compatible protocols

Agent i observes s_i. Before each environment action, agents communicate over an undirected graph $G_{\mathrm{comm}} = ([N], E_{\mathrm{comm}})$ for a supplied integer budget $R_h \geq 0$ of synchronous rounds. A step-h protocol consists of message maps and final action maps,

$$z_{i\to j}^{t} = \mu_{i\to j,h}^{t}(s_i, \mathsf{M}_i^{t-1}), \qquad a_i = \pi_{i,h}(s_i, \mathsf{M}_i^{R_h}), \tag{2}$$

where $z_{i\to j}^{t}$ is the message from i to j in round $t \in [R_h]$, M_i^t is agent i's transcript through round t, M_i^0 is empty, $a_i \in \mathcal{A}_i$, and $(i,j) \in E_{\mathrm{comm}}$. Centralized training installs local model and message programs before each episode. Agents evaluate them using their realized local states. All agents act simultaneously after communication completes. Communication is reliable, the state remains fixed during these rounds, and messages and delays do not enter the reward or horizon. The exact construction permits an arbitrary finite vector per directed edge per round. Section 4.3 accounts separately for message volume and precision.

Let $\mathfrak{P}$ be the permitted class of deterministic Markov protocols. Unless otherwise specified, it contains all local message and action maps satisfying the graph, message alphabet, and round budgets. A more restricted class must still satisfy Assumption 3.3. Tie rules are fixed before observing the current state. We write $\Pi_h(s)$ for the joint action induced by running the distributed protocol at s, although each agent uses only its own observation and received messages. A protocol is safe if every constraint holds almost surely along its trajectories from ρ. Its value is

$$V_h^{\Pi}(s) = \mathbb{E}_{\Pi}\left[\sum\nolimits_{t=h}^{H} r_t(s_t, \mathbf{a}_t) \mid s_h = s\right], \qquad V_1^{\Pi}(\rho) = \mathbb{E}_{s\sim\rho}[V_1^{\Pi}(s)]. \tag{3}$$

The expectation starts from state s at step h, including states unreachable from ρ. For any state function v, write $v(\rho) = \sum_s \rho(s)v(s)$. Let $\mathfrak{P}_{\mathrm{safe}}$ denote the safe protocols and choose an optimal safe comparator $\Pi^\star \in \arg\max_{\Pi\in\mathfrak{P}_{\mathrm{safe}}} V_1^{\Pi}(\rho)$. The interaction history before episode k is $\mathcal{H}_{k-1}$, and Π^k is chosen using this history. Cumulative regret and positive constraint violation are

$$\mathrm{Reg}(K) = \sum\nolimits_{k=1}^{K}\left[V_1^{\Pi^\star}(\rho) - V_1^{\Pi^k}(\rho)\right], \qquad \mathrm{Viol}(K) = \sum\nolimits_{k,h,\ell}[g_{\ell,h}(s_{I_\ell,h}^k, \mathbf{a}_{I_\ell,h}^k)]_+. \tag{4}$$

For a failure probability $\delta \in (0,1)$, the goal is $\mathrm{Viol}(K) = 0$ with probability at least $1-\delta$ and sublinear regret under the stated information conditions.

3.3 Statistical model and observations

Let $\mathcal{E}_r$ be a nonempty family of reward scopes $e \subseteq [N]$, and let $r_{e,h}$ be the mean of reward factor e. Reward and constraint functions have local features. Transitions follow a linear-mixture model that may couple all agents. These known features permit sharing observations across inputs [10, 11], as specified by

$$r_h(s,\mathbf{a}) = \sum\nolimits_{e\in\mathcal{E}_r} r_{e,h}(s_e, \mathbf{a}_e), \quad r_{e,h}(s_e,\mathbf{a}_e) = \langle \phi_{e,h}^r(s_e,\mathbf{a}_e), \theta_{e,h}^r\rangle, \tag{5}$$

$$g_{\ell,h}(s_{I_\ell}, \mathbf{a}_{I_\ell}) = \langle \phi_{\ell,h}^g(s_{I_\ell}, \mathbf{a}_{I_\ell}), \theta_{\ell,h}^g\rangle, \tag{6}$$

$$P_hV(s,\mathbf{a}) = \langle \psi_h^V(s,\mathbf{a}), \theta_h^P\rangle, \qquad \psi_h^V(s,\mathbf{a}) = \sum\nolimits_{s'\in\mathcal{S}} \phi_h^P(s' \mid s,\mathbf{a})V(s'). \tag{7}$$

Here $P_hV(s,\mathbf{a}) := \sum_{s'} P_h(s' \mid s,\mathbf{a})V(s')$ is the expected next-state value. All feature maps and scopes are known, while the parameters are unknown. The transition parameter belongs to a known set Θ_h^P for which $P_\theta(s' \mid s,\mathbf{a}) = \langle \phi_h^P(s' \mid s,\mathbf{a}), \theta\rangle$ is a probability kernel. The same θ_h^P applies to every V.

Assumption 3.1 (Bounded linear model and feedback)**.** Local features have Euclidean norm at most one, and $\|\theta_{e,h}^r\|_2 \leq B_r$, $\|\theta_{\ell,h}^g\|_2 \leq B_g$, and $\|\theta_h^P\|_2 \leq B_P$, for supplied nonnegative bounds. Also $\|\psi_h^V(s,\mathbf{a})\|_2 \leq \|V\|_\infty$ for every $V : \mathcal{S} \to \mathbb{R}$. At each executed step the learner observes the next state, each local reward-factor measurement $Y_{r,e,h}$, and each constraint measurement $Y_{g,\ell,h}$. Their conditional means are the corresponding local reward and constraint values. The centered noises are conditionally R_r- and R_g-sub-Gaussian for supplied nonnegative noise bounds. Any auxiliary constraint observation has response $Y = \langle x, \theta_{\ell,h}^g\rangle + \varepsilon$, where x is a predictable feature with $\|x\|_2 \leq 1$ and ε is conditionally R_g-sub-Gaussian noise. It is obtained without executing an uncertified action.

Let $F = |\mathcal{E}_r|$ count reward factors and let $d_{r,e} = \dim(\phi^r_{e,h})$, $d_g = \dim(\phi^g_{\ell,h})$, and $d_P = \dim(\phi^P_h)$ be the reward, constraint, and transition feature dimensions. Zero-padding gives positive stage-independent dimensions and a common d_g. The total reward dimension is $D_r = \sum_e d_{r,e}$. Separate reward regressions require factor observations. Appendix G.1 gives an aggregate-reward variant. Let $n_g \geq K$ bound all observations per (ℓ, h) through episode K. Auxiliary acquisition costs are separate from environment regret.

Assumption 3.2 (Available safe fallback)**.** A known protocol $\Pi^0 \in \mathfrak{P}$ induces $\mathbf{a}^0_h(s)$ satisfying $g_{\ell,h}(s_{I_\ell}, \mathbf{a}^0_{I_\ell,h}(s)) \leq 0$ for every $s \in \mathcal{S}$, $h \in [H]$, and $\ell \in [m]$.

The learner may additionally use fixed, sound local certificates

$$\mathcal{Z}_\ell = \{(s_{I_\ell}, \mathbf{a}_{I_\ell}) : (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}\}, \quad \mathcal{Z}^0_{\ell,h} \subseteq \{z \in \mathcal{Z}_\ell : g_{\ell,h}(z) \leq 0\}. \tag{8}$$

Here $\mathcal{Z}_\ell$ contains attainable local tuples and $\mathcal{Z}^0_{\ell,h}$ is its initially certified subset, possibly empty. Projecting known safe joint state–action pairs, including fallback pairs, onto the constraint scope provides valid certificates. Each certificate applies to every global configuration containing that tuple, enabling the recombination in Figure 1(a), including boundary-safe tuples. Explicit sets or supplied membership tests represent these certificates, with a separate preprocessing cost.

Exploration can reach states outside the fallback's own trajectories. Assumption 3.2 therefore covers every modeled state without requiring a strict margin. Appendix F studies continuation certificates when this global fallback condition is unavailable.

Assumption 3.3 (Exact, communication-compatible planning)**.** For each episode, the selected maximizers of the unclipped score (16) over (15), using the specified tie rule, are realizable jointly for all states by one protocol $\Pi^k \in \mathfrak{P}$, including the comparison with Π^0. The planner returns this protocol and its exact clipped values (17).

Separate statewise optimizers need not define one admissible policy when agents cannot distinguish states requiring different actions. Section 4.3 provides a construction with evaluable factors, bounded induced width, and sufficient communication for state acquisition and message passing. Linear realizability does not ensure these conditions.

3.4 Examples and structural limits

In discretized robot coordination, minimum next-position separation couples robot controls, while receiver interference limits couple transmitter powers. Chain interactions can have small planning width, but all-to-all pairwise collision checks induce a complete action graph and network-wide interference can create a global constraint. Efficient planning also requires suitable Bellman and bonus scopes. Appendix E.5 gives explicit constraints, sparse and dense variants, and their feature, feedback, and prediction-error requirements.

A communication-dependent safe optimum. Consider a known-model, one-step game on the communication path $1 - \cdots - N$. Only agent 1 observes a fair bit X. The reward is $\mathbf{1}\{a_1 = X\}$, and pairwise constraints force all binary actions to agree. The all-zero fallback is safe with margin one. Despite action-graph treewidth one,

$$V^\star_{\text{full}} = 1, \qquad V^\star_R = 1/2 \ (R < N - 1), \qquad V^\star_R = 1 \ (R \geq N - 1). \tag{9}$$

Here $V^\star_{\text{full}}$ is the full-state optimum and $V^\star_R$ is the optimum over safe R-round protocols. Agent N must agree with agent 1 without knowing X when $R < N - 1$, which proves the value gap. Broadcasting X attains value one when $R \geq N - 1$. The full-state optimizer is implementable only above this threshold, and Assumption 3.3 fails below it. Figure 1(b) shows the four-agent instance, and Proposition H.1 gives the full proof in Appendix H.

4 Safe Learning and Distributed Action Coordination

SAFE-CMG-UCB uses optimistic reward and transition estimates to encourage exploration, while upper confidence bounds on constraints exclude actions whose safety is unresolved. Each episode fits models, plans backward, and executes a fixed distributed policy. Algorithm 1 specifies these phases, Figure 2 relates them to performance and safety, and Figure 3 details the protocol.

4.1 Predictable regression and confidence bounds

Before episode k, fit each local reward and constraint model and one transition model per step using identity-regularized least squares. Write $\Lambda^r_{k,e,h}, \Lambda^g_{k,\ell,h}, \Lambda^P_{k,h}$ for the design matrices, $\widehat{\theta}^r_{k,e,h}, \widehat{\theta}^g_{k,\ell,h}, \widehat{\theta}^P_{k,h}$ for the parameter estimates, and $\beta^r_{k,e,h}, \beta^g_{k,\ell,h}, \beta^P_{k,h}$ for their confidence radii. Appendices A.2 and A.1 specify the regressions and every numerical parameter. The designs record observed feature directions, so informative observations reduce uncertainty.

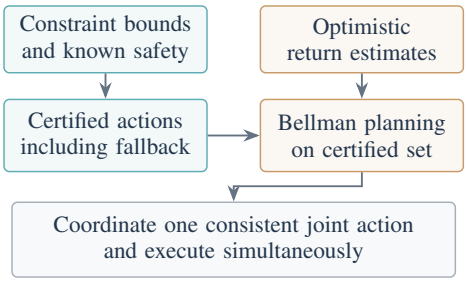


Figure 2: On the confidence event, consistent certified actions ensure safety. Sublinear regret needs sufficient safety information.

With failure allocation $\alpha = \delta/[2H(F+m+1)]$ per regression, the self-normalized inequality [37] gives a simultaneous parameter-confidence event $\mathcal{E}$ of probability at least $1-\delta/2$ (Appendix C.1). Its ellipsoids control predictions uniformly over states, actions, and value functions. Appendix C.6 explains why this regression avoids a value-function covering argument.

Transition regression predicts next-state value. Its normalized feature $x^\tau_{P,h}$ and response $Y^\tau_{P,h}$ retain the value function fixed before the sample was collected,

$$x^\tau_{P,h} = \psi_h^{V_{\tau,h+1}}(s^\tau_h, \mathbf{a}^\tau_h)/H, \qquad Y^\tau_{P,h} = V_{\tau,h+1}(s^\tau_{h+1})/H. \tag{10}$$

Apply (31) for $\tau < k$ to obtain $\Lambda^P_{k,h}$ and $\widehat{\theta}^P_{k,h}$. The feature is measurable before s^τ_{h+1} is observed. Clipping keeps values in $[0,H]$ on every history, and division by H bounds both the response and feature norm by one. Substituting later value estimates need not preserve this predictability.

The estimated mean reward and expected next-state value are $\widehat{r}_{k,h} = \sum_e \langle \phi^r_{e,h}, \widehat{\theta}^r_{k,e,h}\rangle$ and $\widehat{P}_{k,h}V = \langle \psi^V_h, \widehat{\theta}^P_{k,h}\rangle$, with the features' state–action arguments. The estimate $\widehat{P}_{k,h}$ need not be a probability kernel. Define the reward bonus b^r, transition bonus b^P, and constraint upper bound U^g by

$$b^r_{k,h}(s,\mathbf{a}) = \sum\nolimits_{e\in\mathcal{E}_r} \beta^r_{k,e,h}\|\phi^r_{e,h}(s_e,\mathbf{a}_e)\|_{(\Lambda^r_{k,e,h})^{-1}}, \tag{11}$$

$$b^P_{k,h}(s,\mathbf{a};V) = H\beta^P_{k,h}\|\psi^V_h(s,\mathbf{a})/H\|_{(\Lambda^P_{k,h})^{-1}}, \tag{12}$$

$$U^g_{k,\ell,h}(s_{I_\ell},\mathbf{a}_{I_\ell}) = \langle \phi^g_{\ell,h}(s_{I_\ell},\mathbf{a}_{I_\ell}), \widehat{\theta}^g_{k,\ell,h}\rangle + \beta^g_{k,\ell,h}\|\phi^g_{\ell,h}(s_{I_\ell},\mathbf{a}_{I_\ell})\|_{(\Lambda^g_{k,\ell,h})^{-1}}. \tag{13}$$

On $\mathcal{E}$, $U^g_{k,\ell,h} \ge g_{\ell,h}$, $\widehat{r}_{k,h} + b^r_{k,h} \ge r_h$, and $\widehat{P}_{k,h}V + b^P_{k,h}(\cdot;V) \ge P_hV$. An unexplored feature direction increases potential return but also makes its constraint harder to certify. Exploration therefore remains restricted to actions supported by valid safety information.

4.2 Certified action sets and optimistic Bellman backups

The local state–action tuple is $z_\ell(s,\mathbf{a}) = (s_{I_\ell}, \mathbf{a}_{I_\ell})$. Each constraint is certified by initial knowledge or its upper confidence bound. Requiring a valid certificate in every scope gives the immediate certified set $\widehat{\mathcal{A}}^{\mathrm{imm}}_{k,h}(s)$. Retaining the whole fallback yields the candidate set $\widehat{\mathcal{A}}_{k,h}(s)$,

$$\widehat{\mathcal{A}}^{\mathrm{imm}}_{k,h}(s) = \{\mathbf{a} :\ z_\ell(s,\mathbf{a}) \in \mathcal{Z}^0_{\ell,h} \text{ or } U^g_{k,\ell,h}(z_\ell(s,\mathbf{a})) \le 0,\ \text{for every } \ell\}, \tag{14}$$

$$\widehat{\mathcal{A}}_{k,h}(s) = \widehat{\mathcal{A}}^{\mathrm{imm}}_{k,h}(s) \cup \{\mathbf{a}^0_h(s)\}. \tag{15}$$

Every candidate is safe on $\mathcal{E}$, and the fallback makes the set nonempty. Initially certified boundary tuples can be combined with learned certificates. Appendix J.2 gives regret K when the fallback is retained only as an indivisible action and $\min\{K,3\}$ when its local certificates are reused, without auxiliary observations. Initialize $V_{k,H+1} = 0$. The unclipped optimistic score $\widetilde{Q}_{k,h}$, clipped action value $Q_{k,h}$, and state value $V_{k,h}$ satisfy the backward recursion

$$\widetilde{Q}_{k,h}(s,\mathbf{a}) = \widehat{r}_{k,h}(s,\mathbf{a}) + b^r_{k,h}(s,\mathbf{a}) + \widehat{P}_{k,h}V_{k,h+1}(s,\mathbf{a}) + b^P_{k,h}(s,\mathbf{a};V_{k,h+1}), \tag{16}$$

$$Q_{k,h}(s,\mathbf{a}) = \mathrm{clip}_{[0,H-h+1]}\left(\widetilde{Q}_{k,h}(s,\mathbf{a})\right), \qquad V_{k,h}(s) = \max\nolimits_{\mathbf{a}\in\widehat{\mathcal{A}}_{k,h}(s)} Q_{k,h}(s,\mathbf{a}). \tag{17}$$

Algorithm 1 SAFE-CMG-UCB

Input: State and action sets, known features and scopes, bounds B_r, B_g, B_P, R_r, R_g, protocol class $\mathfrak{P}$ with its graph, alphabet, round budgets and tie rule, safe fallback Π^0, optional local certificates $\mathcal{Z}^0_{\ell,h}$, H, K and $\delta \in (0,1)$.

Initialize the regression data sets to be empty.
For $k = 1, \dots, K$:
  Estimate: fit the models and confidence sets from pre-episode data using (31)–(33).
  Set $V_{k,H+1}(s) = 0$ for every $s \in \mathcal{S}$.
  For $h = H, \dots, 1$:
    Certify: combine local certificates and the whole fallback in (15).
    Plan: compute the optimistic scores (16) and values (17). Compile the factor evaluators, routes, and protocol maps (2) for all states, including fallback selection.
  End for
  For $h = 1, \dots, H$:
    Coordinate and act: instantiate factors from local observations and messages. Agree on the certified assignment or whole fallback, then execute $\Pi^k_h(s^k_h)$ simultaneously.
    Observe: store factor rewards, constraint measurements, and the historical-value transition sample (10).
  End for
  Incorporate any safely obtained auxiliary constraint observations permitted by the feedback model.
End for

Here $\mathrm{clip}_{[0,b]}(x) = \max\{0, \min\{b, x\}\}$. Select a maximizer of the unclipped score, preferring the whole fallback on ties and otherwise using fixed action and bag orders. Monotone clipping preserves this maximizer, and Assumption 3.3 makes these choices one admissible protocol.

The globally available fallback ensures safe continuation, so the main algorithm requires only immediate certificates. Replacing this assumption by backward viability tests requires the additional transition-support conditions in Appendix F.

4.3 Exact coordination, communication, and complexity

Suppose the complete score is a known sum of local score factors $q_{k,h,f}$, indexed by $\mathcal{F}^Q_{k,h}$. The score decomposition and constrained planning problem are

$$\begin{aligned} \widetilde{Q}_{k,h}(s, \mathbf{a}) &= \sum\nolimits_{f \in \mathcal{F}^Q_{k,h}} q_{k,h,f}(s_{J_f}, \mathbf{a}_{I_f}), \\ &\max_{\mathbf{a} \in \mathcal{A}} \sum\nolimits_f q_{k,h,f}(s_{J_f}, \mathbf{a}_{I_f}) \quad \text{s.t.} \quad z_\ell \in \mathcal{Z}^0_{\ell,h} \text{ or } U^g_{k,\ell,h}(z_\ell) \le 0 \ \ (\ell \in [m]). \end{aligned} \tag{18}$$

The second expression optimizes over immediate certified actions. Comparing its optimum with the whole fallback implements (15), even if the immediate set is empty. All transition-value and bonus terms are included. Action scopes I_f determine elimination width, while state scopes J_f determine the observations required for factor evaluation.

Exact variable elimination or junction-tree computation solves (18) [7, 8, 38]. Upward messages summarize conditional scores and aggregate the fallback score separately. The root selects the certified optimizer or whole fallback. Downward assignments then fix one value for every shared action variable, ensuring consistency across overlapping constraints. Messages follow routes on G_{comm}, and all agents act after delivery (Appendix E).

Let $\mathcal{F}$ include all score and safety factors, $r_{\max}$ their maximum action-scope size, and ω the induced width of the chosen elimination order on their primal graph. Thus ω is the largest number of remaining neighbors of an eliminated variable after fill edges are added. Constructing the tables takes $O(|\mathcal{F}| A^{r_{\max}}_{\max})$ factor evaluations. For a reduced junction tree $\mathcal{T}$ with bag collection $\mathcal{B}$, exact optimization of these tables requires

$$O\big((|\mathcal{F}| + |\mathcal{B}|) A^{\omega+1}_{\max} + N\big) \tag{19}$$

arithmetic operations per state, plus the fallback evaluation. Reduced junction-tree separator tables have $O(A^{\omega}_{\max})$ scalar entries. The two passes take $O(\mathrm{diam}(\mathcal{T}))$ logical rounds for tree diameter $\mathrm{diam}(\mathcal{T})$, plus routing, state-acquisition, and congestion costs on G_{comm}.

Proposition I.1 compiles this computation into one protocol over all states. Let R_0 be the fallback's round cost, r_{in} the largest required input-routing distance, r_{out} the largest output-routing distance, and D_G the largest sum of physical routing distances on a root-to-leaf junction-tree path. With fixed bag hosts and routes, sufficient rounds are

$$R \geq R_0 + r_{\mathrm{in}} + 2D_G + r_{\mathrm{out}}. \tag{20}$$

This bound includes state acquisition, fallback comparison, assignment, and delivery under the vector-message model. Proposition I.2 also bounds routed bit traffic. Rounding only score factors with quantization step $\zeta > 0$, while keeping safety indicators exact, preserves feasibility and adds at most $|\mathcal{F}^Q_{k,h}|\zeta$ optimization error.

Runtime factor instantiation and message passing evaluate the estimates, certificates, values, and protocol maps fixed before the episode. Proposition E.1 proves equivalence with precomputation under the same tie rule. Tabular planning requires $H|\mathcal{S}|$ state–step evaluations, with separate factor evaluation, storage, and distribution costs. A host may need global state inputs. Dense pairwise scopes or a global Bellman factor can give $\omega = N - 1$ and exponential action cost.

5 Theoretical Guarantees

The largest confidence radii are $\beta^r = \max_{k,e,h} \beta^r_{k,e,h}$, $\beta^P = \max_{k,h} \beta^P_{k,h}$, and $\beta^g = \max_{k,\ell,h} \beta^g_{k,\ell,h}$, with maxima over the first K episodes. The factor $\sqrt{F}$ below comes from summing confidence widths of separately estimated reward factors. Equation (46) gives deterministic bounds on these maxima. The notation $\widetilde{O}$ suppresses logarithmic factors in $K, n_g, 1/\delta$.

Theorem 5.1 (Zero instantaneous violation)**.** *Under Assumptions 3.1 and 3.2, an implementation of* SAFE-CMG-UCB *that executes only actions in* (15) *satisfies every instantaneous constraint with probability at least* $1 - \delta$*. Hence* $\mathrm{Viol}(K) = 0$ *on this event. This conclusion does not require a positive comparator margin or a certification-coverage condition.*

The guarantee holds even when uncertainty prevents certification of the optimal protocol. *Proof sketch.* On the simultaneous confidence event, every statistical upper bound dominates its true constraint. Supplied local certificates and the all-state fallback cover the other candidates. Every executed joint action is therefore safe. The full proof is in Appendix C.2.

Performance also depends on which safe actions can be certified. Let

$$\mathcal{X}^\star_h := \{(s, \mathbf{a}) : \mathbb{P}_{\Pi^\star}(s_h = s, \mathbf{a}_h = \mathbf{a}) > 0\}. \tag{21}$$

This is the support of the comparator's step-h state–action distribution under the true kernels and initial law ρ. Let χ_k indicate an uncertified action in the comparator's reachable support in episode k, and let B_K count such episodes,

$$\chi_k := \mathbf{1}\Big\{\exists h \in [H],\ (s, \mathbf{a}) \in \mathcal{X}^\star_h : \mathbf{a} \notin \widehat{\mathcal{A}}_{k,h}(s)\Big\}, \qquad B_K := \sum\nolimits_{k=1}^{K} \chi_k. \tag{22}$$

Neither $\Pi^\star$, χ_k, nor B_K is an input to the learner.

Theorem 5.2 (Regret with an explicit certification term)**.** *Under Assumptions 3.1–3.3, zero violation and the following regret bound hold jointly with probability at least* $1 - \delta$*,*

$$\begin{aligned}\mathrm{Reg}(K) \leq\ & 2\beta^r \sqrt{2FHD_r T \log(1 + K)} + 2H\beta^P \sqrt{2Hd_P T \log(1 + K/d_P)} \\ & + HB_K + H\sqrt{2T \log(2/\delta)}.\end{aligned} \tag{23}$$

The first two terms measure reward and transition uncertainty, while HB_K measures the cost of excluding the comparator. The final term controls trajectory-sampling error. The bound is sublinear when $B_K = o(K)$ and its estimation terms are sublinear. *Proof sketch.* When $\chi_k = 0$, backward induction establishes comparator optimism. Otherwise the lost value is at most H. Telescoping the executed policy's Bellman errors reduces the remaining regret to confidence widths and trajectory-sampling errors. Stepwise elliptical-potential bounds control the former, and martingale concentration controls the latter. The full proof is in Appendix C.3.

Assumption 5.3 (Comparator margin and certification coverage)**.** There exists $\Delta_{\mathrm{safe}} > 0$ such that, for every $h \in [H]$, $\ell \in [m]$, and $(s, \mathbf{a}) \in \mathcal{X}_h^\star$ with $\mathbf{a} \neq \mathbf{a}_h^0(s)$,

$$z_\ell(s, \mathbf{a}) \notin \mathcal{Z}_{\ell,h}^0 \quad \Longrightarrow \quad g_{\ell,h}(z_\ell(s, \mathbf{a})) \leq -\Delta_{\mathrm{safe}}. \tag{24}$$

Initially certified tuples may lie on the safety boundary. On the confidence event, for every episode with $\chi_k = 1$, there is a witness $(h, \ell, s, \mathbf{a})$ satisfying $(s, \mathbf{a}) \in \mathcal{X}_h^\star$, $\mathbf{a} \notin \widehat{\mathcal{A}}_{k,h}(s)$, $z_\ell(s, \mathbf{a}) \notin \mathcal{Z}_{\ell,h}^0$, and $U_{k,\ell,h}^g(z_\ell(s, \mathbf{a})) > 0$ such that, for $x = \phi_{\ell,h}^g(s_{I_\ell}, \mathbf{a}_{I_\ell})$,

$$\Lambda_{k+1,\ell,h}^g - \Lambda_{k,\ell,h}^g \succeq xx^\top. \tag{25}$$

All observations contributing to this increase are acquired without violating the hard constraints. This condition does not authorize executing an uncertified candidate. The budget n_g counts all measurements per (ℓ, h) through episode K, including safe auxiliary feedback.

In each episode with an excluded comparator action, condition (25) requires safely collected data to provide the specified covariance increase for at least one failed local test. Executed-action feedback or additional safety measurements may supply this information, with auxiliary acquisition costs accounted for separately. Proposition C.8 gives a trajectory-feedback instance with linear regret despite a safe fallback, positive margin, and sparse scopes. The margin and coverage witnesses are analytical conditions, not algorithm inputs.

Corollary 5.4 (Sublinear regret under certification coverage)**.** *Under the assumptions of Theorem 5.2 and Assumption 5.3, on the same event,*

$$B_K \leq 8Hmd_g(\beta^g/\Delta_{\mathrm{safe}})^2 \log(1 + n_g/d_g). \tag{26}$$

Substituting this certification count into Theorem 5.2 gives

$$\mathrm{Reg}(K) \leq \widetilde{O}\Big(\beta^r \sqrt{FHD_rT} + H\beta^P \sqrt{Hd_PT} + H^2 md_g(\beta^g/\Delta_{\mathrm{safe}})^2 + H\sqrt{T}\Big). \tag{27}$$

For fixed H, F, m, dimensions, parameter and noise bounds, $\Delta_{\mathrm{safe}} > 0$, and $\delta \in (0, 1)$, if n_g is polynomial in K, the confidence radii are $O(\sqrt{\log K})$ and the regret is sublinear in K.

The margin sets the confidence-width threshold, while coverage limits how often certification can fail. *Proof sketch.* For each episode counted by B_K, assign one failed test supplied by Assumption 5.3. Its margin implies a confidence half-width exceeding $\Delta_{\mathrm{safe}}/2$, while coverage supplies a corresponding observed design increment. Summing the resulting increases in $\log\det\Lambda$ proves (26). The full proof is in Appendix C.4. Certified approximate planning with state-uniform score error $\varepsilon_{k,h}$ increases regret by at most $\sum_{k,h} \varepsilon_{k,h}$, provided every output remains feasible (Appendix C.5).

Single-agent specialization. For $N = 1$, communication disappears and joint tuples become ordinary state–action pairs. With one reward factor, substituting $F = 1$ and $D_r = d_r$ into (27) gives the rate in Appendix B.1. This specialization retains the all-state fallback, unknown-tuple margin, and feedback coverage. No sharper rate than existing instantaneous-safe MDP results is claimed [4–6]. The appendix compares these assumptions and distinguishes actionwise, expected instantaneous, and cumulative constraints.

Theorem 5.5 (Necessity lower bounds)**.** *The following statements hold.*

1. *For unstructured one-state, one-step cooperative games with N agents, $A \geq 2$ actions per agent, and all joint actions safe, every algorithm has expected regret at least $c\sqrt{A^N K}$ on some instance when $K \geq A^N$, with $c = 1/(16\sqrt{2})$.*

2. *Without initial safety information or safety feedback before the first action, some instance has first-action violation probability at least $1/2$. A uniform zero-violation guarantee with failure probability $\delta < 1/2$ is therefore impossible.*

3. *Let $\gamma > 0$ and $0 < \delta \leq 1/4$. Given n independent observations $Y_1, \ldots, Y_n \sim \mathcal{N}(\theta, 1)$ of a candidate constraint with mean $\theta \in \{-\gamma, +\gamma\}$, any certification rule that declares the safe candidate safe with probability at least $1 - \delta$ and declares the unsafe candidate safe with probability at most δ requires*

$$n \geq (2\gamma^2)^{-1} \mathrm{kl}(1 - \delta, \delta).$$

Here $\mathrm{kl}(p, q) = p\log(p/q) + (1 - p)\log((1 - p)/(1 - q))$. This lower bound remains valid if the observations are supplied without executing the candidate action.

The three bounds distinguish representation complexity, initial safety information, and certification precision. The first concerns unstructured rewards, so retaining reward structure can change the statistical complexity. The third quantifies the cost of distinguishing near-boundary safe and unsafe actions and requires both soundness and successful certification. *Proof sketch.* An A^N-armed Bernoulli bandit gives the first bound by change of measure. The second follows from indistinguishable initial safe-action assignments. The third uses Gaussian binary testing. Full proofs appear in Appendix D.

5.1 Local tabular certification and feasible deviations

For tabular local constraints, use the sample-mean upper confidence bound $U^{g,\mathrm{tab}}_{k,\ell,h}(z)$ defined in Appendix A.3, retaining the other algorithm components. Let $M_\ell = |\mathcal{Z}_\ell|$ count local configurations and $L_\ell = \log(2M_\ell n_g/\alpha)$ be the logarithmic confidence factor, with a supplied integer observation budget $n_g \geq K$ per (ℓ, h). The count of an entry is the number of its observations before episode k. The sufficient certification count q_ℓ below is used only in the analysis.

Proposition 5.6 (Local tabular certification)**.** *Assume the model, fallback, planning, and unknown-tuple margin conditions. Suppose each episode with $\chi_k = 1$ has an unavailable comparator action $\mathbf{a} \notin \widehat{\mathcal{A}}_{k,h}(s)$ and a failed test $U^{g,\mathrm{tab}}_{k,\ell,h}(z) > 0$ at $z = z_\ell(s, \mathbf{a}) \notin \mathcal{Z}^0_{\ell,h}$, whose count increases by at least one through safely acquired observations in that episode. Set $q_\ell = \max\{1, \lceil 8R_g^2 L_\ell/\Delta^2_{\mathrm{safe}}\rceil\}$. With probability at least $1-\delta$, zero violation and Theorem 5.2's regret bound hold jointly, with*

$$B_K \leq \sum\nolimits_{h,\ell} |\mathcal{Z}_\ell \setminus \mathcal{Z}^0_{\ell,h}| q_\ell \leq H \sum\nolimits_{\ell=1}^{m} M_\ell q_\ell. \tag{28}$$

Each local estimate is reused by all joint actions containing the same tuple. A one-state binary pairwise chain has $4H(N-1)$ local entries, versus $H(N-1)2^N$ full constraint entries. This gain is also available to centralized planners that retain the same structure. *Proof sketch.* A failed comparator test has count below q_ℓ, and coverage increases one such count. Each initially uncertified entry can therefore be charged at most q_ℓ times. The full proof is in Appendix J.3.

For safe Π, let $\mathfrak{D}_i(\Pi)$ comprise replacements of agent i's message and action maps, with the other maps fixed, for which the entire resulting protocol remains in $\mathfrak{P}_{\mathrm{safe}}$. Include the identity and define the feasible unilateral-improvement gap

$$\mathrm{Gap}_i(\Pi) = \sup\nolimits_{\widetilde{\Pi}\in\mathfrak{D}_i(\Pi)} [V_1^{\widetilde{\Pi}}(\rho) - V_1^{\Pi}(\rho)]. \tag{29}$$

Corollary 5.7 (Feasible unilateral improvements)**.** *For every safe Π, $0 \leq \max_i \mathrm{Gap}_i(\Pi) \leq V_1^{\Pi^\star}(\rho) - V_1^{\Pi}(\rho)$. On the simultaneous confidence and regret event of Theorem 5.2,*

$$\sum\nolimits_{k=1}^{K} \max_i \mathrm{Gap}_i(\Pi^k) \leq \mathrm{Reg}(K). \tag{30}$$

Sublinear team regret gives a vanishing average maximum feasible-deviation gap. This implication is strict. With no messages, safe pairs $\{00, 11\}$ and reward $(a_1 + a_2)/2$ make 00 unilaterally stable but team-suboptimal by one. The result concerns deviations from ρ. General-sum and off-path equilibrium guarantees require different analyses. *Proof sketch.* The identity deviation gives a nonnegative gap. Every feasible unilateral deviation remains in the comparator's safe protocol class, so optimality of $\Pi^\star$ bounds its gain by team suboptimality. Taking the supremum, maximizing over agents, and summing proves the result. The full proof is in Appendix J.4.

6 Conclusion

SAFE-CMG-UCB combines reusable local safety certificates with a communication-compatible joint-action protocol. Zero violation follows from sound certification, while sublinear regret requires informative feedback. The distributed construction quantifies planning and communication costs.

Three questions remain. Can adaptive safe measurements achieve information coverage with minimal certification delay? What safe value and regret are attainable when communication prevents exact joint-action optimization? Which local transition-support certificates can replace the all-state fallback while preserving zero violation and finite certification cost?

Acknowledgments and Disclosures

Funding. This work received no external funding.

Competing interests. [Author completion required: competing-interests disclosure.]

Language-model assistance. AI tools was used during camera-ready preparation for wording, grammar checking, polishing, and coding for simulations.

References

[1] Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5(1):411–444, 2022.

[2] Kaiqing Zhang, Zhuoran Yang, and Tamer Başar. Multi-agent reinforcement learning: A selective overview of theories and algorithms. In *Handbook of Reinforcement Learning and Control*, pages 321–384. Springer, 2021.

[3] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[4] Sanae Amani, Christos Thrampoulidis, and Lin Yang. Safe reinforcement learning with linear function approximation. In *International Conference on Machine Learning*, pages 243–253. PMLR, 2021.

[5] Ming Shi, Yingbin Liang, and Ness Shroff. A near-optimal algorithm for safe reinforcement learning under instantaneous hard constraints. In *International Conference on Machine Learning*, pages 31243–31268. PMLR, 2023.

[6] Amirhossein Roknilamouki, Arnob Ghosh, Ming Shi, Fatemeh Nourzad, Eylem Ekici, and Ness Shroff. Provably efficient RL for linear MDPs under instantaneous safety constraints in non-convex feature spaces. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 51957–51995. PMLR, 2025. URL https://proceedings.mlr.press/v267/roknilamouki25a.html.

[7] Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored MDPs. In *Advances in Neural Information Processing Systems*, volume 14, pages 1523–1530. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/7af6266cc52234b5aa339b16695f7fc4-Paper.pdf.

[8] Carlos Guestrin, Michail Lagoudakis, and Ronald Parr. Coordinated reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning*, pages 227–234. Morgan Kaufmann, 2002. URL https://users.cs.duke.edu/~parr/icml02.pdf.

[9] Mohammad Gheshlaghi Azar, Ian Osband, and Rémi Munos. Minimax regret bounds for reinforcement learning. In *International Conference on Machine Learning*, pages 263–272. PMLR, 2017.

[10] Alex Ayoub, Zeyu Jia, Csaba Szepesvári, Mengdi Wang, and Lin Yang. Model-based reinforcement learning with value-targeted regression. In *International Conference on Machine Learning*, pages 463–474. PMLR, 2020.

[11] Dongruo Zhou, Quanquan Gu, and Csaba Szepesvári. Nearly minimax optimal reinforcement learning for linear mixture Markov decision processes. In *Conference on Learning Theory*, pages 4532–4576. PMLR, 2021.

[12] Eitan Altman. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, 1st edition, 1999.

[13] Yonathan Efroni, Shie Mannor, and Matteo Pirotta. Exploration-exploitation in constrained MDPs. *arXiv preprint arXiv:2003.02189*, 2020.

[14] Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. Provably efficient safe exploration via primal-dual policy optimization. In *International Conference on Artificial Intelligence and Statistics*, pages 3304–3312. PMLR, 2021.

[15] Arnob Ghosh, Xingyu Zhou, and Ness Shroff. Provably efficient model-free constrained RL with linear function approximation. *Advances in Neural Information Processing Systems*, 35: 13303–13315, 2022.

[16] Tao Liu, Ruida Zhou, Dileep Kalathil, Panganamala Kumar, and Chao Tian. Learning policies with zero or bounded constraint violation for constrained MDPs. *Advances in Neural Information Processing Systems*, 34:17183–17193, 2021.

[17] Archana Bura, Aria HasanzadeZonuzy, Dileep Kalathil, Srinivas Shakkottai, and Jean-Francois Chamberland. DOPE: Doubly optimistic and pessimistic exploration for safe reinforcement learning. *Advances in Neural Information Processing Systems*, 35:1047–1059, 2022.

[18] Aldo Pacchiano, Mohammad Ghavamzadeh, Peter Bartlett, and Heinrich Jiang. Stochastic bandits with linear constraints. In *International Conference on Artificial Intelligence and Statistics*, pages 2827–2835. PMLR, 2021.

[19] Ming Shi, Xiaojun Lin, and Sonia Fahmy. Competitive online convex optimization with switching costs and ramp constraints. *IEEE/ACM Transactions on Networking*, 29(2):876–889, 2021. doi: 10.1109/TNET.2021.3053910.

[20] Ming Shi, Xiaojun Lin, and Lei Jiao. Combining regularization with look-ahead for competitive online convex optimization. *IEEE/ACM Transactions on Networking*, 32(3):2391–2405, 2024. doi: 10.1109/TNET.2024.3350990.

[21] Sanae Amani, Mahnoosh Alizadeh, and Christos Thrampoulidis. Linear stochastic bandits under safety constraints. *Advances in Neural Information Processing Systems*, 32, 2019.

[22] Akifumi Wachi, Wataru Hashimoto, Xun Shen, and Kazumune Hashimoto. Safe exploration in reinforcement learning: A generalized formulation and algorithms. *Advances in Neural Information Processing Systems*, 36:29252–29272, 2023.

[23] Daniel Melcer, Christopher Amato, and Stavros Tripakis. Shield decentralization for safe multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 35: 13367–13379, 2022.

[24] Lijun Zhang, Lin Li, Wei Wei, Huizhong Song, Yaodong Yang, and Jiye Liang. Scalable constrained policy optimization for safe multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 37:138698–138730, 2024.

[25] Luigi Berducci, Shuo Yang, Rahul Mangharam, and Radu Grosu. Learning adaptive safety for multi-agent systems. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2859–2865. IEEE, 2024.

[26] David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca D. Dragan, and Claire J. Tomlin. Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1475–1481. IEEE, 2020. doi: 10.1109/ICRA40945.2020.9197129.

[27] Mo Chen, Zhengyuan Zhou, and Claire J. Tomlin. Multiplayer reach-avoid games via pairwise outcomes. *IEEE Transactions on Automatic Control*, 62(3):1451–1457, 2017. doi: 10.1109/TAC.2016.2577619.

[28] Eitan Altman and Adam Shwartz. Constrained Markov games: Nash equilibria. In *Advances in Dynamic Games and Applications*, pages 213–221. Birkhäuser, 2000.

[29] Pragnya Alatur, Giorgia Ramponi, Niao He, and Andreas Krause. Provably learning Nash policies in constrained Markov potential games. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems*, pages 31–39. International Foundation for Autonomous Agents and Multiagent Systems, 2024.

[30] Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. Provably efficient generalized Lagrangian policy optimization for safe multi-agent reinforcement learning. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, volume 211, pages 315–332. PMLR, 2023.

[31] Songtao Feng, Michael Dorothy, and Jie Fu. Learning Nash equilibrium of Markov potential games with a shared constraint via primal-dual optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16577–16585, 2025.

[32] Jeremy McMahan. Anytime-constrained equilibria in polynomial time. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 43399–43416. PMLR, 2025.

[33] Zeyang Li and Navid Azizan. Safe multi-agent reinforcement learning with convergence to generalized Nash equilibrium. *arXiv preprint arXiv:2411.15036*, 2024. URL https://arxiv.org/abs/2411.15036.

[34] Hsu Kao, Chen-Yu Wei, and Vijay Subramanian. Decentralized cooperative reinforcement learning with hierarchical information structure. In *Proceedings of the 33rd International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, pages 573–605. PMLR, 2022. URL https://proceedings.mlr.press/v167/kao22a.html.

[35] Han Zhong, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Can reinforcement learning find Stackelberg–Nash equilibria in general-sum Markov games with myopically rational followers? *Journal of Machine Learning Research*, 24(35):1–52, 2023. URL https://www.jmlr.org/papers/v24/22-0203.html.

[36] Geng Zhao, Banghua Zhu, Jiantao Jiao, and Michael Jordan. Online learning in Stackelberg games with an omniscient follower. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 42304–42316. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhao23o.html.

[37] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems*, volume 24, pages 2312–2320. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/hash/e1d5be1c7f2f456670de3d53c7b54f4a-Abstract.html.

[38] Rina Dechter. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence*, 113(1–2):41–85, 1999. doi: 10.1016/S0004-3702(99)00059-4.

[39] Jialei Liu, C. Emre Koksal, and Ming Shi. Bi-level online provisioning and scheduling with switching costs and cross-level constraints. In *2026 24th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*. IEEE, 2026. doi: 10.23919/WiOpt71098.2026.11568210.

[40] Qinghua Liu, Csaba Szepesvári, and Chi Jin. Sample-efficient reinforcement learning of partially observable Markov games. In *Advances in Neural Information Processing Systems*, volume 35, pages 18296–18308. Curran Associates, Inc., 2022. doi: 10.52202/068431-1330.

[41] Chi Jin, Zhuoran Yang, Zhaoran Wang, and Michael I. Jordan. Provably efficient reinforcement learning with linear function approximation. In *Proceedings of Thirty Third Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pages 2137–2143. PMLR, 2020. URL https://proceedings.mlr.press/v125/jin20a.html. Full version: arXiv:1907.05388.

[42] Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1): 193–201, 1992. doi: 10.1137/0221015.

Appendix Contents

A Notation and Algorithm Parameters **15**
A.1 Algorithm inputs and explicit parameter choices 16
A.2 Regression estimators and confidence radii 17
A.3 Tabular safety confidence bounds 18

B Related Work and Single-Agent Comparisons **18**
B.1 Single-agent specialization and the meaning of safety 18
B.2 Multi-agent coordination and solution concepts 20

C Proofs of the Safety and Regret Guarantees **21**
C.1 Confidence Event and Preliminary Lemmas 21
C.2 Proof of Theorem 5.1 (Zero Violation) 23
C.3 Proof of Theorem 5.2 (Regret Bound) 24
C.4 Proof of Corollary 5.4 (Certification Coverage) 25
C.5 Regret under Certified Approximate Planning 27
C.6 Role of instantaneous constraints in the regret proof 27

D Proof of Theorem 5.5 (Necessity Lower Bounds) **29**

E Planning and Distributed Coordination **30**
E.1 Factorization of the complete optimistic score 30
E.2 An exact protocol for overlapping scopes 31
E.3 Cost of action selection and state-dependent planning 32
E.4 Proof of Proposition E.1 (Planning Equivalence) 33
E.5 Sparse and dense forms of the motivating applications 34

F Backward Safety Certification and Robust Viability **35**

G Observation Models and Scope of the Guarantees **37**
G.1 Aggregate reward feedback 37
G.2 Scope of the guarantees 38

H Proofs of Proposition H.1 and Corollary H.2 **38**

I Proofs of Distributed Planning and Communication Guarantees **40**
I.1 Proof of Proposition I.1 (Distributed Implementation) 40
I.2 Proof of Proposition I.2 (Communication Precision) 42

J Proofs of Local Certification and Feasible-Deviation Results **44**
J.1 Proof of Lemma J.1 (Local Certificate Closure) 44
J.2 Example of certification at the safety boundary 45
J.3 Proof of Proposition 5.6 (Local Certification) 46
J.4 Proof of Corollary 5.7 (Feasible Deviations) 48

K Experiments on Learning, Certification, and Coordination **49**
K.1 Evaluation criteria and observation models 49
K.2 Regret versus the number of agents at fixed induced action width 49
K.3 Safe feedback and the certification bottleneck 50
K.4 Finite-horizon learning with unknown dynamics 52
K.5 Exact planning and interaction structure 53
K.6 State inputs and physical routing 54
K.7 Finite precision in score communication 55
K.8 Reproducibility, resources, and empirical scope 56

A Notation and Algorithm Parameters

Table 1: Notation used in the main results and proofs.

Symbol	Meaning
N, H, K, $T = KH$	Agents, steps per episode, episodes, and total interactions.
$\mathcal{S}$, $\mathcal{S}_i$, $\mathcal{A}_i$, $\mathcal{A}$	Global state space, local state space, individual action space, and joint-action space.
$\mathrm{proj}_J\,\mathcal{S}$, $\mathcal{A}_I$	Local state tuples attainable on scope J and the action product $\prod_{i\in I}\mathcal{A}_i$.
s_I, $\mathbf{a}_I$, I_ℓ	State and action restrictions to a set of agents, and the scope of constraint ℓ.
r_h, P_h, ρ	Team reward mean, transition kernel, and initial-state law.
m, $g_{\ell,h}$, $\mathcal{A}_h^{\mathrm{safe}}(s)$	Constraint count, constraint function, and true safe actions.
G_{comm}, $\mu^t_{i\to j,h}$, $\pi_{i,h}$	Physical graph, message rule, and local action rule.
$\mathcal{N}_i^{\mathrm{comm}}$	Physical communication neighbors of agent i.
$z^t_{i\to j}$, M^t_i	Round-t message and agent i's transcript, initially empty.
$\mathfrak{P}$, $\mathfrak{P}_{\mathrm{safe}}$	Admissible protocol class and its safe subset.
Π^0, Π^k, $\Pi^\star$	Known safe fallback, executed protocol in episode k, and optimal safe admissible protocol.
$\mathcal{H}_{k-1}$	Feedback and training information before episode k.
$\mathcal{H}^{\mathrm{pre}}_{\tau,h}$	History just before the step-h transition in episode τ, including the chosen state–action pair.
V_h^Π, $V_{k,h}$	Value of Π and optimistic value for episode k.
$\mathcal{E}_r$, $F = \lvert\mathcal{E}_r\rvert$, $d_{r,e}$	Reward scopes, number of factors, and dimension of factor e.
$D_r = \sum_e d_{r,e}$, d_g, d_P	Total reward dimension, padded constraint dimension, and padded transition dimension.
$\phi^r_{e,h}$, $\phi^g_{\ell,h}$, ϕ^P_h	Known reward, constraint, and transition-kernel features.
$\psi^V_h = \sum_{s'}\phi^P_h(s' \mid \cdot)V(s')$	Transition feature for the expectation of V.
$\theta^r_{e,h}$, $\theta^g_{\ell,h}$, θ^P_h	Unknown model parameters. Hats indicate ridge estimates.
B_r, B_g, B_P, R_r, R_g, $R_P = 1$	Parameter norm bounds and sub-Gaussian noise scales. $R_P = 1$ bounds the normalized transition noise.
$\mathcal{D}_k$, $\mathcal{D}^g_{k,\ell,h}$	Indices of individual observations available before episode k in a generic regression and a constraint regression.
$x^\tau_{r,e,h}$, $Y^\tau_{r,e,h}$, $x^\tau_{P,h}$, $Y^\tau_{P,h}$	Reward feature and factor measurement, followed by the historical-value transition feature and target. Only the transition pair is normalized by H.
$\Lambda^r_{k,e,h}$, $\Lambda^g_{k,\ell,h}$, $\Lambda^P_{k,h}$	Identity-regularized design matrices. Transition features and targets are normalized by H.
$\widehat{r}_{k,h}$, $\widehat{P}_{k,h}V$	Estimated mean reward and estimated next-value expectation. $\widehat{P}_{k,h}$ need not itself be a probability kernel.
$\beta^r_{k,e,h}$, $\beta^g_{k,\ell,h}$, $\beta^P_{k,h}$	Parameter-confidence radii. Unindexed radii are their maxima over the analyzed episodes and factors.
$\overline{\beta}(d,n;B,R)$	Deterministic upper radius for dimension d and at most n observations, defined in (45).
$b^r_{k,h}$, $b^P_{k,h}$, $U^g_{k,\ell,h}$	Reward and transition prediction half-widths, and the constraint upper confidence bound.
$\widehat{\mathcal{A}}^{\mathrm{imm}}_{k,h}(s)$, $\widehat{\mathcal{A}}_{k,h}(s)$	Actions certified by local initial or statistical information, and their union with the whole fallback.
$\widetilde{Q}_{k,h}$, $Q_{k,h}$	Unclipped optimistic score and score clipped to $[0, H-h+1]$.
$\mathcal{X}^\star_h$, χ_k, B_K	Comparator occupancy support, uncertified-episode indicator, and total number of such episodes.

Symbol	Meaning
Δ_{safe}, n_g	Margin for initially uncertified comparator tuples, and observation budget per constraint and step.
δ, α, $\mathcal{E}$	Overall failure budget, per-regression budget $\delta/[2H(F+m+1)]$, and simultaneous parameter-confidence event.
$\Delta_{k,h}$, $f_{k,h}$, $Z_{k,h}$	Optimistic-minus-executed value gap, its positive part, and bounded martingale differences in the regret proof.
$\mathfrak{M}_{K,H}(\delta)$, $\varepsilon_{k,h}$	Trajectory-concentration term and state-uniform score-optimization error. Neither permits a safety violation.
$\mathcal{F}$, $r_{\max}$, ω	Full planning factors, maximum action-scope size, and induced elimination width.
I_f, J_f, $\mathcal{F}^Q_{k,h}$	Action scope, state-input scope, and factors of the complete unclipped score.
$\mathcal{T}=(\mathcal{B},\mathcal{E}_{\mathcal{T}})$, B_b, S_b	Junction tree, bag variables, and parent separators.
u_b, $b(f)$, $b(i)$, $A_{\max}$	Bag host, factor's assigned bag, action's output bag, and maximum individual action-set size.
$p(b)$, $\mathrm{ch}(b)$, F_b	Parent bag, child bags, and sum of score and hard factors assigned to a bag.
Q^0_b, $Q^0_{b\uparrow}$	Fallback score from factors assigned to a bag and its subtree.
z_ℓ, $\mathcal{Z}_\ell$, $\mathcal{Z}^0_{\ell,h}$	Local state–action tuple, configuration set, and sound initial local certificates.
M_ℓ, $n_{k,\ell,h}(z)$, $\overline{g}_{k,\ell,h}(z)$	Local table size, entry observation count, and sample mean.
L_ℓ, q_ℓ, $\mathcal{E}_{\mathrm{tab}}$	Tabular confidence logarithm, sufficient certification count, and combined tabular-safety/reward/transition confidence event.
$S_{\max}$, r_g	Maximum local state-alphabet size and maximum safety-scope size in the local-table comparison.
R_h, R_0, r_{in}, D_G, r_{out}	Permitted rounds, fallback rounds, input distance, routed tree height, and output distance.
$\mathcal{D}_s$, $\mathcal{D}_0$, W_s	Distinct state and fallback source–host demands, and routed state-input bits.
L_f, z_f, ζ, m_Q	Uniform absolute score-factor bound, quantized integer factor, quantization step, and number of score factors.
Z_ζ, b_ζ, T_b	Integer score-magnitude bound, bits per score word, and number of separator assignments.
$\mathcal{M}$, b_p, $\mathcal{R}_p$, W_e, c_e	Routed messages, payload bits, physical paths, per-edge load, and directed edge capacity.
$\mathfrak{D}_i(\Pi)$, $\mathrm{Gap}_i(\Pi)$	Feasible unilateral protocol replacements and their improvement gap from ρ.
$\mathcal{X}_{k,h}$, $\mathcal{A}^{\mathrm{rec}}_{k,h}$	Recursively certified states and actions in the optional continuation extension.
$\widehat{\Theta}^P_{k,h}$, $\mathcal{C}^P_{k,h}$	Valid transition parameters inside the ellipsoid and the corresponding nonempty kernel uncertainty set.
η, δ_{conf}, δ_{exit}	Optional continuation escape tolerance, joint confidence-failure bound, and allocated trajectory escape budget.

A.1 Algorithm inputs and explicit parameter choices

The supplied bounds B_r, B_g, B_P, R_r, R_g must satisfy Assumption 3.1. Their validity cannot be inferred from the same finite data merely by substituting empirical norms or variances. State and action sets, features, scopes, the communication graph and alphabet, permitted rounds, the fallback, and sound initial certificates are also model inputs. The unknown parameters, optimal safe protocol, and comparator margin are not inputs to the algorithm.

Table 2: Explicit numerical parameters and their roles.

Quantity	Definition and use
H, K, T, δ	Supplied horizon H, episode count K, and failure budget $0 < \delta < 1$. The $T = KH$ environment steps exclude communication.
λ, R_P	$\lambda = 1$ in every ridge regression. $R_P = 1$ is valid because the normalized transition response lies in $[0, 1]$ on every history.
α	$\delta/[2H(F + m + 1)]$, allocating $\delta/2$ to all $HF + Hm + H$ regression processes. Regret concentration uses the remaining $\delta/2$.
$\beta^r_{k,e,h}$, $\beta^g_{k,\ell,h}$, $\beta^P_{k,h}$	Equations (32)–(33). At empty data, $\beta(I_d; B, R) = B + R\sqrt{2\log(1/\alpha)}$. No additional confidence multiplier is tuned.
$b^r_{k,h}$, $b^P_{k,h}$	Prediction half-widths in (11)–(12). The factor H reverses normalization of the transition regression.
$U^g_{k,\ell,h}$	Estimated constraint plus its prediction half-width, as in (13). Its acceptance threshold is exactly zero.
n_g	An observation bound per (ℓ, h) through episode K. If the auxiliary-feedback model allows at most n_{aux} extra observations per pair, take $n_g = K + n_{\mathrm{aux}}$. Trajectory-only feedback has $n_g = K$. Only the tabular variant requires this bound as input.
L_ℓ, q_ℓ	$L_\ell = \log(2M_\ell n_g/\alpha)$ enters the tabular certificate. $q_\ell = \max\{1, \lceil 8R_g^2 L_\ell/\Delta^2_{\mathrm{safe}}\rceil\}$ is a sufficient sample count in the proof. The learner does not use q_ℓ or Δ_{safe}.
Tie rule	Prefer the whole fallback when its score ties the certified optimum. Resolve other ties using the solver's fixed action and bag order, with conditional backtracking. The same rule is used in precomputation and execution.
$\varepsilon_{k,h}$	Zero for the exact planner. For an approximate planner it is a supplied uniform loss bound for the original, unclipped score. The achieved score defines the backward value.
L_f, $\zeta_{k,h}$	Take L_f to be the maximum absolute value of factor f over its finite local input domain, or any supplied upper bound. For the optional integer-score planner, $\zeta_{k,h} = 1/[T\max\{1, m_{Q,k,h}\}]$ bounds total optimization loss by one. Hard-feasibility indicators remain exact.
$R_0, r_{\mathrm{in}}, D_G, r_{\mathrm{out}}$	Fallback's round cost and the route quantities in (99)–(100). The sufficient vector-message budget $R_0 + r_{\mathrm{in}} + 2D_G + r_{\mathrm{out}}$ must not exceed the permitted R_h rounds.

The auxiliary-feedback mechanism is part of the observation model, not an implicit oracle invoked on rejected actions. The certification indicators χ_k, their sum B_K, the margin Δ_{safe}, and coverage witnesses appear only in the analysis. Hosts, routes, elimination order, and encodings are computational inputs. Appendices E and I specify their costs and correctness conditions.

A.2 Regression estimators and confidence radii

Predictability means that a feature is fixed before its response is observed. For response noise ε, conditional R-sub-Gaussianity means $\mathbb{E}[\exp(u\varepsilon) \mid \mathcal{H}_{\mathrm{pre}}] \le \exp(u^2R^2/2)$ for every $u \in \mathbb{R}$. The conditioning history $\mathcal{H}_{\mathrm{pre}}$ contains the feature and all previously revealed information. Simultaneous factor noises need not be independent.

For a d-dimensional regression, let $x_t \in \mathbb{R}^d$ be the feature and Y_t the observed response, whose conditional mean is $\langle x_t, \theta\rangle$ for an unknown parameter θ. Let $\mathcal{D}_k$ index the observations available before episode k, with distinct indices for repeated measurements. Fix regularization $\lambda = 1$ and define the design matrix and ridge estimate

$$\Lambda_k = I_d + \sum\nolimits_{t\in\mathcal{D}_k} x_t x_t^\top, \qquad \widehat{\theta}_k = \Lambda_k^{-1} \sum\nolimits_{t\in\mathcal{D}_k} x_t Y_t. \tag{31}$$

Here I_d is the d-dimensional identity matrix. It makes every design invertible, including at initialization, when $\Lambda_1 = I_d$ and $\widehat{\theta}_1 = 0$. For reward factor (e, h), use $x^\tau_{r,e,h} = \phi^r_{e,h}(s^\tau_{e,h}, \mathbf{a}^\tau_{e,h})$ and measured

factor reward $Y^{\tau}_{r,e,h}$ for $\tau < k$. For constraint (ℓ, h), $\mathcal{D}^{g}_{k,\ell,h}$ indexes all safely obtained observations, including executed-action feedback. Applying (31) defines $(\Lambda^{r}_{k,e,h}, \widehat{\theta}^{r}_{k,e,h})$ and $(\Lambda^{g}_{k,\ell,h}, \widehat{\theta}^{g}_{k,\ell,h})$.

For transition estimation, apply (31) to the historical-value samples in (10), for $\tau < k$, to obtain $\Lambda^{P}_{k,h}$ and $\widehat{\theta}^{P}_{k,h}$.

Allocate confidence failure $\alpha = \delta/[2H(F+m+1)]$ to each regression process and set the normalized transition-noise bound $R_P = 1$. For design Λ, parameter norm bound B, and noise bound R, define the parameter-confidence radius

$$\beta(\Lambda; B, R) = B + R\sqrt{\log\det\Lambda + 2\log(1/\alpha)}. \tag{32}$$

The three radii are explicitly

$$\beta^{r}_{k,e,h} = \beta(\Lambda^{r}_{k,e,h}; B_r, R_r), \quad \beta^{g}_{k,\ell,h} = \beta(\Lambda^{g}_{k,\ell,h}; B_g, R_g), \quad \beta^{P}_{k,h} = \beta(\Lambda^{P}_{k,h}; B_P, R_P). \tag{33}$$

Appendix C.1 proves simultaneous parameter confidence for these radii. All features in each regression are fixed before their corresponding responses are observed.

A.3 Tabular safety confidence bounds

For the variant in Proposition 5.6, observations are labeled local entries with mean $g_{\ell,h}(z)$. Let $M_\ell = |\mathcal{Z}_\ell|$, and let $n_{k,\ell,h}(z)$ and $\overline{g}_{k,\ell,h}(z)$ be the count and sample mean of observations of entry z before episode k. A supplied integer budget $n_g \geq K$ bounds all observations per (ℓ, h) through episode K. With the failure allocation $\alpha = \delta/[2H(F+m+1)]$, set $L_\ell = \log(2M_\ell n_g/\alpha)$ and define

$$U^{g,\mathrm{tab}}_{k,\ell,h}(z) = \begin{cases} \overline{g}_{k,\ell,h}(z) + R_g\sqrt{2L_\ell/n_{k,\ell,h}(z)}, & n_{k,\ell,h}(z) > 0, \\ +\infty, & n_{k,\ell,h}(z) = 0. \end{cases} \tag{34}$$

This tabular upper confidence bound replaces (13) in the certification test. Supplied local certificates and the whole fallback are retained. The infinite value prevents an unobserved entry from passing the statistical test. Appendix J.3 proves confidence simultaneously over entries and observation counts. Proposition 5.6 uses the margin-dependent count q_ℓ only in its analysis.

B Related Work and Single-Agent Comparisons

B.1 Single-agent specialization and the meaning of safety

Constraint semantics. Instantaneous and cumulative constraints specify different quantities. For this comparison, c_h denotes a state–action cost, τ_h a prescribed per-step threshold, C a cumulative budget, and π a possibly randomized single-agent policy. An actionwise requirement is

$$\mathbb{P}\big(c_h(s^k_h, a^k_h) \leq \tau_h \text{ for every } k, h\big) \geq 1 - \delta. \tag{35}$$

The probability is over the complete learning interaction. The constraint concerns the true cost, not the noise in its measurement. In contrast, an instantaneous expectation constraint restricts the action distribution through

$$\mathbb{E}_{a\sim\pi_h(\cdot|s)}[c_h(s,a)] \leq \tau_h. \tag{36}$$

A distribution satisfying (36) can assign positive probability to actions whose individual costs exceed τ_h. The expected cumulative cost $J_c(\pi)$ is constrained by the budget C through

$$J_c(\pi) := \mathbb{E}_{\rho,P,\pi}\left[\sum\nolimits_{h=1}^{H} c_h(s_h, a_h)\right] \leq C. \tag{37}$$

Even if (37) holds for every policy used during learning with high probability, it does not impose a per-action threshold. For example, a two-step trajectory with costs 0.8 and 0 satisfies a cumulative budget $C = 0.8$ but violates the instantaneous threshold 0.4 at its first step. Likewise, a bound on cumulative constraint violation may permit violations during learning. A pathwise cumulative budget $\sum_h c_h \leq C$ constrains each trajectory but likewise permits uneven costs across steps. A reward penalty $V^{\pi}_1(\rho) - \eta_{\mathrm{pen}} J_c(\pi)$, with penalty weight $\eta_{\mathrm{pen}} \geq 0$, imposes no feasibility requirement by itself. Our guarantees use the actionwise condition (35).

The zero-violation results of Liu et al. [16] and Bura et al. [17] enforce (37) for every selected policy under a known strictly feasible baseline policy. They therefore provide stringent guarantees for cumulative budgets, with a different feasible-policy class from the present actionwise problem. Bi-level provisioning and scheduling [39] couples online budget decisions to a lower-level CMDP through expected episode consumption. Its high-probability feasibility guarantee concerns those episode budgets. The present requirement constrains each executed joint action. Similarly, the randomized-policy extension in Amani et al. [4, Section 4] uses (36), whereas its deterministic-policy algorithm uses actionwise constraints. Randomization over safe actions preserves actionwise safety.

Hard constraints in online optimization. Hard feasibility also arises outside RL. Shi et al. [19] combine known demand constraints with hard ramp limits and a specified uncertainty set for future inputs. Shi et al. [20] study revealed covering constraints and look-ahead under switching costs. Both compare the online cost with an offline dynamic decision sequence through a competitive ratio. In our model, the constraint parameters and Markov dynamics are unknown, and safely obtained observations determine which actions can be certified. The comparator is an optimal admissible safe policy. These different information assumptions and benchmarks prevent a direct comparison of competitive ratios with Theorem 5.2.

Single-agent specialization. For $N = 1$, the joint action becomes an ordinary action and the communication protocol becomes a policy. Taking one reward factor, $F = 1$ and $D_r = d_r$, Corollary 5.4 specializes to

$$\mathrm{Reg}(K) \le \widetilde{O}\Big(\beta^r \sqrt{H d_r T} + H\beta^P \sqrt{H d_P T} + H^2 m d_g (\beta^g)^2/\Delta_{\mathrm{safe}}^2 + H\sqrt{T}\Big), \qquad T = KH. \tag{38}$$

Setting $N = 1$ does not imply $m = 1$ or $F = 1$. We choose one reward factor and retain all m constraints. The transition radius β^P corresponds to the normalized regression in (10). The specialization retains the known safe fallback at every modeled state and the comparator margin and feedback-coverage condition of Assumption 5.3. The zero-violation theorem does not require the latter condition, but the sublinear-regret corollary does.

Comparison with instantaneous-safe MDP results. The parameterizations and sources of safe information differ across the closest single-agent results. In a linear MDP, $P_h(s' \mid s, a) = \langle \phi(s,a), \mu_h(s') \rangle$ has a known state–action feature and unknown vector-valued measure. In the linear mixture model used here, $P_h(s' \mid s, a) = \langle \phi_h^P(s' \mid s, a), \theta_h^P \rangle$ has known transition features and a finite-dimensional unknown parameter. Dimensions from these two representations cannot be identified without an additional representation argument.

Table 3: Assumptions relevant to the single-agent comparison. The instantaneous-safety rows concern the latent constraint value, with noisy measurements used for learning. The symbols τ and $\bar{c}$ denote the cited works' safety thresholds.

Work	Safety and dynamics	Observations and initial safety	Regret conditions
Amani et al. [4], deterministic algorithm	Actionwise $c_h(s,a) \le \tau$ in a linear MDP.	Unknown rewards, costs, and transitions. Executed actions give exact reward and noisy cost feedback. A known action at every state has known cost strictly below τ.	Statewise feature sets are star-convex about the safe feature. A covering-number correction is given by Roknilamouki et al. [6].
Shi et al. [5]	$c_h(s,a,s') \le \bar{c}$ for realized transitions. Unsafe states are permitted in a linear mixture model.	Known rewards, unknown dynamics and transition costs, and noisy cost feedback. Exact transition supports and a seed safe subgraph are known.	Star-convex subgraph features, Lipschitz conditions on rewards and safe transitions, and a positive safety reserve after accounting for successor-feature variation.

Work	Safety and dynamics	Observations and initial safety	Regret conditions
Roknilamouki et al. [6]	Actionwise $c_h(s,a) \leq \tau$ in a linear MDP with continuous actions.	Unknown rewards, costs, and transitions. Noisy reward and cost feedback. A zero-cost action is known at each state.	Star-convexity, or a Local Point Assumption providing a safe neighborhood and suitable segments near the constraint boundary. The latter case includes an initial safe exploration phase.
Present $N = 1$ specialization	All $g_{\ell,h}(s,a) \leq 0$ on executed actions. Finite states/actions with linear mixture kernels.	Unknown rewards, constraints, and transitions. Noisy reward and constraint feedback. A safe fallback is available at every state. Any auxiliary safety observations must be specified and obtained safely.	Exact planning, strict margins on initially uncertified tuples of nonfallback comparator actions, and explicit growth of the observed constraint designs in uncertified comparator directions. No interpolation is required.

The geometric conditions in the prior actionwise results support safe exploration toward an optimal action, including an optimum on the constraint boundary. Our corollary instead requires a strictly positive margin for each initially uncertified tuple of a nonfallback comparator action and a separate condition on information obtained safely. Known local certificates can already cover boundary tuples, as the example in Appendix J.2 demonstrates. A margin alone does not provide information in an unobserved feature direction. The finite-action model removes the need for physically available actions along a continuous line segment, but explicitly requires informative safe observations. These assumptions therefore do not uniformly weaken the single-agent assumptions. Any acquisition cost for auxiliary safety observations is outside the environment-episode regret and must be included when comparing total learning resources.

The cited instantaneous-safe algorithms obtain regret of square-root order in the number of episodes under their respective assumptions, up to logarithmic factors and their stated exploration costs. In particular, Roknilamouki et al. [6] give a corrected analysis for the star-convex setting of Amani et al. [4] and extend the analysis to a specified non-star-convex class. Equation (38) provides the same qualitative combination of sublinear regret and high-probability actionwise safety under the stated margin and coverage assumptions. The comparison is qualitative and establishes neither a sharper single-agent rate nor a guarantee covering all of these models.

B.2 Multi-agent coordination and solution concepts

Local observations and latent-state games. The distinction between cooperative, competitive, and general-sum objectives is standard in MARL [2]. Local observations also require a precise information model. Liu et al. [40] study weakly revealing partially observable Markov games, using optimistic maximum-likelihood estimation to learn equilibria or compete with adversarial opponents. Here, centralized training observes the joint state. Each agent observes its own state component at execution and communicates before the joint action. The resulting protocol is a Markov map of the joint state, even though no agent need know that state in full. Our Bellman recursion relies on this model and the exact implementability condition. It does not establish a result for general latent-state games or arbitrary history-dependent decentralized policies.

Structured coordination and equilibrium benchmarks. Factored planning and coordinated reinforcement learning already retain multi-agent structure within a centralized MDP representation and use message passing to select a joint action from local value factors [7, 8]. Our use of variable elimination and induced-width complexity builds on this literature. For decentralized safety enforcement, Melcer et al. [23] decompose a supplied centralized shield using a sound safety-relevant environment abstraction. Here, local constraint parameters are learned from specified feedback, and initial and learned certificates enter the same local feasibility factors. These factors support team-regret analysis with safety during learning and execution through admissible communication.

Constrained equilibrium formulations encompass several safety notions. Alatur et al. [29] learn Nash policies in Markov potential games with coupled expected cumulative-cost constraints. McMahan [32] impose almost-sure bounds on every prefix of cumulative cost and reduce equilibrium computation to

action-constrained Markov games. Their computational guarantees take the game model as input, and they also discuss model-based learning extensions. Statewise constraints occur in equilibrium-based safe MARL as well. Li and Azizan [33] study controlled invariant sets and convergence to a generalized Nash equilibrium in statewise-constrained cooperative Markov games. A generalized Nash equilibrium is defined through feasible unilateral deviations. A globally optimal admissible safe team protocol maximizes the common return over the full admissible class. The two benchmarks need not coincide. Corollary 5.7 bounds the average feasible unilateral-deviation gap by team regret for the common payoff. The converse counterexample in Appendix J shows that unilateral stability need not imply team optimality. This consequence concerns admissible safe protocol replacements from the initial distribution. Convergence of independent player updates and Markov-perfect equilibrium conditions at every state require separate analyses.

Hierarchical information and Stackelberg learning. Kao et al. [34] study cooperative bandits and MDPs in which a follower observes the leader's action before choosing its own. Zhong et al. [35] study Stackelberg–Nash equilibria with myopically rational followers, and Zhao et al. [36] analyze repeated cooperative Stackelberg games with an omniscient follower. These information and response structures differ from an elimination order in a cooperative planner. Exact elimination computes a joint maximizing assignment, after which all agents execute simultaneously. No agent optimizes against a strategic follower response. Exact elimination orders preserve the feasible set and optimal value but can change computation, communication, and the selected tied maximizer.

Differential games and reach-avoid control. Fridovich-Keil et al. [26] develop iterative linear-quadratic approximations for solving nonlinear general-sum differential games. This approach models strategic interaction through repeated local game approximations. Chen et al. [27] combine pairwise reach-avoid solutions with maximum matching to obtain defending-team performance guarantees in multiplayer differential games. These methods concern game solving and reachability for specified dynamical models. Our analysis concerns statistical learning with unknown reward, transition, and constraint parameters and accounts for safety during data collection.

C Proofs of the Safety and Regret Guarantees

We use the action sets $\widehat{\mathcal{A}}_{k,h}(s) = \widehat{\mathcal{A}}^{\mathrm{imm}}_{k,h}(s) \cup \{\mathbf{a}^0_h(s)\}$, where $\mathbf{a}^0_h(s)$ is the known-safe fallback action. All value functions are defined on the full state space. The terminal value is zero, and the optimistic backup is clipped to $[0, H-h+1]$. Exact optimization and a fixed tie-breaking rule determine the executed admissible protocol Π^k before episode k. Certificate soundness guarantees safety independently of comparator availability. The latter affects regret through the episodes in which the certified action sets exclude a comparator action.

C.1 Confidence Event and Preliminary Lemmas

For each reward factor, let

$$x^\tau_{r,e,h} = \phi^r_{e,h}(s^\tau_{e,h}, \mathbf{a}^\tau_{e,h}), \qquad Y^\tau_{r,e,h} = \langle x^\tau_{r,e,h}, \theta^r_{e,h}\rangle + \varepsilon^\tau_{r,e,h}. \tag{39}$$

Here $x^\tau_{r,e,h}$ is the observed local feature, $Y^\tau_{r,e,h}$ is its factor reward, and $\varepsilon^\tau_{r,e,h}$ is the centered measurement noise. The identity-regularized design matrix and least-squares estimate are

$$\Lambda^r_{k,e,h} = I + \sum\nolimits_{\tau<k} x^\tau_{r,e,h}(x^\tau_{r,e,h})^\top, \qquad \widehat{\theta}^r_{k,e,h} = (\Lambda^r_{k,e,h})^{-1} \sum\nolimits_{\tau<k} x^\tau_{r,e,h} Y^\tau_{r,e,h}. \tag{40}$$

For each constraint (ℓ, h), let $\mathcal{D}^g_{k,\ell,h}$ index all constraint observations available before episode k, including auxiliary observations when supplied by the feedback model. Each measurement has its own index, including repeated measurements at the same feature. Every observation has a predictable feature x_t with $\|x_t\|_2 \le 1$ and response $Y_t = \langle x_t, \theta^g_{\ell,h}\rangle + \varepsilon_t$. Their design matrix and least-squares estimate are

$$\Lambda^g_{k,\ell,h} = I + \sum_{t\in\mathcal{D}^g_{k,\ell,h}} x_t x_t^\top, \qquad \widehat{\theta}^g_{k,\ell,h} = (\Lambda^g_{k,\ell,h})^{-1} \sum_{t\in\mathcal{D}^g_{k,\ell,h}} x_t Y_t. \tag{41}$$

The noise sequences are conditionally R_r- and R_g-sub-Gaussian, respectively. Their parameters satisfy $\|\theta^r_{e,h}\|_2 \le B_r$ and $\|\theta^g_{\ell,h}\|_2 \le B_g$.

For transitions, the normalized feature and response retain the value function fixed before that transition,

$$x^\tau_{P,h} = \psi_h^{V_{\tau,h+1}}(s^\tau_h, \mathbf{a}^\tau_h)/H, \qquad Y^\tau_{P,h} = V_{\tau,h+1}(s^\tau_{h+1})/H. \tag{42}$$

The design and estimator are

$$\Lambda^P_{k,h} = I + \sum\nolimits_{\tau<k} x^\tau_{P,h}(x^\tau_{P,h})^\top, \qquad \widehat{\theta}^P_{k,h} = (\Lambda^P_{k,h})^{-1} \sum\nolimits_{\tau<k} x^\tau_{P,h} Y^\tau_{P,h}. \tag{43}$$

Both $V_{\tau,h+1}$ and $x^\tau_{P,h}$ are measurable before s^τ_{h+1} is sampled. Let $\mathcal{H}^{\mathrm{pre}}_{\tau,h}$ denote the history immediately before this sample, including the current state and selected action. By linear-mixture realizability,

$$\mathbb{E}[Y^\tau_{P,h} \mid \mathcal{H}^{\mathrm{pre}}_{\tau,h}] = \langle x^\tau_{P,h}, \theta^P_h \rangle.$$

Since the value function is clipped on every history, including histories outside the confidence event, $Y^\tau_{P,h} \in [0,1]$ and $\|x^\tau_{P,h}\|_2 \le 1$. The centered response is conditionally R_P-sub-Gaussian for the declared conservative choice $R_P := 1$. The bound $\|\theta^P_h\|_2 \le B_P$ is part of Assumption 3.1.

Allocate failure probability $\alpha = \delta/[2H(F+m+1)]$ to each regression process. For a process with design Λ, parameter bound B, and noise parameter R, use

$$\beta(\Lambda; B, R) = B + R\sqrt{\log\det(\Lambda) + 2\log(1/\alpha)}. \tag{44}$$

The radii $\beta^r_{k,e,h}$, $\beta^g_{k,\ell,h}$, and $\beta^P_{k,h}$ use (B_r, R_r), (B_g, R_g), and (B_P, R_P), respectively. Any larger valid radii also satisfy the results below.

For explicit bounds that do not depend on the realized design, define

$$\overline{\beta}(d, n; B, R) := B + R\sqrt{d\log(1 + n/d) + 2\log(1/\alpha)}, \qquad d \ge 1, \quad n \ge 0. \tag{45}$$

Here d is the regression dimension and n bounds its sample count. For $\Lambda = I + \sum_{t=1}^{n'} x_t x_t^\top$, with $n' \le n$ and $\|x_t\|_2 \le 1$, the eigenvalue arithmetic–geometric mean inequality gives $\log\det\Lambda \le d\log(1 + n/d)$. Consequently, the maxima over the first K episodes used in Theorem 5.2 satisfy

$$\begin{aligned} \beta^r &\le \max_{e\in\mathcal{E}_r} \overline{\beta}(d_{r,e}, K; B_r, R_r), \\ \beta^P &\le \overline{\beta}(d_P, K; B_P, R_P), \qquad \beta^g \le \overline{\beta}(d_g, n_g; B_g, R_g). \end{aligned} \tag{46}$$

Here n_g bounds the observations in any one constraint regression. The algorithm uses the design-dependent radii in (32). These envelopes make their finite-sample and asymptotic dependence explicit. The comparator margin Δ_{safe} and failure count B_K are used only in the analysis.

Define the local predictions $\widehat{r}_{k,e,h} = \langle \phi^r_{e,h}, \widehat{\theta}^r_{k,e,h} \rangle$ and $\widehat{g}_{k,\ell,h} = \langle \phi^g_{\ell,h}, \widehat{\theta}^g_{k,\ell,h} \rangle$ with the same local arguments as the true functions.

Lemma C.1 (Simultaneous parameter confidence)**.** *With probability at least $1 - \delta/2$, the event*

$$\begin{aligned} \mathcal{E} := \big\{ &\|\widehat{\theta}^r_{k,e,h} - \theta^r_{e,h}\|_{\Lambda^r_{k,e,h}} \le \beta^r_{k,e,h}, \quad \|\widehat{\theta}^g_{k,\ell,h} - \theta^g_{\ell,h}\|_{\Lambda^g_{k,\ell,h}} \le \beta^g_{k,\ell,h}, \\ &\|\widehat{\theta}^P_{k,h} - \theta^P_h\|_{\Lambda^P_{k,h}} \le \beta^P_{k,h} \textit{ for all } k, h, e, \ell \big\} \end{aligned} \tag{47}$$

holds, with indices ranging over $k \in [K]$, $h \in [H]$, $e \in \mathcal{E}_r$, and $\ell \in [m]$. Consequently, on $\mathcal{E}$, for every state, joint action, and bounded value function V,

$$\begin{aligned} |r_{e,h} - \widehat{r}_{k,e,h}| &\le \beta^r_{k,e,h} \|\phi^r_{e,h}\|_{(\Lambda^r_{k,e,h})^{-1}}, \\ |g_{\ell,h} - \widehat{g}_{k,\ell,h}| &\le \beta^g_{k,\ell,h} \|\phi^g_{\ell,h}\|_{(\Lambda^g_{k,\ell,h})^{-1}}, \\ |P_h V - \widehat{P}_{k,h} V| &\le H\beta^P_{k,h} \|\psi^V_h / H\|_{(\Lambda^P_{k,h})^{-1}}. \end{aligned}$$

Arguments of the functions have been suppressed in these three inequalities.

Proof of Lemma C.1. For a predictable linear-regression sequence, write $\varepsilon_t = Y_t - \langle x_t, \theta \rangle$ and define the noise-feature sum $S = \sum_t x_t \varepsilon_t$. With regularization I, the ridge error obeys

$$\widehat{\theta} - \theta = \Lambda^{-1}(S - \theta), \qquad \|\widehat{\theta} - \theta\|_\Lambda \le \|S\|_{\Lambda^{-1}} + \|\theta\|_{\Lambda^{-1}} \le \|S\|_{\Lambda^{-1}} + B.$$

The last step uses $\Lambda \succeq I$ and $\|\theta\|_2 \le B$. For conditionally R-sub-Gaussian noise, the self-normalized inequality [37, Theorems 1 and 2] therefore gives, simultaneously at every sample size,

$$\|\widehat{\theta} - \theta\|_\Lambda \le B + R\sqrt{\log\det\Lambda + 2\log(1/\alpha)}$$

with probability at least $1-\alpha$. Apply this result to the HF reward processes, the Hm constraint processes, and the H transition processes. The preceding predictability and boundedness argument verifies the transition-regression hypotheses. The union bound over these processes has failure probability at most $H(F+m+1)\alpha = \delta/2$.

For any feature vector x, Cauchy–Schwarz gives

$$|\langle x, \widehat{\theta} - \theta\rangle| \le \|\widehat{\theta} - \theta\|_\Lambda \|x\|_{\Lambda^{-1}}.$$

Apply this inequality to each reward feature, constraint feature, and $x = \psi_h^V / H$. The parameter event implies the prediction bounds simultaneously for every such vector. In particular, no union bound over value functions or states is required. The same statement applies to a value function selected from the data after the parameter event is established. □

At the episode's fixed next-step value function, write $b^P_{k,h}(s,\mathbf{a}) = b^P_{k,h}(s,\mathbf{a}; V_{k,h+1})$. The confidence half-widths are

$$b^r_{k,h}(s,\mathbf{a}) := \sum\nolimits_{e\in\mathcal{E}_r} \beta^r_{k,e,h} \|\phi^r_{e,h}(s_e,\mathbf{a}_e)\|_{(\Lambda^r_{k,e,h})^{-1}}, \tag{48}$$

$$b^P_{k,h}(s,\mathbf{a}) := H\beta^P_{k,h} \|\psi_h^{V_{k,h+1}}(s,\mathbf{a})/H\|_{(\Lambda^P_{k,h})^{-1}}. \tag{49}$$

Thus, on $\mathcal{E}$, the optimistic reward and transition-value estimates overestimate their true counterparts by amounts in $[0, 2b^r_{k,h}]$ and $[0, 2b^P_{k,h}]$, respectively.

C.2 Proof of Theorem 5.1 (Zero Violation)

Lemma C.2 (Soundness of the coupled certificates)**.** *On $\mathcal{E}$, every action in $\widehat{\mathcal{A}}^{\mathrm{imm}}_{k,h}(s)$ satisfies $g_{\ell,h}(s_{I_\ell}, \mathbf{a}_{I_\ell}) \le 0$ for every ℓ. Every action in $\widehat{\mathcal{A}}_{k,h}(s)$ is therefore safe.*

Proof of Lemma C.2. Lemma C.1 implies $g_{\ell,h} \le U^g_{k,\ell,h}$ for every constraint. Membership in the immediate certified set provides either $z_\ell \in \mathcal{Z}^0_{\ell,h}$, which is safe by the known certificate, or $U^g_{k,\ell,h} \le 0$, which implies safety on $\mathcal{E}$. The remaining candidate, $\mathbf{a}^0_h(s)$, is safe by the initial safety certificate. □

Proof of Theorem 5.1. The candidate set is nonempty at every state because it contains the fallback action. The implemented protocol selects a joint action from this set. Lemma C.2 therefore establishes safety at every episode and step on $\mathcal{E}$, including states reached after previous exploratory actions. Lemma C.1 bounds the failure probability by $\delta/2 \le \delta$. □

Lemma C.3 (Optimism for an available comparator)**.** *On $\mathcal{E}$, if an admissible protocol Π selects candidate actions at every state-action pair in its reachable support, then $V_{k,h}(s) \ge V^\Pi_h(s)$ at every state reachable under Π at step h. Moreover, $V_{k,h}(s) \ge V^{\Pi^k}_h(s)$ for every h, s.*

Proof of Lemma C.3. The terminal values are zero. Suppose the assertion for Π holds at step $h+1$, and let $(s,\mathbf{a})$ be in its step-h support. Every successor with positive transition probability is reachable under Π. Monotonicity of the true transition kernel and confidence give

$$r_h(s,\mathbf{a}) + P_h V^\Pi_{h+1}(s,\mathbf{a}) \le r_h(s,\mathbf{a}) + P_h V_{k,h+1}(s,\mathbf{a}) \le \widetilde{Q}_{k,h}(s,\mathbf{a}).$$

The left-hand side belongs to $[0, H-h+1]$, so clipping the right-hand side to this interval preserves the inequality. Exact maximization over the candidate actions gives $V_{k,h}(s) \ge Q_{k,h}(s,\mathbf{a})$. Because the protocol is deterministic, the resulting inequality is the required comparison with its Bellman value. For Π^k, its selected actions are candidates at every state and its value recursion is defined on the full state space, so the same induction applies at every s. □

C.3 Proof of Theorem 5.2 (Regret Bound)

Let $b^r_{k,h} = b^r_{k,h}(s^k_h, \mathbf{a}^k_h)$ and define $b^P_{k,h}$ similarly. Recall χ_k from (22). Sums over k, h below range over $[K] \times [H]$ unless specified otherwise. Because $0 \le V_{k,1}(\rho) \le H$, Lemma C.3 implies, on $\mathcal{E}$,

$$V_1^{\Pi^\star}(\rho) - V_1^{\Pi^k}(\rho) \le V_{k,1}(\rho) - V_1^{\Pi^k}(\rho) + H\chi_k. \tag{50}$$

The indicator χ_k concerns the comparator's complete reachable support, including states that the learner never visits.

Lemma C.4 (High-probability regret decomposition)**.** *With probability at least $1-\delta$, jointly with $\mathcal{E}$,*

$$\mathrm{Reg}(K) \le 2\sum\nolimits_{k,h} b^r_{k,h} + 2\sum\nolimits_{k,h} b^P_{k,h} + HB_K + \mathfrak{M}_{K,H}(\delta), \tag{51}$$

$$\mathfrak{M}_{K,H}(\delta) := \sqrt{KH(H+1)(2H+1)\log(2/\delta)/3} \le H\sqrt{2T\log(2/\delta)}. \tag{52}$$

Here $\mathfrak{M}_{K,H}(\delta)$ bounds sampling error from the initial states and transitions.

Proof of Lemma C.4. Define the optimistic value error

$$\Delta_{k,h}(s) := V_{k,h}(s) - V_h^{\Pi^k}(s). \tag{53}$$

On $\mathcal{E}$, Lemma C.3 gives $0 \le \Delta_{k,h}(s) \le H - h + 1$. For the action selected by Π^k, the clipped optimistic Bellman backup and the confidence bounds imply

$$\Delta_{k,h}(s^k_h) \le 2b^r_{k,h} + 2b^P_{k,h} + P_h\Delta_{k,h+1}(s^k_h, \mathbf{a}^k_h). \tag{54}$$

Indeed, the unclipped optimistic backup is at least $r_h + P_hV_{k,h+1} \ge 0$ on $\mathcal{E}$, so its lower clipping is inactive. Upper clipping can only reduce the estimation error. More explicitly, the selected action has $V_{k,h} = Q_{k,h} \le \widetilde{Q}_{k,h}$ and

$$\begin{aligned}\widetilde{Q}_{k,h} - r_h - P_hV_{k,h+1} &= (\widehat{r}_{k,h} + b^r_{k,h} - r_h) \\ &\quad + (\widehat{P}_{k,h}V_{k,h+1} + b^P_{k,h} - P_hV_{k,h+1}) \\ &\le 2b^r_{k,h} + 2b^P_{k,h}.\end{aligned}$$

Subtracting $V_h^{\Pi^k} = r_h + P_hV_{h+1}^{\Pi^k}$ gives (54). All terms use the selected state–action pair.

To apply concentration without conditioning on $\mathcal{E}$, define the nonnegative error function $f_{k,h}(s) = [\Delta_{k,h}(s)]_+$, where $[u]_+ = \max\{u, 0\}$ denotes the positive part. The function is measurable at the start of episode k and satisfies $0 \le f_{k,h}(s) \le H - h + 1$ on every history, because both value functions lie in this interval. On $\mathcal{E}$, $f_{k,h} = \Delta_{k,h}$. Write $f_{k,1}(\rho) = \sum_{s\in\mathcal{S}} \rho(s) f_{k,1}(s)$ and define the initial-state and transition sampling errors

$$\begin{aligned}Z_{k,0} &:= f_{k,1}(\rho) - f_{k,1}(s^k_1), \\ Z_{k,h} &:= P_hf_{k,h+1}(s^k_h, \mathbf{a}^k_h) - f_{k,h+1}(s^k_{h+1}), \qquad h = 1, \dots, H-1.\end{aligned} \tag{55}$$

Conditional on the preceding history, the initial state is drawn from ρ, and each subsequent state is drawn from the true transition kernel. Thus, these variables in temporal order are martingale differences. They satisfy $|Z_{k,0}| \le H$ and $|Z_{k,h}| \le H - h$ unconditionally. Azuma–Hoeffding therefore gives

$$\sum\nolimits_{k=1}^{K}\sum\nolimits_{h=0}^{H-1} Z_{k,h} \le \mathfrak{M}_{K,H}(\delta)$$

with probability at least $1 - \delta/2$. The sum of squared increment bounds is $K\sum_{j=1}^{H} j^2 = KH(H+1)(2H+1)/6$, which gives the closed form in (52).

On $\mathcal{E}$, replacing each conditional next error by its sampled value plus $Z_{k,h}$ in (54) makes the value errors telescope. Since $\Delta_{k,H+1} = 0$,

$$\Delta_{k,1}(\rho) \le 2\sum\nolimits_{h=1}^{H}(b^r_{k,h} + b^P_{k,h}) + \sum\nolimits_{h=0}^{H-1} Z_{k,h}. \tag{56}$$

Summing this inequality and intersecting with the martingale event bounds its last term by $\mathfrak{M}_{K,H}(\delta)$. Combining this with (50) and taking a union bound over the confidence and martingale events proves the claim. The bound $\sum_{j=1}^{H} j^2 \le H^3$ gives the final inequality in (52). □

Lemma C.5 (Cumulative reward and transition widths)**.** *For the actual trajectories,*

$$2\sum\nolimits_{k,h} b^r_{k,h} \le 2\beta^r \sqrt{2FHD_rT\log(1+K)}, \tag{57}$$

$$2\sum\nolimits_{k,h} b^P_{k,h} \le 2H\beta^P \sqrt{2Hd_PT\log(1+K/d_P)}. \tag{58}$$

Proof of Lemma C.5. For vectors $x_1, \dots, x_K \in \mathbb{R}^d$ with $\|x_k\|_2 \le 1$, let $\Lambda_k = I + \sum_{\tau<k} x_\tau x_\tau^\top$ and $u_k = x_k^\top \Lambda_k^{-1} x_k$, the squared confidence width of x_k. Then $0 \le u_k \le 1$. The determinant lemma and $u \le 2\log(1+u)$ for $u \in [0,1]$ imply

$$\sum\nolimits_{k=1}^{K} u_k \le 2\log\det\Lambda_{K+1} \le 2d\log(1+K/d).$$

The last inequality follows from $\mathrm{tr}(\Lambda_{K+1}) \le d+K$ and the arithmetic–geometric mean inequality for its eigenvalues. By Cauchy–Schwarz,

$$\sum\nolimits_{k=1}^{K} \|x_k\|_{\Lambda_k^{-1}} \le \sqrt{2Kd\log(1+K/d)}.$$

Apply this separately at every (e,h) to the reward features. Summing over H steps, using $\log(1+K/d_{r,e}) \le \log(1+K)$, and applying $\sum_e \sqrt{d_{r,e}} \le \sqrt{FD_r}$ gives

$$\sum\nolimits_{k,h,e} \|\phi^r_{e,h}(s^k_{e,h}, \mathbf{a}^k_{e,h})\|_{(\Lambda^r_{k,e,h})^{-1}} \le H\sqrt{2KFD_r\log(1+K)}.$$

Multiplying by $2\beta^r$ and substituting $T = KH$ proves (57).

For transitions, the vectors used in the current width, $\psi_h^{V_{k,h+1}}(s^k_h, \mathbf{a}^k_h)/H$, are exactly the predictable features inserted into $\Lambda^P_{k+1,h}$ after the transition. They have norm at most one. Applying the same bound for each h gives

$$\sum\nolimits_{k,h} \|\psi_h^{V_{k,h+1}}(s^k_h, \mathbf{a}^k_h)/H\|_{(\Lambda^P_{k,h})^{-1}} \le H\sqrt{2Kd_P\log(1+K/d_P)}.$$

Multiply by $2H\beta^P$ and use $T = KH$ to obtain (58). Separate step-indexed matrices ensure that each estimate uses samples of its own transition parameter. □

Proof of Theorem 5.2. Combine Lemmas C.4 and C.5. The same confidence event proves zero violation, so the probability statement is joint with Theorem 5.1. □

C.4 Proof of Corollary 5.4 (Certification Coverage)

A sufficient feedback model. For example, suppose a safe sensing mechanism supplies one noisy observation in each coordinate direction of $\theta^g_{\ell,h}$ after every episode, for every (ℓ,h). These observations have features equal to the coordinate vectors and need not execute the candidate actions. Their design increment is I, which dominates $xx^\top$ for every $\|x\|_2 \le 1$. The coverage condition then holds with at most $(d_g+1)K$ constraint observations per pair, including the executed-action measurements. This additional sensing model must be specified separately from trajectory feedback. Its measurement cost remains separate from the stated regret.

Lemma C.6 (Certification of an interior action)**.** *On $\mathcal{E}$, if $g_{\ell,h}(s_{I_\ell}, \mathbf{a}_{I_\ell}) \le -\Delta_{\mathrm{safe}}$ and*

$$\beta^g_{k,\ell,h}\|\phi^g_{\ell,h}(s_{I_\ell}, \mathbf{a}_{I_\ell})\|_{(\Lambda^g_{k,\ell,h})^{-1}} \le \Delta_{\mathrm{safe}}/2,$$

then $U^g_{k,\ell,h}(s_{I_\ell}, \mathbf{a}_{I_\ell}) \le 0$.

Proof of Lemma C.6. Parameter confidence implies

$$U^g_{k,\ell,h} \le g_{\ell,h} + 2\beta^g_{k,\ell,h}\|\phi^g_{\ell,h}\|_{(\Lambda^g_{k,\ell,h})^{-1}} \le -\Delta_{\mathrm{safe}} + \Delta_{\mathrm{safe}} = 0.$$

□

Lemma C.7 (Number of uncertified comparator episodes)**.** *Under Assumption 5.3, on $\mathcal{E}$,*

$$B_K \le \frac{8Hmd_g(\beta^g)^2}{\Delta^2_{\mathrm{safe}}}\log(1+n_g/d_g).$$

Proof of Lemma C.7. If $\beta^g = 0$, confidence gives $U^g_{k,\ell,h} = g_{\ell,h}$ on $\mathcal{E}$ for every tuple and index. Every safe comparator action is then certified, so $B_K = 0$ and the bound holds. Hence suppose $\beta^g > 0$. Assign each episode with $\chi_k = 1$ to one witness (ℓ, h) supplied by Assumption 5.3. Denote its feature by x_k and write $u_k = x_k^\top (\Lambda^g_{k,\ell,h})^{-1} x_k$. The witness belongs to a nonfallback comparator action and its tuple lies outside $\mathcal{Z}^0_{\ell,h}$, so its constraint has the stipulated margin. Since its upper confidence bound is positive, Lemma C.6 implies

$$u_k > \Delta^2_{\mathrm{safe}}/[4(\beta^g)^2]. \tag{59}$$

Also $u_k \le 1$, because $\|x_k\|_2 \le 1$ and $\Lambda^g_{k,\ell,h} \succeq I$.

Fix a pair (ℓ, h), and let $\mathcal{I}_{\ell,h}$ be its assigned episodes. The covariance-growth condition and monotonicity of the logarithmic determinant give, for each assigned episode,

$$\log\det \Lambda^g_{k+1,\ell,h} - \log\det \Lambda^g_{k,\ell,h} \ge \log(1 + u_k) \ge u_k/2.$$

All design increments are positive semidefinite, including increments in unassigned episodes. Summing only the assigned increments is therefore at most the full determinant increase. Padding features with zeros when needed gives dimension d_g, and at most n_g observations of norm at most one yield

$$\sum\nolimits_{k \in \mathcal{I}_{\ell,h}} u_k \le 2 \log\det \Lambda^g_{K+1,\ell,h} \le 2 d_g \log(1 + n_g/d_g).$$

Combining this with (59) bounds $|\mathcal{I}_{\ell,h}|$ by $8 d_g (\beta^g)^2 \Delta^{-2}_{\mathrm{safe}} \log(1 + n_g/d_g)$. Each episode with an unavailable comparator action is assigned once, so summing over the Hm pairs proves the result. This argument charges only design growth from observations actually collected. □

Proof of Corollary 5.4. Substitute Lemma C.7 into Theorem 5.2. Fix H, F, m, the regression dimensions, the parameter and noise bounds, $\Delta_{\mathrm{safe}} > 0$, and $\delta \in (0, 1)$. If n_g is polynomial in K, the explicit envelopes in (46) give $\beta^r, \beta^P, \beta^g = O(\sqrt{\log(1 + K)})$. The certification term is therefore $O(\log^2(1 + K))$, while estimation and sampling contribute at most $O(\sqrt{K}\log(1 + K))$. Both contributions are sublinear in K. □

Necessity of informative safe feedback. A globally safe fallback and a positive margin do not suffice for sublinear regret under arbitrary trajectory-only feedback. The following construction makes this information requirement explicit.

Proposition C.8 (Safe feedback need not identify a better safe action)**.** *There is a pair of one-state, one-step linear-constraint problems with a known safe fallback such that every algorithm guaranteeing zero violation with probability at least $1 - \delta$ on both problems has expected regret at least $(1 - \delta)K$ on the problem whose optimal action is strictly safe.*

Proof of Proposition C.8. Let the actions be a_0 and a_c, with known rewards zero and one. Use constraint features $\phi^g(a_0) = (1, 0)^\top$, $\phi^g(a_c) = (0, 1)^\top$, and parameters $\theta^g_\pm = (-1, \pm\gamma)^\top$ for $0 < \gamma \le 1$. Take noiseless constraint observations, so $R_g = 0$, and the valid norm bound $B_g = \sqrt{2}$. Thus a_0 is a known safe fallback in both problems, whereas a_c is safe with margin γ only in the minus problem. Constraint measurements are obtained only by executing the corresponding action. Fallback measurements have identical laws in the two instances.

Write $\mathbb{P}_\pm$ and $\mathbb{E}_\pm$ for probability and expectation under the respective parameters. Let τ be the first episode in which a_c is executed. All observations before τ have identical laws, so $\mathbb{P}_-(\tau \le K) = \mathbb{P}_+(\tau \le K)$. Uniform safety in the plus problem requires the latter probability to be at most δ. If $N_c(K)$ is the number of candidate executions, then

$$\mathbb{E}_- \operatorname{Reg}(K) = K - \mathbb{E}_- N_c(K) \ge K\big(1 - \mathbb{P}_-(\tau \le K)\big) \ge (1 - \delta)K.$$

The features have norm one and the parameter norm is at most $\sqrt{2}$. Thus a positive comparator margin and global fallback do not make the untried candidate identifiable from the available observations. □

Coverage must also concern the comparator's reachable states, which can differ from those visited by the learner. Consider a two-step problem with two known-safe actions at the initial state. Action L gives reward $1/2$ and leads to a state with only a zero-reward fallback. Action R gives reward zero

and leads to a different state with a zero-reward fallback and a candidate action of reward one. The candidate is truly safe with positive margin, but its constraint feature is not observed or spanned by the observations obtained along the L branch. Even with known rewards and transitions, pessimistic planning can select L in every episode while the optimum selects R and the candidate. Regret is then $K/2$. No certification failure occurs at the states actually visited by the learner, although the comparator is uncertified on its own reachable set in every episode. Thus B_K must account for uncertified comparator actions beyond the learner's visited states.

C.5 Regret under Certified Approximate Planning

For a state-uniform error bound $\varepsilon_{k,h} \geq 0$, suppose an implementable planner returns a certified action $\widetilde{\mathbf{a}}_{k,h}(s) \in \widehat{\mathcal{A}}_{k,h}(s)$ satisfying

$$\max_{\mathbf{a} \in \widehat{\mathcal{A}}_{k,h}(s)} \widetilde{Q}_{k,h}(s, \mathbf{a}) - \widetilde{Q}_{k,h}(s, \widetilde{\mathbf{a}}_{k,h}(s)) \leq \varepsilon_{k,h}. \tag{60}$$

Define $V_{k,h}(s)$ to be the clipped score achieved by this action, and use these values in the normalized transition regressions. Clipping is nondecreasing and one-Lipschitz, so it preserves the score-error bound. Membership in the candidate set preserves safety.

For $\chi_k = 0$, let $\mathbf{a}^\star = \Pi^\star_h(s)$ at a comparator-reachable state. Confidence and (60) give

$$\widetilde{Q}_{k,h}(s, \widetilde{\mathbf{a}}_{k,h}(s)) \geq r_h(s, \mathbf{a}^\star) + P_h V_{k,h+1}(s, \mathbf{a}^\star) - \varepsilon_{k,h}.$$

The sum $r_h + P_h V_{k,h+1}$ lies in $[0, H - h + 1]$, so clipping the left side preserves this lower bound. Subtracting it from the comparator Bellman recursion gives

$$V_h^{\Pi^\star}(s) - V_{k,h}(s) \leq \varepsilon_{k,h} + P_h(V_{h+1}^{\Pi^\star} - V_{k,h+1})(s, \Pi^\star_h(s)) \tag{61}$$

at comparator-reachable states. Backward induction bounds this difference by $\sum_{t=h}^{H} \varepsilon_{k,t}$. The optimism and Bellman-error argument for the actually executed protocol is unchanged. Its selected action defines its optimistic value at each state, and the induction does not require that action to be an exact maximizer. Hence the general regret bound gains at most

$$\sum_{k=1}^{K} \sum_{h=1}^{H} \varepsilon_{k,h}. \tag{62}$$

This accumulated planning error is at most $KH\varepsilon$ for a uniform error bound ε. The output must remain certified feasible because reward-regret control alone does not ensure safety.

C.6 Role of instantaneous constraints in the regret proof

The proof combines return estimation from adaptive trajectories with feasible-action certification and execution of one joint policy. Standard optimistic-RL tools control return estimation. Applying these tools to the optimal admissible safe protocol also requires certification and implementability, which determine the extra assumptions for sublinear regret.

Predictable regression and value-dependent queries. The transition model has one fixed finite-dimensional parameter θ_h^P at each step. The observations in (10) satisfy a linear regression with that same parameter, although the historical values $V_{\tau,h+1}$ change across episodes. Each value function is fixed before its associated next state is sampled. Thus its centered response is conditionally mean zero, and the self-normalized parameter bound in Lemma C.1 applies to the adaptive features [37]. On that single confidence event, Cauchy–Schwarz gives, simultaneously for every $(s, \mathbf{a})$ and every function $V : \mathcal{S} \to [0, H]$,

$$\left|\widehat{P}_{k,h} V(s, \mathbf{a}) - P_h V(s, \mathbf{a})\right| \leq H \beta^P_{k,h} \|\psi_h^V(s, \mathbf{a})/H\|_{(\Lambda^P_{k,h})^{-1}}. \tag{63}$$

The query V may therefore depend on the current data and certified sets. No additional probability union over those queries is needed. The factor H converts the normalized response back to return units. The design must still contain the normalized features used in the observations. This is a value-targeted linear-mixture regression, in the model-based framework of Ayoub et al. [10], rather than a regression that relabels all previous transitions with the current value function.

Why a covering-number argument cannot be transferred unchanged. LSVI-UCB estimates Bellman targets using the current value function on previously collected next states. Its uniform concentration argument controls a class of possible value functions through a covering number [41, full version, Lemmas B.3, D.4, and D.6]. With instantaneous constraints, the maximization set also depends on the data. A covering of the optimistic action-value functions alone need not cover their maxima over changing safe sets. Roknilamouki et al. [6, Theorem 5.2 and Lemma 5.3] identify this obstruction and derive suitable bounds under additional geometric conditions. Equation (63) avoids that step by estimating a fixed transition parameter from predictable historical targets. It does not repair an arbitrary constrained-LSVI analysis, and it is not justified merely by having finite action sets. If historical targets were recomputed using $V_{k,h+1}$, the martingale argument above would no longer apply without a separate uniform concentration analysis.

Optimism, safety uncertainty, and certification loss. The reward and transition bonuses compensate their respective prediction errors, whereas U^g is an upper bound used to enforce feasibility. It is not subtracted from reward as a tunable penalty. On $\mathcal{E}$, the complete backup over a selected action satisfies

$$0 \le \widetilde{Q}_{k,h} - r_h - P_h V_{k,h+1} \le 2b^r_{k,h} + 2b^P_{k,h}. \tag{64}$$

This inequality bounds the optimistic backup error by the reward and transition confidence widths. All terms use the same state–action arguments. Backward induction then proves optimism for the executed certified policy. The comparator induction applies when the actions of $\Pi^\star$ are available on its entire reachable support. Episodes with $\chi_k = 1$ instead incur the explicit allowance $H\chi_k$, whose sum is HB_K. This certification cost appears only in the regret analysis. Increasing reward optimism alone does not, in general, eliminate this certification loss.

Safety-related exploration bonuses can serve a different role under different assumptions. The safe-subgraph method of Shi et al. [5] uses additional bonuses for current transition uncertainty and the effects of future and past safety uncertainty, together with known transition supports and geometric conditions. Our main model uses an all-state fallback and an explicit safe-information condition instead. Those bonuses do not by themselves establish coverage in our feedback model. Proposition C.8 exhibits linear regret when fallback observations cannot certify a better safe action.

Elliptical potential and certification counts. The usual tabular counting argument controls sums of inverse-count uncertainty widths in terms of the number of state–action pairs. Its linear counterpart bounds accumulated design-matrix widths through the logarithmic determinant. The transition width at an executed action uses exactly the feature added to the next design matrix, which is essential in Lemma C.5. Reward matrices are separate across factors and steps. Summing their widths produces the dimension factor $\sqrt{FD_r}$, with horizon dependence shown explicitly in (57).

An uncertified comparator action need not be executed by the learner, so its uncertainty cannot be charged to a visit count without further information. Assumption 5.3 supplies actual positive semidefinite design growth in a failed comparator direction x. For $\beta^g > 0$, a failure with margin Δ_{safe} implies $x^\top (\Lambda^g_{k,\ell,h})^{-1} x > \Delta^2_{\mathrm{safe}}/[4(\beta^g)^2]$. The determinant bound then limits the number of failures. The tabular analogue in Proposition 5.6 charges observations to local configurations that can be reused across many joint actions. Neither bound can charge unobserved tuples.

Concentration and the dependence on the horizon. Three concentration steps have distinct roles. Conditional sub-Gaussian noise and self-normalized concentration control the regressions, including adaptive feature choices. Hoeffding's lemma ensures that a centered normalized next-state value is 1-sub-Gaussian because its response lies in $[0, 1]$. The separately specified reward and constraint noise bounds need not assume bounded observations. Finally, Azuma–Hoeffding controls the difference between conditional next-state errors and sampled errors, including the random initial state. The functions $[\Delta_{k,h}]_+$ in (55) are bounded on every history. The martingale event is therefore established unconditionally and only then intersected with the confidence event $\mathcal{E}$.

The resulting bonuses use range bounds rather than conditional variances. Bernstein-based bonuses can sharpen horizon dependence in tabular RL [9, Algorithms 2–4 and Theorems 1–2], but they require a valid variance estimate and control of its error. Such a replacement here would need a corresponding variance-sensitive regression and regret proof. Substituting a smaller radius alone is unjustified. Theorem 5.2 therefore states the rate established by the range-based analysis without claiming minimax-optimal dependence on H. Instantaneous constraints enter this proof through

admissible actions, comparator certification, and safe information acquisition, while the concentration inequalities retain their stated probabilistic assumptions.

D Proof of Theorem 5.5 (Necessity Lower Bounds)

These one-state, one-step constructions separate unstructured joint-action complexity, missing initial safety information, and statistical uncertainty near the constraint boundary.

Theorem D.1 (Unstructured joint-action lower bound)**.** *For $A \geq 2$, $K \geq A^N$, and any learning algorithm, there is a one-state, one-step cooperative game with N agents, A actions per agent, all joint actions safe, and expected regret at least $c\sqrt{A^N K}$, with the universal choice $c = (16\sqrt{2})^{-1}$.*

Proof of Theorem D.1. Use one global reward factor and let $M = A^N \geq 2$. Identify the joint actions with the arms of an M-armed stochastic bandit. In instance $j \in [M]$, arm j has Bernoulli mean $1/2 + \epsilon$ and the other arms have Bernoulli mean $1/2$, so $\epsilon > 0$ is the optimal arm's reward advantage. Let instance 0 have all means $1/2$, and write $N_j(K)$ for the number of plays of arm j. Let $\mathbb{P}_j$ be the law of the complete interaction on instance j, including the learner's randomization, and let $\mathbb{E}_j$ denote expectation under this law. Write KL for relative entropy and kl for its Bernoulli specialization. The chain rule and $\mathrm{kl}(1/2, 1/2 + \epsilon) \leq 4\epsilon^2$ for $0 < \epsilon \leq 1/4$ give

$$\mathrm{KL}(\mathbb{P}_0 \| \mathbb{P}_j) \leq 4\epsilon^2 \mathbb{E}_0 N_j(K).$$

Since $N_j(K) \in [0, K]$, Pinsker's inequality implies

$$\mathbb{E}_j N_j(K) \leq \mathbb{E}_0 N_j(K) + K\sqrt{2}\epsilon\sqrt{\mathbb{E}_0 N_j(K)}.$$

Summing over j and using $\sum_j N_j(K) = K$ yields

$$\frac{1}{M}\sum\nolimits_{j=1}^{M} \mathbb{E}_j \mathrm{Reg}(K) \geq \epsilon K \left(1 - \frac{1}{M} - \sqrt{2}\epsilon\sqrt{K/M}\right).$$

Choose $\epsilon = (4\sqrt{2})^{-1}\sqrt{M/K} \leq 1/4$. The bracket is at least $1/4$, so the average regret is at least $(16\sqrt{2})^{-1}\sqrt{MK}$. At least one instance has this regret. The safety constraints may be identically -1, so the lower bound applies even when safety is completely known. It concerns the unstructured reward class and does not establish the same dependence for a fixed-dimensional structured model. □

Theorem D.2 (Initial information is necessary for stringent safety)**.** *Without any initial safety certificate or safety information before the first action, some instance has first-action violation probability at least $1/2$. Thus a uniform $1 - \delta$ zero-violation guarantee is impossible for $\delta < 1/2$.*

Proof of Theorem D.2. Consider two one-state, one-step instances with two actions and identical information before the first action. In the first instance, only action one is safe, and in the second, only action two is safe. If the algorithm chooses action one with probability p, its violation probabilities in the two instances are $1 - p$ and p. Their maximum is at least $1/2$. Both instances admit a safe action. The impossibility is caused by the absence of information identifying which action is safe. □

Theorem D.3 (Safety-margin dependence of nontrivial certification)**.** *Let $Y_1, \ldots, Y_n$ be independent $\mathcal{N}(\theta, 1)$ observations, where $\theta \in \{-\gamma, +\gamma\}$ and $\gamma > 0$. A candidate action has linear constraint $g(a) = \theta$. If a rule declares it safe with probability at least $1 - \delta$ when $\theta = -\gamma$ and at most δ when $\theta = +\gamma$, where $0 < \delta \leq 1/4$, then*

$$n \geq \frac{\mathrm{kl}(1 - \delta, \delta)}{2\gamma^2} = \Omega\left(\frac{\log(1/\delta)}{\gamma^2}\right).$$

The observations may be supplied without executing the candidate.

Proof of Theorem D.3. Let $\mathbb{P}_-$ and $\mathbb{P}_+$ be the laws of the observations under $-\gamma$ and $+\gamma$. The Gaussian relative entropy is

$$\mathrm{KL}(\mathbb{P}_- \| \mathbb{P}_+) = 2n\gamma^2.$$

Let C be the event that the rule declares the candidate safe. Data processing gives

$$2n\gamma^2 \geq \mathrm{kl}(\mathbb{P}_-(C), \mathbb{P}_+(C)) \geq \mathrm{kl}(1-\delta, \delta).$$

The stipulated probability bounds imply the last step by monotonicity of binary relative entropy. Finally, $\mathrm{kl}(1-\delta,\delta) = (1-2\delta)\log((1-\delta)/\delta)$ has order $\log(1/\delta)$ when $0 < \delta \leq 1/4$.

This is a lower bound for simultaneous soundness and successful certification. Without the latter condition, always refusing to certify is a zero-sample rule. Because the construction already grants safe measurements, the bound also applies when the learner receives less information. □

E Planning and Distributed Coordination

E.1 Factorization of the complete optimistic score

The linear model and the planning factorization impose different requirements. Linear realizability controls statistical estimation but does not imply that an optimistic Bellman score has small action scopes. For example, the transition-value term $P_h V(s, \mathbf{a})$ can depend on all actions even when the immediate reward decomposes locally. The norm in a transition confidence bonus can introduce additional couplings. We therefore require a factorization of the complete, unclipped score,

$$\widetilde{Q}_{k,h}(s, \mathbf{a}) = \sum\nolimits_{f \in \mathcal{F}^{Q}_{k,h}} q_{k,h,f}(s_{J_f}, \mathbf{a}_{I_f}), \tag{65}$$

Here $\mathcal{F}^{Q}_{k,h}$ indexes the real-valued score factors $q_{k,h,f}$, whose action and state scopes are I_f and J_f. For any agent sets $I, J \subseteq [N]$, write

$$\mathcal{A}_I := \prod\nolimits_{i \in I} \mathcal{A}_i, \qquad \mathrm{proj}_J \mathcal{S} := \{s_J : s \in \mathcal{S}\}. \tag{66}$$

These are the local action alphabet and the possible local state inputs. An empty scope has the single empty tuple as its input. Thus factor f is evaluated on $\mathrm{proj}_{J_f} \mathcal{S} \times \mathcal{A}_{I_f}$. The local tables in this representation must be available to the planner. Their construction, including evaluation of transition-value features, is part of the computational cost. In particular, evaluating a feature such as $\psi^V_h(s, \mathbf{a}) = \sum_{s'} \phi^P_h(s' \mid s, \mathbf{a}) V(s')$ may require summation over the next-state space.

The score in (65) is the quantity optimized before value clipping. For a fixed state and step, the map $z \mapsto \max\{0, \min\{H - h + 1, z\}\}$ is nondecreasing. Consequently, a maximizer of $\widetilde{Q}_{k,h}$ is also a maximizer of the clipped score. The clipped function itself need not have an additive factorization.

Each immediate safety constraint defines a hard feasibility factor

$$c_{k,\ell,h}(s_{I_\ell}, \mathbf{a}_{I_\ell}) = \begin{cases} 0, & z_\ell \in \mathcal{Z}^0_{\ell,h} \text{ or } U^g_{k,\ell,h}(z_\ell) \leq 0, \\ -\infty, & \text{otherwise.} \end{cases} \tag{67}$$

Here $z_\ell = (s_{I_\ell}, \mathbf{a}_{I_\ell})$, so initial and statistical certificates enter the same local feasibility factor. Its value $-\infty$ excludes an uncertified tuple from the maximization. Maximizing the sum of the score factors and these feasibility factors solves the optimization over $\widehat{\mathcal{A}}^{\mathrm{imm}}_{k,h}(s)$ whenever this set is nonempty. To evaluate the known fallback separately, compute $\widetilde{Q}_{k,h}(s, \mathbf{a}^0_h(s))$ and compare it with the certified optimum. This comparison implements optimization over $\widehat{\mathcal{A}}^{\mathrm{imm}}_{k,h}(s) \cup \{\mathbf{a}^0_h(s)\}$ without introducing a global disjunctive constraint into the factor graph.

State inputs and the physical communication graph. The action scopes I_f define the primal graph used for elimination. The state scopes J_f determine which observations are needed to instantiate a factor. A junction tree groups action variables into bags and specifies logical separator messages. The physical graph G_{comm} instead specifies one-round communication links between agents. Bags have computing hosts, and logical messages can be routed through agents outside their action scopes. These objects need not coincide. Hard safety factors use $J_f = I_f = I_\ell$, whereas score factors may use different state and action scopes.

For binary local states and a binary action a_1, consider the score

$$q(s, \mathbf{a}) = \mathbf{1}\{a_1 = s_1 \oplus \cdots \oplus s_N\} \tag{68}$$

This factor has action width zero and state scope $[N]$. Here $\oplus$ denotes addition modulo two, so its right-hand argument is the parity of the local state bits. It can arise as the first-step Bellman term of a known two-step model. The initial reward is zero, agent 1's next state equals the displayed indicator, and the other next local states are fixed. The terminal reward is agent 1's local state. All actions may be safe. A parity summary can avoid transmitting the raw global state to every agent, but it must convey information from all N inputs. Action width alone does not ensure local state access.

A direct encoding of one realized global state uses at most $\sum_i \lceil \log_2 |\mathcal{S}_i| \rceil$ bits before routing and headers. A tabular value function instead has $|\mathcal{S}|$ entries. An exponential state-space cardinality does not by itself imply exponential communication of one realized state.

E.2 An exact protocol for overlapping scopes

Write $\mathcal{N}_i^{\mathrm{comm}} := \{j : \{i,j\} \in E_{\mathrm{comm}}\}$ for agent i's physical neighbors. With integer round budget $0 \le R_{k,h} \le R_h$, a deterministic protocol at episode k and step h uses the message and action maps

$$M^t_{i\to j,k,h} = \mu^t_{i\to j,k,h}\left(s_i, \{M^u_{v\to i,k,h} : v \in \mathcal{N}_i^{\mathrm{comm}},\ u < t\}\right), \qquad \{i,j\} \in E_{\mathrm{comm}}, \tag{69}$$

$$a_i = \pi_{i,k,h}\left(s_i, \{M^u_{v\to i,k,h} : v \in \mathcal{N}_i^{\mathrm{comm}},\ u \le R_{k,h}\}\right). \tag{70}$$

Here $M^t_{i\to j,k,h}$ is the round-t message and $\mathbf{a} = (a_i)_{i\in[N]} = \Pi^k_h(s)$ is the final joint output. The maps and episode-specific model information are fixed using the training history before the episode. Their evaluation during execution uses only the local state and permitted messages. Thus $\Pi^k_h(s)$ is the maps' joint output, computed without a global-state policy-table lookup.

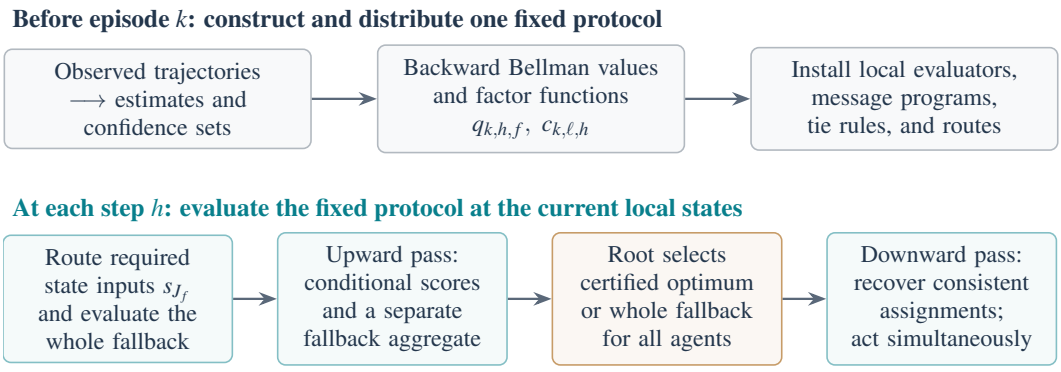


Figure 3: Execution of a fixed certified protocol. Pre-episode training fixes the model estimates, value functions, certificate maps, and factor programs. At the realized state, agents route local state and fallback inputs to their designated hosts. The upward pass aggregates conditional certified scores and the whole fallback's separate score. The root makes one common branch choice. The downward pass recovers consistent shared actions, which are delivered before simultaneous execution. Evaluating the current-state factors uses the fixed episode maps and does not refit the model.

Exact variable elimination solves the factorized optimization through conditional maximizations over action variables [38]. Exact implementability is a condition on one state-contingent protocol over the entire state space. Selecting a feasible action separately at each global state is insufficient if the resulting selections cannot be implemented from the agents' information. A sufficient implementation uses a junction tree of the full action-factor graph, together with routing over G_{comm} that permits each factor to be instantiated and every required message to be delivered before execution.

Let $\mathcal{T} = (\mathcal{B}, \mathcal{E}_{\mathcal{T}})$ be a rooted junction tree. Each bag $b \in \mathcal{B}$ contains a set B_b of action variables, every factor scope lies in at least one bag, and bags containing any given variable form a connected subtree. Let b_0 be the root, $p(b)$ the parent of a nonroot bag, and $\mathrm{ch}(b) := \{c : p(c) = b\}$ its children. Assign each score or feasibility factor to exactly one bag containing its scope. At the current state, let $F_b(\mathbf{a}_{B_b})$ denote the sum of the factors assigned to bag b. For a nonroot bag b with parent $p(b)$, write

$S_b = B_b \cap B_{p(b)}$ for its separator. Its upward message is the conditional subtree optimum,

$$m_{b\to p(b)}(\mathbf{a}_{S_b}) = \max_{\mathbf{a}_{B_b\setminus S_b}} \left\{ F_b(\mathbf{a}_{B_b}) + \sum\nolimits_{c\in \mathrm{ch}(b)} m_{c\to b}(\mathbf{a}_{S_c}) \right\}. \tag{71}$$

For each separator assignment, the bag stores a maximizing assignment of its remaining variables. Fix a total order on each finite action alphabet and use the increasing agent-index order within each bag to select the lexicographically first maximizing residual assignment. The root uses the same convention for its bag assignment. These choices and fallback priority on a score tie define the induced deterministic tie rule. Each upward message also carries the sum of the *score* factors evaluated at the whole fallback in its subtree, excluding hard feasibility factors from this scalar. The root obtains the certified optimum and the fallback score before selecting either branch, with fallback priority on a tie. An empty certified set has value $-\infty$ and therefore selects the fallback. The root sends one common branch flag down the tree. On the certified branch it also sends its separator assignments. Each child retrieves the maximizing assignment conditioned on the received separator and continues the pass. The running-intersection property ensures that each shared action variable has one value in every bag containing it. Output bags deliver each action and the common branch flag before the prescribed execution time.

Overlapping safety scopes thus share the same action variable throughout the optimization. They cannot select incompatible actions for a shared agent. The downward assignment pass is essential because independently selecting local maximizers can yield inconsistent actions under ties. A unilateral switch to a fallback component can violate a coupled constraint, even when both whole joint assignments are safe. The protocol assumes reliable, synchronous delivery and does not provide fault-tolerant commitment. Loopy max-sum alone does not guarantee exact optimization.

The order of elimination and message passing specifies computational dependencies. All agents share the team objective, and no follower best-response map is imposed. With exact optimization, the order can change runtime and the selected maximizer under ties, but not the optimal value or feasible set. This differs from a Stackelberg game, where the commitment order is part of the strategic model. Theorem 5.2 applies to every such exact, implementable planner. Corollary 5.4 additionally requires its coverage condition. Different tie choices can produce different observations and realized regret.

Example of shared-action consistency and branch selection. Consider binary actions with certified constraints $a_1 = a_2$ and $a_2 = a_3$, score factors $(1 - a_1)/3$ and $2a_3/3$, and fallback 000. Assign the first factor and constraint to bag $B_L = \{1, 2\}$ and the second pair to bag $B_R = \{2, 3\}$. Independent maximization selects $a_2 = 0$ in the left bag and $a_2 = 1$ in the right bag. Instead, root the junction tree at B_L. The right bag sends the score row $m_{R\to L}$ and its fallback score 0, while storing the conditional actions $a_3^\star$ locally as follows,

a_2	0	1
$m_{R\to L}(a_2)$	0	$2/3$
$a_3^\star(a_2)$	0	1

(72)

The root compares total scores $1/3$ for $a_1 = a_2 = 0$ and $2/3$ for $a_1 = a_2 = 1$, together with the fallback score $1/3$. It selects the certified branch and sends $a_2 = 1$ downward, recovering 111. If the physical graph is $1 - 2 - 3$ and the bags are hosted at agents 1 and 3, a logical tree message traverses two physical edges, and the chosen component must also reach agent 2. Both 000 and 111 are safe, whereas a componentwise mixture such as 101 is unsafe. Thus provisional local choices and individual fallback switches cannot replace the common branch and conditional assignment.

E.3 Cost of action selection and state-dependent planning

Let $\mathcal{F}$ contain all score and feasibility factors in the current planning problem. The primal action graph joins any two agents whose actions occur in the same factor. Let ω be the induced width of the selected elimination order, $A_{\max} = \max_i |\mathcal{A}_i|$, and $r_{\max} = \max_{f\in\mathcal{F}} |I_f|$. The induced width is the largest number of remaining neighbors of an eliminated variable after fill edges are added. Minimizing it over elimination orders gives treewidth. Local table construction takes at most $\sum_{f\in\mathcal{F}} A_{\max}^{|I_f|}$ factor evaluations. After those tables are available, exact variable elimination requires

$$O\big(\mathrm{poly}(N, |\mathcal{F}|) A_{\max}^{\omega+1}\big) \tag{73}$$

arithmetic operations per state. This bound includes the scopes of transition values and confidence bonuses. The immediate safety graph alone is insufficient to determine ω.

A reduced junction tree with bags of size at most $\omega + 1$ has separator messages with at most $A_{\max}^{\omega}$ scalar entries. An upward pass and downward assignment pass use $O(|\mathcal{B}|)$ logical messages and $O(\mathrm{diam}(\mathcal{T}))$ synchronous junction-tree rounds. These counts assume that each logical tree edge can transmit its message in one round. Realizing them over G_{comm} adds routing latency and congestion. Gathering the state variables in J_f, constructing local tables, and agreeing on the fallback comparison also require permitted communication. These table bounds count scalar entries, rather than bits. Appendix I.2 gives bit-traffic and edge-capacity bounds for certified score quantization.

The computational benefit therefore concerns action optimization at a fixed state. Tabular backward planning evaluates $H|\mathcal{S}|$ state-step pairs, in addition to the cost of constructing transition-value features. Since $\mathcal{S}$ can contain the full product $\prod_i \mathcal{S}_i$, this state-space cost can itself be exponential in N. An implementation may instantiate the current state's action factors and solve its action optimization only when that state is visited. This avoids storing unused action tables, but does not remove the need to obtain the future value functions used in the Bellman score. The present analysis does not establish efficient planning over an arbitrarily large factored state space.

Small scopes alone do not ensure small induced width. For example, all-to-all pairwise factors have scopes of size two but induce a complete primal graph with $\omega = N - 1$. A factor containing all action variables also gives a complete primal graph. In these cases the generic elimination bound is exponential in N. Other graphs with many edges can have different widths, so the relevant parameter is the chosen induced width. Special algebraic structure may permit additional reductions, but is not implied by the scope-based representation used here.

E.4 Proof of Proposition E.1 (Planning Equivalence)

The pre-episode planner and the execution protocol operate on different inputs. Before episode k, the learner fixes its estimates, confidence sets, certificate maps, and the value functions $V_{k,h}$ through the backward recursion. It also fixes the complete score-factor maps, their hosting agents, routes, message schedule, and deterministic tie rules. At step h, the numeric tables $q_{k,h,f}(s^k_{J_f,h}, \cdot)$ and $c_{k,\ell,h}(s^k_{I_\ell,h}, \cdot)$ depend on the newly observed local states and are instantiated at their hosts. Upward maximization, fallback comparison, downward assignment, and output delivery are then computations performed during execution. Observations from this episode are stored for subsequent model updates and do not alter the maps used to make the current episode's decisions.

Define the immediate certified-set optimum and the whole-fallback score by

$$M^{\mathrm{imm}}_{k,h}(s) := \max_{\mathbf{a}\in\mathcal{A}} \left\{ \sum\nolimits_{f\in\mathcal{F}^Q_{k,h}} q_{k,h,f}(s_{J_f}, \mathbf{a}_{I_f}) + \sum\nolimits_{\ell=1}^{m} c_{k,\ell,h}(s_{I_\ell}, \mathbf{a}_{I_\ell}) \right\}, \tag{74}$$

$$M^{0}_{k,h}(s) := \widetilde{Q}_{k,h}(s, \mathbf{a}^0_h(s)). \tag{75}$$

The certified-set optimum is $-\infty$ when the immediate certified set is empty. Since the fallback is always a candidate and has finite score,

$$\begin{aligned} \max_{\mathbf{a}\in\widehat{\mathcal{A}}_{k,h}(s)} \widetilde{Q}_{k,h}(s,\mathbf{a}) &= \max\{M^{\mathrm{imm}}_{k,h}(s), M^0_{k,h}(s)\}, \\ V_{k,h}(s) &= \mathrm{clip}_{[0,H-h+1]}\big(\max\{M^{\mathrm{imm}}_{k,h}(s), M^0_{k,h}(s)\}\big). \end{aligned} \tag{76}$$

Thus the local constrained optimization, followed by a separate fallback comparison, implements the Bellman action set exactly. The known fallback is evaluated using only score factors and need not pass the statistical hard factors. If recursive continuation is imposed as in Appendix F, its test must also occur among the feasibility factors, with its full scope.

Proposition E.1 (Equivalence of precomputation and exact evaluation)**.** *Fix the training history before episode k. Suppose that the values, certificate maps, complete score factors, fallback, and coordination program of Proposition I.1 are fixed from that history at every step. Use the same induced deterministic tie rule in all evaluations. Then tabulating the program's output for every state and evaluating the program only at the realized states produce the same joint action and clipped score at every state. If the program satisfies the communication budget, its evaluation at realized states is a single admissible protocol and requires no centralized global-state table lookup. Exact*

caching of its factor tables also leaves its output unchanged. Caches are indexed by episode, step, factor, and the factor's local state input, or invalidated when the corresponding map changes.

Use the same rule for acquiring any auxiliary safety observations in both implementations. Consequently, coupling the initial states, transition randomness, and feedback in the two evaluation schedules gives identical trajectories and regression updates. The statistical guarantees are unchanged by this choice of evaluation schedule.

Proof of Proposition E.1. Conditional on the pre-episode history, every value function, factor map, route, and tie rule in the program is fixed. At any state s, a factor therefore has the same entries whether they are computed in advance, read from an exact cache, or evaluated when the state is encountered. The fallback protocol likewise induces the same component actions. Induction through the fixed upward message dependencies gives the same separator tables and fallback score at the root. The root comparison and the downward conditional assignments use the same tie rules, so they recover the same joint action and clipped maximum. The construction in Proposition I.1 expresses every message and output using permitted local information, without querying the reference all-state table.

The equality holds for every state, including states not yet visited. With common initial-state and transition randomness, induction over environment steps gives the same states and actions. Coupling the feedback gives the same observations. Because the two schedules use the same fixed $V_{k,h+1}$, they also produce identical normalized transition features and responses in (10). Model updates at the end of the episode are therefore identical. Repeating this argument conditional on each subsequent training history proves the final claim. □

This equivalence requires exact evaluation of the fixed episode functions, including their successor-state dependencies. The feature $\psi_h^{V_{k,h+1}}(s, \mathbf{a})$ can require values at many next states. Memoization is valid only when all requested dependencies are evaluated and accessible. Arbitrary values at unvisited successors or model refitting during the episode would change the policy. Exact caching need not reduce the worst-case state-space computation. In particular, the finite-state implementation still computes $H|\mathcal{S}|$ value entries. An all-state joint-action table alone does not establish a decentralized lookup rule, because the agents may lack the information needed to identify its global-state row.

State scope, representation, and distribution cost. Action scopes I_f determine the induced width of elimination. State scopes J_f determine which current observations a factor host requires and need not equal I_f. Let $\mathcal{F}_{k,h}$ contain all score and hard feasibility factors at episode k and step h. If every factor is stored as a local table for all of its possible inputs, the number of entries is

$$\sum\nolimits_{h=1}^{H} \sum\nolimits_{f \in \mathcal{F}_{k,h}} |\operatorname{proj}_{J_f} \mathcal{S}| \prod\nolimits_{i \in I_f} |\mathcal{A}_i|. \tag{77}$$

Hard safety factors have $J_f = I_f = I_\ell$, and fallback storage is counted separately. Formula (77) measures the tabular storage cost. A compact exact evaluator may use less storage, but its construction and internal evaluation cost must still be counted. Loading these maps or tables at their designated agents is pre-episode communication, excluded from the runtime round bound in Proposition I.1. Local action scopes do not by themselves bound storage or distribution costs. A transition-value factor can depend on the global state even with a single action input.

E.5 Sparse and dense forms of the motivating applications

The robotic and wireless examples illustrate the constraint structure. Their finite-state, finite-action versions fall within the formulation only when the stated feature and feedback assumptions also hold. At a fixed horizon, a common shift and positive scaling into $[0, 1]$ preserve policy ordering.

For collision avoidance, consider discretized robot states and controls and a constraint

$$g_{ij,h}(s_{\{i,j\}}, a_i, a_j) = d_{\min}^2 - \|p_{i,h}^+(s_i, a_i) - p_{j,h}^+(s_j, a_j)\|_2^2 \le 0. \tag{78}$$

Here $d_{\min} > 0$ is the required separation, and $p_{i,h}^+(s_i, a_i)$ is robot i's next-position vector associated with the local state and control. The constraint has pairwise action scope. It guarantees separation at the next sampled time only when the positions $p_{i,h}^+$ describe the actual next positions, or when a valid

Phase	Computation and information	Cost accounting
Before episode k	Use $\mathcal{H}_{k-1}$ to fix estimates, certificates, and all Bellman values. Compile and distribute the factor evaluators, fallback maps, hosts, routes, schedule, and tie rules.	$H\|\mathcal{S}\|$ value entries in the tabular implementation. Factor construction, storage, and distribution are additional pre-episode costs.
Input acquisition at step h	Evaluate the fallback without acting. Route the current state components required by each factor and the fallback components needed for its score to the designated hosts.	At most $R_0 + r_{\text{in}}$ rounds under the vector-message model. Payload and routing loads are separate resources.
Factor instantiation	Evaluate $q_{k,h,f}(s^k_{J_f,h}, \cdot)$ and the exact local feasibility indicators from the frozen episode maps.	At most $\sum_f A_{\max}^{\|I_f\|}$ factor evaluations. Internal evaluation cost is not included in this count.
Upward pass	Compute conditional maximum-score tables and stored maximizing assignments. Aggregate the whole fallback's score separately.	At most D_G rounds. Separator tables have at most $A^{\omega}_{\max}$ score entries.
Joint assignment and delivery	The root selects the certified candidate or the fallback. Propagate separator assignments or the common fallback flag, then deliver each selected component to its owner.	At most $D_G + r_{\text{out}}$ rounds. All agents wait until the prescribed delivery phase has completed.
Environment action and feedback	Execute the committed components simultaneously and record rewards, constraint measurements, and the transition sample using the frozen $V_{k,h+1}$. Incorporate feedback into the next episode's estimates.	One environment step. The regret horizon counts these steps and excludes communication and computation time.

Table 4: Pre-episode planning and execution-time evaluation of one fixed certified protocol. Caching can replace exact table evaluations without changing the policy. It does not eliminate the need for future values or the communication needed to coordinate the current joint action.

robust bound accounts for prediction error. Local collision constraints on a chain have bounded width if the optimistic value factors preserve that chain structure. Requiring pairwise collision checks for every robot pair instead gives a complete safety graph. A formation constraint involving all robots gives a global factor. These cases illustrate how coordination can remain expensive even though each physical action is local. A geometric-neighbor description alone does not guarantee bounded width.

For wireless resource allocation, let the finite action of transmitter i include a quantized power p_i. An interference constraint at receiver j with transmitter scope I_j may take the form

$$g_{j,h}(s_{I_j}, \mathbf{a}_{I_j}) = \sum\nolimits_{i \in I_j \setminus \{j\}} H_{ij,h} p_i - I_j^{\max} \leq 0. \tag{79}$$

Here $H_{ij,h}$ is the interference-channel power gain from transmitter i to receiver j at step h, and $I_j^{\max}$ is its interference limit. Small interference neighborhoods can yield small factors, but a sparse implementation still requires bounded width of the complete planning graph. Network-wide interference or a shared total-power constraint can introduce a large factor. Some additive resource constraints admit specialized resource-indexed dynamic programs, so a large factor does not by itself establish intrinsic computational hardness. These examples specify constraint structures and their computational consequences. Controlled numerical evaluations appear in Appendix K.

F Backward Safety Certification and Robust Viability

The main analysis assumes a fallback that is safe at every modeled state and step. Hence, after any immediately safe action, at least one safe action is available at the next step. A transition-based viability test is unnecessary for this model. A different construction is required when the application specifies a terminal safe set, excludes states without safe actions, or restricts the fallback's domain.

For this extension, let $\mathcal{X}_{H+1} \subseteq \mathcal{S}$ be a supplied terminal safe set. An explicit immediate-certificate set is

$$\mathcal{A}^{\text{cert}}_{k,h}(s) := \widehat{\mathcal{A}}^{\text{imm}}_{k,h}(s), \tag{80}$$

using (14) with sound local initial certificates. Unlike the main candidate set, this set need not contain a fallback and can be empty. Known-safe pairs from a restricted fallback may be included in the

initial certificates only where its immediate safety is known. The viability argument below also applies to any other immediately safe certificate set.

Define the admissible parameter intersection and its kernel confidence set by

$$\widehat{\Theta}^P_{k,h} := \left\{\theta \in \Theta^P_h : \|\theta - \widehat{\theta}^P_{k,h}\|_{\Lambda^P_{k,h}} \le \beta^P_{k,h}\right\},$$
$$\mathcal{C}^P_{k,h} := \begin{cases} \{P_\theta : \theta \in \widehat{\Theta}^P_{k,h}\}, & \widehat{\Theta}^P_{k,h} \neq \varnothing, \\ \{P_\theta : \theta \in \Theta^P_h\}, & \text{otherwise.} \end{cases} \tag{81}$$

Here P_θ is the transition kernel induced by parameter θ. The set Θ^P_h contains the valid-kernel parameters specified in Section 3.3. It is nonempty because it contains the true parameter. On the simultaneous confidence event, $\theta^P_h \in \widehat{\Theta}^P_{k,h}$ and $P_h \in \mathcal{C}^P_{k,h}$. The second branch makes the kernel set nonempty even when the intersection is empty off that event. It conservatively restores the full prior kernel family, without providing a safe recovery action or establishing recursive feasibility. Define the backward recursion

$$\mathcal{X}_{k,H+1} = \mathcal{X}_{H+1}, \tag{82}$$
$$L^P_{k,h}(\mathcal{X}_{k,h+1} \mid s, \mathbf{a}) = \inf_{P \in \mathcal{C}^P_{k,h}} P(\mathcal{X}_{k,h+1} \mid s, \mathbf{a}), \tag{83}$$
$$\mathcal{A}^{\mathrm{rec}}_{k,h}(s) = \left\{\mathbf{a} \in \mathcal{A}^{\mathrm{cert}}_{k,h}(s) : L^P_{k,h}(\mathcal{X}_{k,h+1} \mid s, \mathbf{a}) = 1\right\}, \tag{84}$$
$$\mathcal{X}_{k,h} = \left\{s : \mathcal{A}^{\mathrm{rec}}_{k,h}(s) \neq \varnothing\right\}. \tag{85}$$

Here $L^P_{k,h}$ is the worst-case probability of reaching the next certified state set over the valid kernels in (81). The set $\mathcal{A}^{\mathrm{rec}}_{k,h}(s)$ collects immediately safe actions with probability-one certified continuation, and $\mathcal{X}_{k,h}$ contains states admitting such an action. The valid-kernel restriction ensures nonnegative, normalized transitions, so the infimum lies in $[0, 1]$.

The recursion removes immediately safe actions whose plausible successors lack a certified continuation. Inclusion of the true kernel transfers the robust transition condition to the actual dynamics.

Proposition F.1 (Robust viability)**.** *Suppose the true kernels belong to $\mathcal{C}^P_{k,h}$, every action in $\mathcal{A}^{\mathrm{cert}}_{k,h}(s)$ is immediately safe, and $\mathrm{supp}(\rho) \subseteq \mathcal{X}_{k,1}$. A protocol that selects an action in $\mathcal{A}^{\mathrm{rec}}_{k,h}(s_h)$ at every step satisfies all immediate constraints and remains in $\mathcal{X}_{k,h}$ at every step $h \in [H+1]$, almost surely. The assertion requires an implementable protocol. It does not establish a regret guarantee or provide an action when a recursive candidate set is empty.*

Proof of Proposition F.1. The initial state belongs to $\mathcal{X}_{k,1}$ almost surely. If $s_h \in \mathcal{X}_{k,h}$ and $\mathbf{a}_h \in \mathcal{A}^{\mathrm{rec}}_{k,h}(s_h)$, immediate safety follows from membership in $\mathcal{A}^{\mathrm{cert}}_{k,h}(s_h)$. Because the true kernel is one of the kernels in the confidence set,

$$P_h(\mathcal{X}_{k,h+1} \mid s_h, \mathbf{a}_h) \ge L^P_{k,h}(\mathcal{X}_{k,h+1} \mid s_h, \mathbf{a}_h) = 1.$$

Induction over h proves the claim. □

A fallback used in this extension must also satisfy the recursive continuation condition. Its immediate safety at a state does not imply that it avoids states without a safe next action. Feasibility from the initial distribution remains an additional assumption. In contrast, if $\mathcal{X}_{H+1} = \mathcal{S}$ and the globally safe fallback of the main model is included at every state, then $\mathcal{X}_{k,h} = \mathcal{S}$ for every h. This follows by backward induction because every valid kernel assigns probability one to $\mathcal{S}$. The continuation condition then imposes no additional restriction.

If the condition in (84) is relaxed to $L^P_{k,h}(\mathcal{X}_{k,h+1} \mid s, \mathbf{a}) \ge 1 - \eta$ for a specified $\eta \in [0, 1]$, each accepted action has true escape probability at most η while the certificates are valid. Suppose the initial-feasibility condition $\mathrm{supp}(\rho) \subseteq \mathcal{X}_{k,1}$ holds in every episode, and let δ_{conf} bound failure of the joint immediate-safety and transition-confidence event. A first-exit union bound over at most KH actions gives escape or confidence-failure probability at most $\delta_{\mathrm{conf}} + KH\eta$. This argument uses the conditional transition bound at each history with valid current certificates, without conditioning future transitions on the entire confidence event. Even without a safe recovery action outside the

recursively certified sets, zero violation therefore has probability at least $1 - \delta_{\mathrm{conf}} - KH\eta$. For a prescribed exit-error budget $\delta_{\mathrm{exit}} \in [0, 1]$, the explicit choice

$$\eta := \delta_{\mathrm{exit}}/(KH) \tag{86}$$

gives failure probability at most $\delta_{\mathrm{conf}} + \delta_{\mathrm{exit}}$. For example, the main confidence allocation permits $\delta_{\mathrm{conf}} = \delta/2$ and $\delta_{\mathrm{exit}} = \delta/2$. All feasibility and implementability conditions still apply. A positive tolerance cannot be treated as almost-sure invariance.

The strict threshold one requires certifying that the escape probability is exactly zero, which finite observations need not establish. For example, if the model family permits arbitrarily small positive probabilities of reaching an excluded state, observing only safe successors can leave such kernels in the confidence set after every finite sample sequence. The corresponding lower bound on safe-transition probability remains below one. A positive margin for the instantaneous constraint functions does not resolve this support-identification problem.

The immediate-constraint certification bound in the main theorem therefore does not imply a finite certification bound for this extension. Such a bound requires additional known transition-support information or a separate assumption ensuring that the comparator's continuation tests become certified. Any continuation test used in the action optimization must also be represented among its feasibility factors. Its scope contributes to the induced width and may be global even when all immediate safety constraints are local.

G Observation Models and Scope of the Guarantees

G.1 Aggregate reward feedback

Separate local reward regressions require the factor measurements in Assumption 3.1. If the learner instead observes only a scalar team reward Y^{agg} with conditional mean $r_h(s, \mathbf{a})$ and conditionally R_{agg}-sub-Gaussian noise, where R_{agg} is a supplied noise bound, it can use a joint regression. The stacked feature and parameter vectors in $\mathbb{R}^{D_r}$ are

$$\Phi_h^r(s, \mathbf{a}) = \big(\phi_{e,h}^r(s_e, \mathbf{a}_e)\big)_{e \in \mathcal{E}_r}, \qquad \Theta_h^r = \big(\theta_{e,h}^r\big)_{e \in \mathcal{E}_r}. \tag{87}$$

Stacking the $F \geq 1$ blocks gives $r_h = \langle \Phi_h^r, \Theta_h^r \rangle$, $\|\Phi_h^r\|_2 \leq \sqrt{F}$, and $\|\Theta_h^r\|_2 \leq \sqrt{F} B_r$. For each past episode $\tau < k$, the normalized feature and response are

$$x_{r,h}^{\mathrm{agg},\tau} = \Phi_h^r(s_h^\tau, \mathbf{a}_h^\tau)/\sqrt{F}, \qquad Y_{r,h}^{\mathrm{agg},\tau} = Y_h^{\mathrm{agg},\tau}/\sqrt{F}, \tag{88}$$

where $Y_h^{\mathrm{agg},\tau}$ is the observed, unnormalized team-reward measurement. The feature has norm at most one, the conditional mean response is $\langle x_{r,h}^{\mathrm{agg},\tau}, \Theta_h^r \rangle$, and the normalized noise scale is $R_{\mathrm{agg}}/\sqrt{F}$. Applying (31) gives the joint design, estimate, and confidence radius

$$\begin{aligned} \Lambda_{k,h}^{\mathrm{agg}} &= I_{D_r} + \sum\nolimits_{\tau<k} x_{r,h}^{\mathrm{agg},\tau} (x_{r,h}^{\mathrm{agg},\tau})^\top, \\ \widehat{\Theta}_{k,h}^{r,\mathrm{agg}} &= (\Lambda_{k,h}^{\mathrm{agg}})^{-1} \sum\nolimits_{\tau<k} x_{r,h}^{\mathrm{agg},\tau} Y_{r,h}^{\mathrm{agg},\tau}, \\ \beta_{k,h}^{\mathrm{agg}} &= \beta(\Lambda_{k,h}^{\mathrm{agg}}; \sqrt{F} B_r, R_{\mathrm{agg}}/\sqrt{F}). \end{aligned} \tag{89}$$

Use the same failure allocation α as in (32). Replacing F reward regressions by one makes this allocation conservative. The unnormalized reward estimate and prediction bonus are

$$\begin{aligned} \widehat{r}_{k,h}^{\mathrm{agg}}(s, \mathbf{a}) &= \langle \Phi_h^r(s, \mathbf{a}), \widehat{\Theta}_{k,h}^{r,\mathrm{agg}} \rangle, \\ b_{k,h}^{\mathrm{agg}}(s, \mathbf{a}) &= \sqrt{F}\, \beta_{k,h}^{\mathrm{agg}} \|\Phi_h^r(s, \mathbf{a})/\sqrt{F}\|_{(\Lambda_{k,h}^{\mathrm{agg}})^{-1}}. \end{aligned} \tag{90}$$

Replacing the reward estimate and bonus throughout the algorithm yields the same safety proof and regret decomposition. Write $\beta^{\mathrm{agg}} = \max_{k \in [K], h \in [H]} \beta_{k,h}^{\mathrm{agg}}$ for the largest joint-regression radius. Its reward-uncertainty contribution to regret is at most

$$2\sqrt{F}\, \beta^{\mathrm{agg}} \sqrt{2HD_rT \log(1 + K/D_r)}. \tag{91}$$

This follows by applying Lemma C.5's determinant argument to the normalized joint features at each step. The other terms are unchanged. The joint confidence norm generally couples all reward scopes, so aggregate feedback can require larger planning factors than factorwise feedback.

G.2 Scope of the guarantees

The main model assumes a known safe fallback at every modeled state and step. This condition is stronger than knowing a policy that is safe only along its trajectories from the initial distribution and excludes irreversible dead ends. The optional recursion in Appendix F addresses viability on a smaller domain, but requires feasible initial states and transition-support certificates. An immediate safety margin alone does not imply the main finite certification bound for that extension.

The sublinear-regret corollary additionally assumes that safely obtained data reduce uncertainty in relevant comparator feature directions. The existence of a baseline and strict safety margins on initially uncertified comparator tuples does not imply this coverage. Initial local certificates can cover boundary tuples, but their soundness must be supplied independently of noisy point estimates. Auxiliary safety observations, when available, must enter the constraint designs and sample counts. Their acquisition cost is absent from the environment regret. The result therefore does not establish sample-efficient safe exploration under every trajectory-only feedback model.

Exact decentralized execution requires sufficient reliable pre-action communication to reproduce the certified joint assignment. Proposition I.2 accounts for finite score-message precision and gives a sufficient schedule under edge capacities, conditional on correctly computed feasibility indicators. It does not supply a finite-precision implementation of arbitrary-real regression and safety tests. Packet losses, asynchronous execution, adversarial participants, and the effect of communication time on reward are outside the model. Independent local fallback switches after a communication failure can violate coupled constraints. Proposition H.1 shows that even treewidth one does not ensure implementability under a short round budget.

Computational scalability additionally depends on state-input acquisition, trajectory collection for centralized training, and distribution of the learned factor programs. The state space, program representation, factor evaluations, and routing loads can be large despite bounded action width. These costs are separate from the per-state action-optimization bound, which does not establish scalable planning for arbitrary factored state spaces.

The reward, constraint, and transition models must be correctly specified. Safety is asserted for the deterministic functions $g_{\ell,h}$, whose noisy measurements are used for estimation. The result does not constrain every realization of an unrelated random measurement or disturbance. Extending the analysis to misspecified or nonlinear models, partial observations that do not support the required coordination, and competitive objectives remains open. Appendix K evaluates the specified models, feedback budgets, complete action factors, and communication protocols. These synthetic results do not establish performance in physical systems or under model misspecification.

Broader impacts. Learning coordinated actions subject to instantaneous constraints could help reduce unsafe behavior in robotic teams and limit interference in wireless systems. These potential benefits depend on the validity of the modeled constraints, the availability of the assumed safe fallback and feedback, and the communication required to execute a certified joint action. Model misspecification or implementation failures can invalidate the guarantee and create unwarranted confidence in an unsafe controller. Moreover, maximizing a common team reward subject to safety constraints does not by itself ensure fair resource allocation. Applications require validation of the constraint model and execution mechanism, together with any additional fairness or privacy requirements relevant to the deployment.

H Proofs of Proposition H.1 and Corollary H.2

A centralized MDP representation preserves the joint dynamics and team reward. It does not, by itself, enforce the information available to each executing agent. The following example separates the optimum over communication-compatible safe protocols from the optimum of a full-state controller, even when statistical estimation and large factor scopes are absent. The argument uses causal locality of synchronous communication [42].

Proposition H.1 (A sharp communication requirement for hard-safe coordination)**.** *For every $N \geq 2$, there is a one-step cooperative game with two global states, binary actions, a unary reward factor, pairwise safety factors, and a known safe fallback at every state, with the following properties. All model functions can be revealed to the agents, every feasible action has safety margin one, and the*

complete reward–constraint action graph has treewidth one. Let $V_R^\star$ be the optimal value among safe protocols using at most R synchronous communication rounds on a path, and let $V_{\mathrm{full}}^\star$ be the optimal value of a full-state safe controller. Then

$$V_{\mathrm{full}}^\star = 1, \qquad V_R^\star = \begin{cases} \frac{1}{2}, & 0 \le R < N-1, \\ 1, & R \ge N-1. \end{cases} \tag{92}$$

The lower bound permits arbitrary message sizes and shared or private randomness independent of the current state. More generally, every $R < N-1$ protocol, whether safe or not, satisfies

$$\mathbb{E}[r(s,\mathbf{a})] \le \tfrac{1}{2} + \mathbb{P}(\exists \ell : g_\ell(s_{I_\ell}, \mathbf{a}_{I_\ell}) > 0)\,. \tag{93}$$

Equality constraints force the remote endpoint to share the informed agent's action. Until a message can traverse the path, that endpoint's decision cannot depend on the reward-relevant bit. The resulting loss of safe value persists even when the model is completely known.

Proof of Proposition H.1. Let G_{comm} be the path $1 - 2 - \cdots - N$. The only random local state is $s_1 = X$, where $X \sim \mathrm{Bernoulli}(1/2)$, and s_i is fixed for every $i > 1$. The bit is drawn after the episode's shared training information has been fixed. Each action belongs to $\{0,1\}$, and

$$r(s,\mathbf{a}) = \mathbf{1}\{a_1 = X\}, \qquad g_i(a_i, a_{i+1}) = 2\mathbf{1}\{a_i \ne a_{i+1}\} - 1, \quad i \in [N-1]. \tag{94}$$

The constraints require the common action $a_1 = \cdots = a_N$. Their graph is a path, and every safe joint action has $g_i = -1$ on every edge. The all-zero protocol is a known fallback safe in both states. The reward is a single local factor at agent 1. Both the reward and each constraint have a one-dimensional linear representation with feature equal to the corresponding displayed function and coefficient one. All feature magnitudes are at most one. The transition after the single decision can be any fixed known kernel on the two states and does not affect the value.

Condition on the shared training information and all random seeds. After zero communication rounds, an agent's information depends only on its own local state. Inductively, after t rounds, its information can depend only on local states within graph distance t, because a round-t message is a function of the sender's local state and its earlier transcript, and traverses one edge. Thus, if $R < N-1$, agent N cannot distinguish the two values of X. Its action a_N is independent of X, also after averaging over the state-independent seeds. Consequently, $\mathbb{P}(a_N = X) = 1/2$.

If $a_1 \ne a_N$, at least one adjacent pair has unequal actions, so

$$\mathbf{1}\{a_1 = X\} \le \mathbf{1}\{a_N = X\} + \mathbf{1}\{\exists i \in [N-1] : a_i \ne a_{i+1}\}. \tag{95}$$

Taking expectations proves (93). An almost-surely safe protocol therefore has value at most $1/2$, and the all-zero fallback attains this value.

A full-state controller chooses $a_i = X$ for every agent, obtaining reward one and satisfying every constraint. This choice is also implementable in $N-1$ rounds. Agent 1 sends X to agent 2, and each subsequent round forwards the bit one edge farther along the path. Afterwards all agents choose the bit they received. This proves (92). Finally, without the coupling constraints, agent 1 could obtain reward one with no communication by setting $a_1 = X$. The round requirement in the example arises from coordinating the safe joint action. □

Corollary H.2 (Centralized regret under a communication limit)**.** *Consider K independent episodes of* (94)*, with a fresh fair bit in each episode and at most $R < N-1$ rounds per decision. The protocols may depend on the entire preceding training history. If the probability of any constraint violation during the K episodes is at most δ, then*

$$KV_{\mathrm{full}}^\star - \mathbb{E}\left[\sum\nolimits_{k=1}^{K} r(s^k, \mathbf{a}^k)\right] \ge K\big(\tfrac{1}{2} - \delta\big). \tag{96}$$

Proof of Corollary H.2. Let E_k be the event of an edge-constraint violation in episode k. Conditional on the previous training history, the new bit is independent and fair, so the proof of (93) applies unchanged. Taking expectations and summing gives $\mathbb{E}\sum_k r(s^k,\mathbf{a}^k) \le K/2 + \sum_k \mathbb{P}(E_k)$. Since each $E_k \subseteq \bigcup_{j=1}^K E_j$, the last sum is at most $K\delta$. Use $V_{\mathrm{full}}^\star = 1$. □

Equation (96) compares against the unrestricted full-state controller. For $R < N - 1$, the admissible comparator in (4) has value $1/2$, which the fallback already attains. The full-state maximizing rule is then unimplementable, so Assumption 3.3 fails. This example isolates the communication requirement in that assumption. Bounded action-graph treewidth alone does not bound the rounds needed to attain the full-state safe optimum.

I Proofs of Distributed Planning and Communication Guarantees

I.1 Proof of Proposition I.1 (Distributed Implementation)

The following construction gives verifiable sufficient conditions for Assumption 3.3. Exact elimination and distributed action coordination in factored cooperative models are established techniques [7, 8, 38]. Here the construction applies them to the complete certified planning problem and accounts separately for state acquisition, the safe fallback, and delivery of the selected actions.

Fix an episode and a step, and suppress their indices. Suppose the complete unclipped score in (16) has known factors

$$\widetilde{Q}(s, \mathbf{a}) = \sum\nolimits_{f \in \mathcal{F}^Q} q_f(s_{J_f}, \mathbf{a}_{I_f}), \tag{97}$$

with finite values. Here $\mathcal{F}^Q$ indexes score factors, and I_f, J_f specify their action and state inputs. Allow sound initial local certificates $\mathcal{Z}^0_{\ell,h}$, meaning that $g_{\ell,h}(z) \le 0$ for every $z \in \mathcal{Z}^0_{\ell,h}$. For the fixed episode and step, define the local feasibility indicator

$$c_\ell(z) = \begin{cases} 0, & z \in \mathcal{Z}^0_{\ell,h} \text{ or } U^g_{k,\ell,h}(z) \le 0, \\ -\infty, & \text{otherwise,} \end{cases} \qquad z = (s_{I_\ell}, \mathbf{a}_{I_\ell}). \tag{98}$$

The immediate certified set is the intersection of these local tests. Each disjunction is evaluated within its original constraint scope and adds no new action dependency. Empty initial certificate sets recover the statistical tests alone. Let $\mathcal{F}$ contain all score factors and the hard factors (98). For a hard factor associated with constraint ℓ, set $J_f = I_f = I_\ell$. All factor maps and their scopes are fixed by the pre-episode training information. The relevant maps, including the membership rule for each $\mathcal{Z}^0_{\ell,h}$, are available at their designated computing agents. Their state arguments are acquired during execution. Write $G = G_{\mathrm{comm}}$ for the undirected physical graph. Each directed edge can carry an arbitrary finite vector per synchronous round. Local computations and routing instructions are deterministic. Communication is measured in rounds here. Bit traffic and packet sizes are accounted for separately below. The class $\mathfrak{P}_R(G)$ contains the deterministic protocols in (2) using this message model and at most R rounds on G.

Let $\mathcal{T}$ be a fixed rooted, reduced junction tree, meaning that no bag is contained in an adjacent bag, with bags $\{B_b : b \in \mathcal{B}\}$ of size at most $\omega + 1$. The bags cover all action variables, contain every factor's action scope, and satisfy the running-intersection property. Each bag b is hosted by an agent u_b. Assign each factor f to one bag $b(f)$ containing I_f, and each action variable i to one output bag $b(i)$ containing i. Thus $u_b \in [N]$ is a computing agent, $b(f)$ is the factor's bag, and $b(i)$ is the bag responsible for delivering agent i's selected action. Let b_0 be the root, $p(b)$ the parent of nonroot bag b, and $\mathrm{ch}(b) := \{c : p(c) = b\}$ its children. Use fixed shortest physical paths for this round construction. Ties between shortest paths are resolved before observing the current state. The bag assignments, hosts, action-alphabet orders, and paths are all part of the installed program. The later bit-traffic accounting also allows other specified simple routes and uses their actual lengths.

Write d_G for graph distance. Define the maximum input-routing distance

$$\begin{aligned} r_{\mathrm{in}} = \max\big(\{0\} \cup \{d_G(j, u_{b(f)}) : f \in \mathcal{F},\ j \in J_f\} \\ \cup \{d_G(i, u_{b(f)}) : f \in \mathcal{F}^Q,\ i \in I_f\}\big), \end{aligned} \tag{99}$$

where the second line accounts for acquiring the fallback's component actions to evaluate its score. With root b_0, define

$$\begin{aligned} D_G &= \max_{b \in \mathcal{B}} \sum_{(c, p(c)) \in \mathrm{path}_{\mathcal{T}}(b, b_0)} d_G(u_c, u_{p(c)}), \\ r_{\mathrm{out}} &= \max_{i \in [N]} d_G(u_{b(i)}, i). \end{aligned} \tag{100}$$

All distances used here must be finite. The quantity D_G is the largest sum of routed edge lengths along a logical root-to-bag path, while r_{out} is the maximum distance from an output host to its action owner. In particular, D_G need not equal the physical diameter. A known fallback protocol computes its component actions within a supplied, state-uniform integer budget $R_0 \geq 0$. Its maps can be evaluated without acting, with completion padded to R_0 rounds.

Proposition I.1 (Compilation of a certified planner into a protocol)**.** *Under the preceding conditions, if the permitted round budget satisfies*

$$R \;\geq\; R_0 + r_{\mathrm{in}} + 2D_G + r_{\mathrm{out}}, \tag{101}$$

there is a single deterministic state-contingent protocol in $\mathfrak{P}_R(G)$ that returns, for every state s,

$$\mathbf{a}(s) \in \arg\max_{\mathbf{a} \in \widehat{\mathcal{A}}^{\mathrm{imm}}(s) \cup \{\mathbf{a}^0(s)\}} \widetilde{Q}(s, \mathbf{a}). \tag{102}$$

The protocol has a fixed tie rule and also computes the exact clipped maximum in (17)*. It needs at most $\sum_{f \in \mathcal{F}} A_{\max}^{|I_f|}$ factor-table evaluations, followed by $O((|\mathcal{F}| + |\mathcal{B}|)A_{\max}^{\omega+1} + N)$ arithmetic operations for optimization at a fixed state. Each upward junction-tree message contains at most $A_{\max}^{\omega}$ score entries and one additional scalar for the fallback score. Each direction uses $|\mathcal{B}| - 1$ logical messages. Input acquisition, output delivery, and physical routing are counted separately.*

If the conditions hold at every step of the backward recursion and the permitted protocol class allows all local message maps with the stated graph, alphabet, and step-specific round budgets, this construction verifies Assumption 3.3 without a runtime global-state policy-table lookup.

Proof of Proposition I.1. First evaluate the fallback's message and action maps, without acting in the environment, in R_0 rounds. Agent i now knows $a_i^0(s)$. Send each state component required by a factor to that factor's host, and send each fallback component required by a score factor to its host. Use fixed shortest paths with tagged messages. Because an edge may transmit an arbitrary finite vector, packets sharing an edge can be concatenated, and all these transmissions finish in at most r_{in} rounds. Each bag host then evaluates its local score and feasibility tables and its assigned scores at the whole fallback.

At bag b, let $F_b(\mathbf{a}_{B_b})$ be the sum of all its assigned score and hard feasibility factors. The bag's fallback score is

$$Q_b^0(s) = \sum_{f \in \mathcal{F}^Q:\, b(f) = b} q_f(s_{J_f}, \mathbf{a}_{I_f}^0(s)). \tag{103}$$

Only score factors occur in (103), because the known fallback need not pass the statistical upper-confidence tests. For a nonroot bag, let $S_b = B_b \cap B_{p(b)}$. Compute the upward message by the exact recursion

$$m_{b \to p(b)}(\mathbf{a}_{S_b}) = \max_{\mathbf{a}_{B_b \setminus S_b}} \left[F_b(\mathbf{a}_{B_b}) + \sum\nolimits_{c \in \mathrm{ch}(b)} m_{c \to b}(\mathbf{a}_{S_c}) \right]. \tag{104}$$

Store a maximizing assignment for each separator value, using a fixed ordering of the bag variables and their actions to resolve ties as in Appendix E. Each upward message also carries the subtree fallback score

$$Q_{b\uparrow}^0(s) := Q_b^0(s) + \sum\nolimits_{c \in \mathrm{ch}(b)} Q_{c\uparrow}^0(s). \tag{105}$$

At a leaf the sum is zero. The root also evaluates the two branch scores

$$\begin{aligned} M_{\mathrm{cert}}(s) &:= \max_{\mathbf{a}_{B_{b_0}} \in \mathcal{A}_{B_{b_0}}} \left[F_{b_0}(\mathbf{a}_{B_{b_0}}) + \sum\nolimits_{c \in \mathrm{ch}(b_0)} m_{c \to b_0}(\mathbf{a}_{S_c}) \right], \\ M_0(s) &:= Q_{b_0\uparrow}^0(s) = \widetilde{Q}(s, \mathbf{a}^0(s)). \end{aligned} \tag{106}$$

The scores M_{cert} and M_0 compare the certified branch with the whole fallback. The root's product action alphabet $\mathcal{A}_{B_{b_0}}$ is defined in (66).

The running-intersection property implies that a variable occurring in the subtree of b and outside that subtree must occur in S_b. Therefore, conditional on $\mathbf{a}_{S_b}$, all other variables in that subtree can be maximized internally. Induction from the leaves proves that (104) is exactly the maximum sum of the subtree's factors subject to the separator assignment. The root's maximum $M_{\mathrm{cert}}(s)$ is consequently the exact optimum over $\widehat{\mathcal{A}}^{\mathrm{imm}}(s)$, taking $-\infty$ when empty. The fallback aggregate is $M_0(s)$.

The root selects the certified optimum when $M_{\mathrm{cert}}(s) > M_0(s)$ and otherwise selects the fallback. This fixes fallback priority on a tie. If the certified optimum is selected, the root chooses its maximizing bag assignment, then sends the corresponding separator assignments down the tree. Each child retrieves the stored maximizing assignment conditioned on the received separator and continues the pass. Since the root's selected value is finite, every visited conditional optimum is feasible. The running-intersection property makes every shared action variable consistent throughout the tree, and the recovered joint action satisfies every hard factor. On the confidence event, such an action satisfies every true constraint, since each accepted local tuple is initially certified or has a nonpositive valid upper bound. If the fallback is selected, the downward pass sends this common selection instead. Each output bag then sends the appropriate selection flag and, when needed, the selected action component to its owning agent. On the fallback branch, agent i uses its already computed $a_i^0(s)$, so its output bag need not acquire that component to deliver the selection flag. All agents execute after final delivery. Both branches satisfy (102), and monotone clipping gives the Bellman value in (17).

Each logical tree edge is routed along its fixed shortest physical path. An upward message is sent after all of its child messages have arrived. These dependency times can be fixed in advance from the routing lengths. The longest root-to-leaf sum of those lengths is D_G, so the upward pass finishes in at most D_G rounds. The downward pass obeys the reverse dependency order and takes at most another D_G rounds. Concurrent routed messages do not incur a capacity constraint under the stated vector-message model. Final delivery takes at most r_{out} rounds, proving (101).

Every message is a deterministic function of the hosting agent's local state, fixed pre-episode information, and previously received messages. All table evaluations, routes, schedules, and tie rules are specified independently of the realized state. Their composition is therefore one protocol over the entire state space, rather than an unrelated collection of statewise assignments. The computed optimum can also be evaluated for every state during finite-state backward planning. This establishes the full value and protocol requirement in Assumption 3.3 when the conditions hold at each step.

There are at most $A_{\max}^{|I_f|}$ entries to evaluate in factor f. Once these tables are available, summing factors in the bags and forming all messages takes $O((|\mathcal{F}| + |\mathcal{B}|)A_{\max}^{\omega+1} + N)$ arithmetic operations, including recovery of the N output components. A reduced junction tree with maximal bag size $\omega + 1$ has separators of size at most ω. Thus the upward table and message counts have the asserted sizes. Downward messages contain a selection flag and one separator assignment. □

The sufficient round budget can be reduced when additional structure permits a shorter protocol, as in the direct bit broadcast of Proposition H.1. Its applicability still requires factorization of the complete score and access to each factor's state arguments. Local safety scopes and linear realizability alone provide neither condition. The factor-table evaluation count excludes each factor's internal evaluation cost, such as summing transition features over next states. Finite-state backward planning also retains its $H|\mathcal{S}|$ state–step cost. Accounting for finite score messages additionally requires the state-input demands, payloads, and physical routes specified below.

I.2 Proof of Proposition I.2 (Communication Precision)

The round bound in Proposition I.1 assumes unbounded vector payloads. The following accounting separates local state acquisition from separator messages and shows the effect of communicating scores at a prescribed precision. All state and action alphabets are finite. Let b_j^s and b_i^a be the fixed bit lengths for encoding agent j's state and agent i's action,

$$b_j^s = \lceil \log_2 |\mathcal{S}_j| \rceil, \qquad b_i^a = \lceil \log_2 |\mathcal{A}_i| \rceil. \tag{107}$$

For a hard factor associated with constraint ℓ, set $J_f = I_f = I_\ell$. The state-input demands $\mathcal{D}_s$ and fallback-action demands $\mathcal{D}_0$ are the source–host pairs

$$\begin{aligned} \mathcal{D}_s &= \{(j, u_{b(f)}) : f \in \mathcal{F},\ j \in J_f\}, \\ \mathcal{D}_0 &= \{(i, u_{b(f)}) : f \in \mathcal{F}^Q,\ i \in I_f\}. \end{aligned} \tag{108}$$

Repeated requests for the same component at the same host are merged. Send each requested component from its source to its host along a fixed simple path. Write $\mathcal{R}_{j\to u}$ for a state-input route. The state-input traffic, summed over traversed directed edges, is

$$W_s = \sum\nolimits_{(j,u)\in\mathcal{D}_s} |\mathcal{R}_{j\to u}| b_j^s. \tag{109}$$

This quantity depends on the state scopes and routing, not only on action treewidth. A factor with a small action scope can still require many state components. Conversely, local state scopes and local hosts can avoid broadcasting the entire state to every agent. The message protocol imposes no state-privacy guarantee, since a factor table or separator message can reveal information about states outside the recipient's local observation.

Proposition I.2 (Score precision and certified feasibility)**.** *Fix an episode and step satisfying the structural and access conditions of Proposition I.1. Suppose the hard-factor indicators are evaluated correctly. A state-uniform score bound is*

$$L_f := \max_{\substack{u \in \mathrm{proj}_{J_f} \mathcal{S} \\ v \in \mathcal{A}_{I_f}}} |q_f(u, v)|, \qquad f \in \mathcal{F}^Q. \tag{110}$$

The local alphabets are finite and the score maps have finite values, so $0 \le L_f < \infty$. This quantity depends only on the frozen episode maps, not on the realized state. An explicitly established larger bound may be used instead. Computing the displayed maximum is part of factor preprocessing. For a precision $\zeta > 0$ fixed before the current state is observed, replace each score factor in action selection by its quantized integer score

$$z_f(s_{J_f}, \mathbf{a}_{I_f}) = \left\lfloor q_f(s_{J_f}, \mathbf{a}_{I_f})/\zeta \right\rfloor, \qquad m_Q = |\mathcal{F}^Q|. \tag{111}$$

Run exact integer max-sum on these factors, retain the same exact hard factors, and compare the certified optimum with the fallback under the same integer objective. The returned action $\widehat{\mathbf{a}}(s)$ belongs to $\widehat{\mathcal{A}}(s)$ for every state and satisfies

$$0 \le \max_{\mathbf{a} \in \widehat{\mathcal{A}}(s)} \widetilde{Q}(s, \mathbf{a}) - \widetilde{Q}(s, \widehat{\mathbf{a}}(s)) \le m_Q \zeta. \tag{112}$$

To bound score encodings and table sizes, define

$$Z_\zeta = \sum\nolimits_{f \in \mathcal{F}^Q} \lceil L_f/\zeta \rceil, \qquad b_\zeta = \lceil \log_2(2Z_\zeta + 2) \rceil, \qquad T_b = \prod\nolimits_{i \in S_b} |\mathcal{A}_i|. \tag{113}$$

Here Z_ζ bounds the magnitude of any integer partial score sum, b_ζ is a uniform codeword length, and T_b is the number of separator assignments for nonroot bag b. The factor count m_Q controls the total rounding loss. Every upward table entry and fallback-score aggregate can be encoded in b_ζ bits, including an infeasibility symbol. The execution messages other than those internal to the fallback protocol admit the following fixed payload bounds shown below.

Message	*Source and destination*	*Payload in bits*
State input	$(j, u) \in \mathcal{D}_s$	b_j^s
Fallback action	$(i, u) \in \mathcal{D}_0$	b_i^a
Upward table and fallback sum	$u_b \to u_{p(b)}$	$(T_b + 1)b_\zeta$
Downward commitment	$u_{p(b)} \to u_b$	$1 + \sum_{i \in S_b} b_i^a$
Final action delivery	$u_{b(i)} \to i$	$1 + b_i^a$

Let $\mathcal{M}$ be this fixed list of messages, with payload length b_p and simple directed physical route $\mathcal{R}_p$ for message $p \in \mathcal{M}$. For a specified finite-bit fallback implementation, let $\mathrm{bits}_e^0(s)$ count the bits that it sends on directed physical edge e at state s. If these counts are finite, take $W_e^0 := \max_{s \in \mathcal{S}} \mathrm{bits}_e^0(s)$, or any explicitly established larger bound. Then the complete execution has per-edge load and total traffic bounded by

$$W_e = W_e^0 + \sum\nolimits_{p \in \mathcal{M}:\ e \in \mathcal{R}_p} b_p, \qquad W = \sum\nolimits_e W_e. \tag{114}$$

Fixed schedules and encodings identify packets implicitly. The bounds exclude transport headers.

Quantization is applied once to the original score factors. Intermediate messages are exact integers, so rounding loss accumulates across factors independently of the number of routing hops. The hard tests and the separate fallback branch are preserved, leaving the candidate set unchanged.

Proof of Proposition I.2. For every factor and input, $0 \le q_f - \zeta z_f < \zeta$. Summing the m_Q factors gives

$$0 \le \widetilde{Q}(s, \mathbf{a}) - \zeta \sum\nolimits_f z_f(s_{J_f}, \mathbf{a}_{I_f}) < m_Q \zeta \tag{115}$$

when $m_Q > 0$, with zero quantization error when $m_Q = 0$. Let $\mathbf{a}^\star$ maximize the unrounded score over the candidate set. The integer optimizer, including its fallback comparison, satisfies $\sum_f z_f(s, \mathbf{a}^\star) \le \sum_f z_f(s, \widehat{\mathbf{a}})$. Apply (115) to both actions to obtain (112). The feasible set did not change, so feasibility and the confidence-event safety guarantee are preserved.

Every integer local score has magnitude at most $\lceil L_f/\zeta \rceil$. Because each factor is assigned to exactly one bag, a feasible message entry is a maximum of sums of distinct factors in one subtree. Its magnitude is at most Z_ζ. An infeasible entry uses a separate symbol. The alphabet $\{-Z_\zeta, \dots, Z_\zeta\} \cup \{\perp\}$ therefore needs b_ζ bits. Fallback aggregates obey the same bound. There are T_b separator assignments, plus one fallback scalar. Downward messages carry the common branch flag and one separator assignment, while output messages carry that flag and one action component. Padding the fallback branch to the same lengths gives fixed payloads in both cases. Recipients know their fallback components and ignore padded action bits. Summing routed payloads gives (114). □

The precision ζ is distinct from the continuation tolerance η in Appendix F. Correct evaluation of a hard test such as $U^g \le 0$ may require exact or certified local arithmetic. An uncertain feasibility indicator must not be rounded into acceptance. The proposition therefore bounds finite-bit *communication* conditional on correct indicators. It does not establish a finite-precision implementation of their computation. The episode's model maps, certificate descriptions, routing schedules, and factor programs must already be installed at their hosts. Their storage and distribution cost is separate and need not have a finite fixed-precision representation under the main statistical model.

Regret consequence. For this approximate planner, compute each backward value from the original unrounded score of the selected action,

$$V_{k,h}(s) = \mathrm{clip}_{[0,H-h+1]}\big(\widetilde{Q}_{k,h}(s, \widehat{\mathbf{a}}_{k,h}(s))\big).$$

At every backward step, the factorization, factor bounds, routing conditions, and exact feasibility tests must hold for the score built from this planner's own achieved $V_{k,h+1}$. They are not inherited from a separate exact-maximization recursion. Freeze the resulting values before the episode and use them in the historical transition samples (10). Define $m_{Q,k,h} := |\mathcal{F}^Q_{k,h}|$, the number of complete score factors at that episode and step. Choose $\zeta_{k,h} > 0$ before observing the current state, uniformly over the state space. Appendix C.5 then adds at most

$$\sum_{k=1}^{K} \sum_{h=1}^{H} m_{Q,k,h} \zeta_{k,h} \tag{116}$$

to the existing regret bound, while preserving zero violation on the confidence event. For a known $T = KH$, choosing $\zeta_{k,h} = 1/[T \max\{1, m_{Q,k,h}\}]$ makes this additional term at most one. For this choice, $Z_\zeta \le m_Q + T \max\{1, m_Q\} \sum_f L_f$, giving an explicit logarithmic score-word bound through (113). Certification coverage must hold for the modified algorithm's actual feedback. Deterministic rounding adds no failure event, but can change the trajectories and numerical confidence widths. The backward values remain the clipped original scores achieved by the selected actions.

A sufficient schedule with edge capacities. If directed physical edge e can transmit a positive integer c_e bits per round, one valid, conservative implementation sends the listed messages one at a time, hop by hop, in the input, upward, downward, and output dependency order. If the fallback finishes in R_0 rounds under those same capacities, the whole protocol finishes within

$$R_0 + \sum_{p \in \mathcal{M}} \sum_{e \in \mathcal{R}_p} \lceil b_p / c_e \rceil \tag{117}$$

rounds. Serial transmission removes congestion between messages, while parallel use of different edges can yield shorter schedules. The permitted round budget must accommodate the schedule actually used. Both the original vector-message bound and this capacitated bound assume that the environment does not advance until joint commitment completes.

J Proofs of Local Certification and Feasible-Deviation Results

J.1 Proof of Lemma J.1 (Local Certificate Closure)

For constraint ℓ, write

$$\mathcal{Z}_\ell = \{(s_{I_\ell}, \mathbf{a}_{I_\ell}) : (s, \mathbf{a}) \in \mathcal{S} \times \mathcal{A}\}, \qquad z_\ell(s, \mathbf{a}) = (s_{I_\ell}, \mathbf{a}_{I_\ell}). \tag{118}$$

Thus $\mathcal{Z}_\ell$ is the finite configuration set for constraint ℓ, and $z_\ell(s, \mathbf{a})$ extracts its local state–action tuple. The initial local certificate set $\mathcal{Z}^0_{\ell,h}$ contains only tuples whose constraint values are known to be nonpositive. Given a collection $\mathcal{B}_h \subseteq \mathcal{S} \times \mathcal{A}$ of known-safe joint state-action pairs, one valid choice is

$$\mathcal{Z}^0_{\ell,h} = \{z_\ell(s, \mathbf{a}) : (s, \mathbf{a}) \in \mathcal{B}_h\}. \tag{119}$$

The global fallback supplies the collection $\mathcal{B}_h = \{(s, \mathbf{a}^0_h(s)) : s \in \mathcal{S}\}$. Additional known-safe pairs may also be included. Because $g_{\ell,h}$ depends only on its local tuple, each projected certificate remains valid under every global completion of that tuple. Combining these projections can therefore certify additional joint actions whenever their shared variables are consistent. Constructing the local certificate sets from a global seed description incurs a separate preprocessing cost.

Lemma J.1 (Maximal safety information from seed projections)**.** *Fix the scopes and safe seeds $\mathcal{B}_h$, and use the projections in* (119)*. Define their consistent join*

$$\mathcal{J}_h(s) := \{\mathbf{a} : z_\ell(s, \mathbf{a}) \in \mathcal{Z}^0_{\ell,h} \text{ for every } \ell\}. \tag{120}$$

Every action in $\mathcal{J}_h(s)$ is safe under every collection of local constraint tables consistent with the seeds. If the tables are otherwise unrestricted, every action outside $\mathcal{J}_h(s)$ is unsafe for some collection of local tables consistent with the same seeds. Thus $\mathcal{J}_h(s)$ is exactly the set whose safety follows from the seed membership statements and the known scopes alone.

Proof of Lemma J.1. Every seed satisfies every constraint. Whenever a tuple occurs in a seed's projection for constraint ℓ, the value of that tuple under this constraint is nonpositive. Membership in $\mathcal{J}_h(s)$ supplies such a certificate for every constraint on the same joint action, proving safety.

Conversely, let $\mathbf{a} \notin \mathcal{J}_h(s)$. There is a constraint ℓ for which $z = z_\ell(s, \mathbf{a}) \notin \mathcal{Z}^0_{\ell,h}$. Choose its table to equal $+1$ at z and -1 at every other configuration, and choose all other constraint tables to be constantly -1. The distinguished tuple occurs in no seed for this constraint. Every seed is therefore safe, but $(s, \mathbf{a})$ violates constraint ℓ. The values ± 1 can be replaced by any allowed common positive magnitude if bounded tables are required. This proves maximality in the unrestricted local table class. Additional known linear relations can imply further safe configurations. □

For example, at one state with scopes $\{1, 2\}$ and $\{2, 3\}$, the safe seeds 000 and 101 certify the local relations $\{00, 10\}$ and $\{00, 01\}$. Their consistent join contains $000, 001, 100, 101$. The recombined actions 001 and 100 require no additional constraint observations to establish their safety. An initial local certificate can likewise be combined with a statistical certificate from another scope. In each case, shared action variables must have the same values in every constraint containing them.

J.2 Example of certification at the safety boundary

Local certificates can admit an optimal action on one constraint's boundary while observations certify its remaining constraints with positive margins. To isolate this effect, compare local certification with treating the fallback as one indivisible candidate. Consider one state, one step, and three binary-action agents, with

$$\begin{aligned} r(\mathbf{a}) &= a_3, \qquad g_1(a_1, a_2) = |a_1 - a_2|, \\ g_2(a_2, a_3) &= 2a_2a_3 - 1, \qquad \mathbf{a}^0 = 000. \end{aligned} \tag{121}$$

The two constraint scopes are $I_1 = \{1, 2\}$ and $I_2 = \{2, 3\}$. We omit the single fixed state from feature and tuple arguments. Use the one-hot feature map for g_1, with entries ordered as $00, 01, 10, 11$ and true parameter $\theta^g_{1,1} = (0, 1, 1, 0)^\top$. For g_2, use $\phi^g_{2,1}(a_2, a_3) = (2a_2a_3 - 1, 0, 0, 0)^\top$ and $\theta^g_{2,1} = (1, 0, 0, 0)^\top$. Both models thus have dimension $d_g = 4$. Set $B_g = \sqrt{2}$. Measurements have zero noise, so $R_g = 0$ and every constraint radius in (32) is $\sqrt{2}$. The reward feature is a_3, with true parameter one, $B_r = 1$, and $R_r = 0$. All feature norms and parameter norms meet the model assumptions. The transition after the single decision has no effect on value. Regression data sets are initially empty, and the regularization is I.

The safe team optimum 001 has reward one, whereas the fallback has reward zero. Consider first the candidate-set rule

$$\{\mathbf{a} : U^g_{k,\ell,1}(z_\ell(s, \mathbf{a})) \le 0 \text{ for every } \ell\} \cup \{000\},$$

which uses the seed only as a complete joint action. This rule executes 000 forever. Indeed, after n such observations, the 00 estimate in the first constraint table is zero, and its upper confidence bound is $\sqrt{2}/\sqrt{1+n} > 0$. Each other entry remains unobserved, with upper bound $\sqrt{2} > 0$. Thus every joint action fails the first statistical test, leaving only the fallback. Regret is K.

Projecting the seed instead gives $\mathcal{Z}^0_{1,1} = \mathcal{Z}^0_{2,1} = \{00\}$. Its first local certificate applies to both 000 and 001, even though the first constraint holds with equality there. The fallback and the candidate have the same second-constraint feature, namely $(-1, 0, 0, 0)^\top$. After n fallback observations, the candidate's second upper confidence bound is

$$U^g_{k,2,1}(01) = -\frac{n}{1+n} + \frac{\sqrt{2}}{\sqrt{1+n}}.$$

This expression is positive for $n = 0, 1, 2$ and negative for every $n \geq 3$, since $n^2 > 2(1+n)$ holds exactly for integers $n \geq 3$. The local composition rule therefore selects 000 for the first three episodes and admits 001 at episode four. Before its first execution, the candidate's reward score is one, while the fallback's score is zero. After n_r observations of the candidate, its optimistic reward score is

$$\frac{n_r}{1+n_r} + \frac{1}{\sqrt{1+n_r}} \geq 1.$$

Every execution of 001 adds the same second-constraint feature, so its safety certificate remains valid. The planner consequently continues to select 001, with no tie between it and the fallback. Regret is exactly $\min\{K, 3\}$, and safety holds throughout.

The initially uncertified comparator tuple in the second constraint has margin one, and each fallback observation adds exactly its rank-one design increment. The margin and coverage conditions restricted to initially uncertified tuples therefore hold without auxiliary observations. The optimal tuple in the first constraint has zero margin but is initially certified. The restricted margin condition therefore covers this instance, whereas a positive-margin requirement on every constraint of every nonfallback comparator action would exclude it. A centralized method with the same local certificates can exploit this information. The two certification rules differ in how they represent and reuse the safe seed.

J.3 Proof of Proposition 5.6 (Local Certification)

Definitions and probability allocation. For each local constraint table, let $M_\ell = |\mathcal{Z}_\ell|$ and let $n_{k,\ell,h}(z)$ be the number of observations of entry z available before episode k. The observations have conditional mean $g_{\ell,h}(z)$ and conditionally R_g-sub-Gaussian noise. Their selection is predictable, and the known bound n_g limits the total number of observations for each pair (ℓ, h) through episode K. For positive count $n = n_{k,\ell,h}(z)$, let

$$\overline{g}_{k,\ell,h}(z) := n^{-1} \sum\nolimits_{j=1}^{n} Y^{(j)}_{\ell,h,z}, \tag{122}$$

where $Y^{(j)}_{\ell,h,z}$ is the jth observed value of entry z. Auxiliary observations contribute to this count only when their conditional mean is that entry's value. An arbitrary linear functional is not an entry observation. The scalar certificate uses

$$\alpha = \frac{\delta}{2H(F+m+1)}, \qquad L_\ell = \log\left(\frac{2M_\ell n_g}{\alpha}\right). \tag{123}$$

Here α allocates failure probability per process, and L_ℓ covers all entries and positive counts. For a positive count, define the tabular safety upper bound

$$U^{g,\mathrm{tab}}_{k,\ell,h}(z) = \overline{g}_{k,\ell,h}(z) + R_g\sqrt{\frac{2L_\ell}{n_{k,\ell,h}(z)}}.$$

Set $U^{g,\mathrm{tab}}_{k,\ell,h}(z) = +\infty$ at count zero. Initially certified tuples in $\mathcal{Z}^0_{\ell,h}$ require no observations.

Fix (ℓ, h, z). For each $n \in \{1, \ldots, n_g\}$, the conditional sub-Gaussian exponential bound gives

$$\mathbb{P}\left(\left|n^{-1}\sum\nolimits_{j=1}^{n} Y^{(j)}_{\ell,h,z} - g_{\ell,h}(z)\right| > R_g\sqrt{\frac{2L_\ell}{n}},\ \text{the entry is observed at least } n \text{ times}\right) \leq 2e^{-L_\ell}.$$

The sum uses the first n observations of this entry, even when observations of other entries occur between them. To justify the bound under adaptive sampling, let z_t be the entry selected at observation index t for the fixed pair (ℓ, h), and let $\widetilde{Y}_t$ be its response. Define its cumulative selected count and centered response sum by

$$N_t(z) := \sum_{v=1}^{t} \mathbf{1}\{z_v = z\}, \qquad S_t(z) := \sum_{v=1}^{t} \mathbf{1}\{z_v = z\}[\widetilde{Y}_v - g_{\ell,h}(z)]. \tag{124}$$

If fewer than n_g observations are acquired, extend the count and sum processes to n_g positions with zero increments. For each real tilting parameter u, conditional sub-Gaussianity and predictable selection make $\exp\{uS_t(z) - u^2 R_g^2 N_t(z)/2\}$ a nonnegative supermartingale with initial value one. Stop it when $N_t(z)$ first reaches n, or at the final observation bound n_g if this has not occurred. For $R_g > 0$, choose $u = \sqrt{2L_\ell/n}/R_g$ for the upper tail and its negative for the lower tail. On each corresponding deviation event at count n, the stopped exponential is at least e^{L_ℓ}. The supermartingale expectation bound therefore gives e^{-L_ℓ} for each tail and $2e^{-L_\ell}$ in total. If $R_g = 0$, observed responses equal their conditional means almost surely, so the claim is immediate.

A union bound over the M_ℓ entries and their at most n_g positive counts has failure probability at most $2M_\ell n_g e^{-L_\ell} = \alpha$. Thus, simultaneously for every episode-start data set,

$$|\overline{g}_{k,\ell,h}(z) - g_{\ell,h}(z)| \le R_g \sqrt{\frac{2L_\ell}{n_{k,\ell,h}(z)}} \quad \text{if } n_{k,\ell,h}(z) \ge 1. \tag{125}$$

There are Hm such table events. Retain the reward and transition parameter-confidence events with failure probability α for each of their $HF + H$ processes. The total confidence failure probability is at most $H(F + m + 1)\alpha = \delta/2$. Denote the combined event by $\mathcal{E}_{\mathrm{tab}}$.

On this event every statistical local certificate is sound. Unobserved entries require an initial certificate or the whole fallback. Every candidate satisfies the true constraints, so the zero-violation proof applies to the modified sets.

Counting unavailable comparator episodes. Let

$$q_\ell = \max\left\{1, \left\lceil \frac{8R_g^2 L_\ell}{\Delta_{\mathrm{safe}}^2} \right\rceil \right\}.$$

The integer q_ℓ is a sufficient sample count used only in the analysis. Evaluating the statistical certificate requires neither this threshold nor the unknown margin Δ_{safe}. An action outside the candidate set is not the fallback and fails at least one local test at a tuple outside $\mathcal{Z}_{\ell,h}^0$. For a comparator action, this tuple has $g_{\ell,h}(z) \le -\Delta_{\mathrm{safe}}$ by the restricted margin condition. If $n_{k,\ell,h}(z) \ge q_\ell$, its count is positive, and (125) implies

$$U_{k,\ell,h}^{g,\mathrm{tab}}(z) \le g_{\ell,h}(z) + 2R_g \sqrt{\frac{2L_\ell}{n_{k,\ell,h}(z)}} \le -\Delta_{\mathrm{safe}} + \Delta_{\mathrm{safe}} = 0.$$

For $R_g = 0$, one observation suffices and the confidence term is zero. Every failed test at an initially uncertified comparator tuple therefore has $n_{k,\ell,h}(z) < q_\ell$.

Assign each episode with $\chi_k = 1$ to one witness (ℓ, h, z) whose count increases by at least one during that episode, as required by the proposition's coverage condition. The count never decreases. A fixed triple can be assigned at most q_ℓ episodes, because its pre-episode counts at those assignments are distinct integers in $\{0, \ldots, q_\ell - 1\}$. Only initially uncertified entries can be assigned. Hence

$$B_K \le \sum_{h=1}^{H} \sum_{\ell=1}^{m} |\mathcal{Z}_\ell \setminus \mathcal{Z}_{\ell,h}^0|\, q_\ell \le H \sum_{\ell=1}^{m} M_\ell q_\ell. \tag{126}$$

For one-hot table observations, let e_z be the coordinate unit vector of $\mathbb{R}^{M_\ell}$ indexed by z, and write $n_{k,\ell,h} = (n_{k,\ell,h}(z))_{z \in \mathcal{Z}_\ell}$ for the count vector. The design is $I + \mathrm{diag}(n_{k,\ell,h})$. Its increment dominates $e_z e_z^\top$ precisely when the count of entry z increases by at least one. Thus the count condition is the one-hot specialization of the covariance-growth condition in (25). The counting argument therefore requires actual observations of each charged entry.

Regret. The proof of Lemma C.3 requires sound candidate sets and availability of comparator actions on their reachable support. It does not require ellipsoidal safety bounds. The regret decomposition and the reward and transition potential bounds are also unchanged. Intersect $\mathcal{E}_{\mathrm{tab}}$ with the trajectory concentration event of Lemma C.4, whose failure probability is $\delta/2$. The intersection has probability at least $1-\delta$. On that event the zero-violation and regret statements hold jointly, and substituting (126) gives

$$\begin{aligned}\mathrm{Reg}(K) \le 2\beta^r\sqrt{2FHD_rT\log(1+K)} + 2H\beta^P\sqrt{2Hd_PT\log(1+K/d_P)} \\ + H\sum_{h=1}^{H}\sum_{\ell=1}^{m}|\mathcal{Z}_\ell \setminus \mathcal{Z}^0_{\ell,h}|\,q_\ell + H\sqrt{2T\log(2/\delta)}.\end{aligned}$$

Replacing the middle sum by $H\sum_\ell M_\ell q_\ell$ yields the stated bound. Every auxiliary safety observation contributes to these counts, while regret measures only the loss in environment returns.

Local representation size. The product form of the joint action space gives

$$M_\ell = |\operatorname{proj}_{I_\ell}\mathcal{S}|\prod_{i\in I_\ell}|\mathcal{A}_i|.$$

Define the maximum local state-space size $S_{\max} := \max_i|\mathcal{S}_i|$ and $r_g := \max(\{0\}\cup\{|I_\ell| : \ell \in [m]\})$, the largest constraint-scope size. Since $|\mathcal{A}_i| \le A_{\max}$, $\sum_\ell M_\ell \le m(S_{\max}A_{\max})^{r_g}$. The certification count is therefore governed by distinct local configurations. For a one-state binary-action chain with pairwise constraints, there are $4H(N-1)$ local entries across all steps, compared with $H(N-1)2^N$ entries in an unstructured table of each constraint over full joint actions. These are representation and certification counts. They do not bound state-space planning, transition representation, or induced action-graph width.

J.4 Proof of Corollary 5.7 (Feasible Deviations)

For a safe protocol Π, the feasible unilateral-deviation class $\mathfrak{D}_i(\Pi)$ contains the identity replacement and consists only of protocols in $\mathfrak{P}_{\mathrm{safe}}$. In particular, feasibility applies to the whole trajectory induced by the deviated message and action maps. The quantity $\mathrm{Gap}_i(\Pi)$ is the supremum of the common-return improvement over this class. The identity replacement gives $\mathrm{Gap}_i(\Pi)\ge 0$. Every $\widetilde{\Pi}\in\mathfrak{D}_i(\Pi)$ satisfies

$$V_1^{\widetilde{\Pi}}(\rho) - V_1^{\Pi}(\rho) \le V_1^{\Pi^\star}(\rho) - V_1^{\Pi}(\rho),$$

because $\Pi^\star$ maximizes the same common payoff over $\mathfrak{P}_{\mathrm{safe}}$. Taking the supremum over deviations, then the maximum over agents, proves the pointwise gap inequality.

On the simultaneous confidence and regret event, soundness of every candidate set ensures that each learned protocol Π^k belongs to $\mathfrak{P}_{\mathrm{safe}}$. Applying the pointwise inequality to each such protocol and summing gives

$$\sum_{k=1}^{K}\max_i \mathrm{Gap}_i(\Pi^k) \le \sum_{k=1}^{K}\left[V_1^{\Pi^\star}(\rho) - V_1^{\Pi^k}(\rho)\right] = \mathrm{Reg}(K).$$

The inequality holds on the same joint event. Consequently, the average maximum feasible-deviation gap is at most $\mathrm{Reg}(K)/K$. For any $\epsilon>0$, at most $\mathrm{Reg}(K)/\epsilon$ episodes have maximum gap greater than ϵ. These gaps concern safe admissible protocol deviations and values from ρ. Equilibrium conditions at off-path states are not implied.

Feasible equilibria can have strictly suboptimal team value. Consider one state, one step, two binary-action agents, and no pre-action messages. Let the common reward and constraint be

$$r(a_1,a_2) = \frac{a_1+a_2}{2}, \qquad g(a_1,a_2) = |a_1-a_2| - \frac{1}{2}.$$

The safe set is exactly $\{(0,0),(1,1)\}$, and both actions have safety margin $1/2$. The reward is the sum of two local factors, and the constraint has a one-hot linear realization. All model values can be known in advance, and $(0,0)$ is a globally safe fallback.

At $(0,0)$, changing either agent's action alone produces a mismatch and violates the constraint. Each feasible unilateral-deviation class therefore contains only the identity action, and both unilateral gaps are zero. The admissible team optimum $(1,1)$ has value one, while $(0,0)$ has value zero. Repeating $(0,0)$ gives zero cumulative feasible-deviation gap but team regret K. Thus small feasible unilateral gaps do not imply team optimality. Team optimality implies unilateral stability for the shared reward.

K Experiments on Learning, Certification, and Coordination

K.1 Evaluation criteria and observation models

We evaluate how local interaction structure and informative feedback affect safe learning, together with the computational and communication costs of implementing the certified planner. The synthetic models permit exact evaluation of the optimal safe policy and latent constraints, with explicit control over each estimator's observations. The experiments examine learning at fixed induced action width, certification under different feedback models, finite-horizon learning with unknown dynamics, exact planning, and communication with quantized scores. Learning studies use noisy trajectories, planning studies measure execution time, and routing studies count the packets required by a specified protocol.

For each learning run, we compute the value of the episode's frozen policy exactly under the true simulator. The reported cumulative gap is

$$G_K = \sum_{k=1}^{K} \big[V_1^{\Pi^\star}(\rho) - V_1^{\Pi^k}(\rho)\big]. \tag{127}$$

For safe methods this is the regret in (4). Exact policy evaluation removes Monte Carlo error from a separate test sample. For the unconstrained diagnostic, we retain the signed gap and report violations of the true functions $g_{\ell,h}$ separately. We also record B_K on the comparator's reachable support and the fraction of true-safe state–action entries admitted by the certified candidate set. Every run is included regardless of its return or safety outcome.

Learning experiments use 30 independent seeds and $K = 5000$ episodes. At a fixed plotted episode, let y_j be the metric from run j. The sample mean, sample variance, and reported pointwise approximate 95% interval are, respectively,

$$\bar{y} = \sum_{j=1}^{30} y_j/30, \qquad \widehat{\sigma}_y^2 = \sum_{j=1}^{30} (y_j - \bar{y})^2/29, \qquad \bar{y} \pm t_{29,0.975}\widehat{\sigma}_y/\sqrt{30}. \tag{128}$$

Here $t_{29,0.975}$ is the 0.975 quantile of the Student-t distribution with 29 degrees of freedom. These intervals describe variability across independent runs. They are not simultaneous confidence bands and do not estimate the algorithm's high-probability safety event. The chain sweep varies both the model parameters and observation noise across seeds, whereas the finite-horizon model is fixed and its initial states, transition draws, and observation noise vary. Matched methods use the same seed-indexed random streams. Timing plots instead show medians and empirical interquartile ranges over 25 instances. Deterministic routing counts and analytical bounds have no sampling intervals.

Shared safety model. Both learning environments have binary actions and a common observed mode $s \in \{0, 1\}$, with global state $(s, \dots, s)$. Each agent observes the mode locally. There are $F = N$ singleton reward scopes, with factor i assigned to agent i. Action $a \in \{0, 1\}$ selects power $p_s(a) = p_s^0 + (1 - p_s^0)a$, where $p_0^0 = 0.1$ and $p_1^0 = 0.5$ are the low-power levels and action one selects unit power. Each neighboring pair $I_\ell = \{\ell, \ell + 1\}$ has constraint residual, feature, and parameter

$$\begin{aligned} g_{\ell,h}(s, a_\ell, a_{\ell+1}) &= c_{\ell,h}\big[p_s(a_\ell) + p_s(a_{\ell+1})\big] - b_{\ell,h}, \\ \phi_{\ell,h}^g &= \frac{(p_s(a_\ell) + p_s(a_{\ell+1}), -1)^\top}{\sqrt{5}}, \qquad \theta_{\ell,h}^g = \sqrt{5}(c_{\ell,h}, b_{\ell,h})^\top. \end{aligned} \tag{129}$$

Here $c_{\ell,h}$ is the constraint gain and $b_{\ell,h}$ its threshold. Both are unknown to the learner. The feature norm is at most one and the parameter bound is $B_g = 5$. The chosen parameter ranges make $(0, 0)$, $(0, 1)$, and $(1, 0)$ safe and $(1, 1)$ unsafe in both modes. Initially, only the local $(0, 0)$ tuples and the whole all-zero fallback are supplied as safe. Each step provides noisy measurements of all constraints and reward factors. Independent Gaussian noise uses its stated standard deviation as the sub-Gaussian bound. We use $\lambda = 1$, $\delta = 0.05$, and the radii (32).

K.2 Regret versus the number of agents at fixed induced action width

We first consider the $H = 1$ specialization, with $N \in \{4, 8, 16, 32, 64\}$ and independent equiprobable modes. For each seed, c_ℓ is uniform on $[0.95, 1.05]$, b_ℓ/c_ℓ is uniform on $[1.72, 1.78]$, and the reward weight $w_{i,s}$ is uniform on $[0.4, 1]$. These parameters are drawn independently. Let $e_0 = (1, 0)^\top$ and $e_1 = (0, 1)^\top$ be the two mode-coordinate vectors. The reward factors, features, and parameters are

$$r_i(s, a_i) = \frac{w_{i,s}}{N} p_s(a_i), \qquad \phi_i^r(s, a_i) = p_s(a_i) e_s, \qquad \theta_i^r = \frac{(w_{i,0}, w_{i,1})^\top}{N}. \tag{130}$$

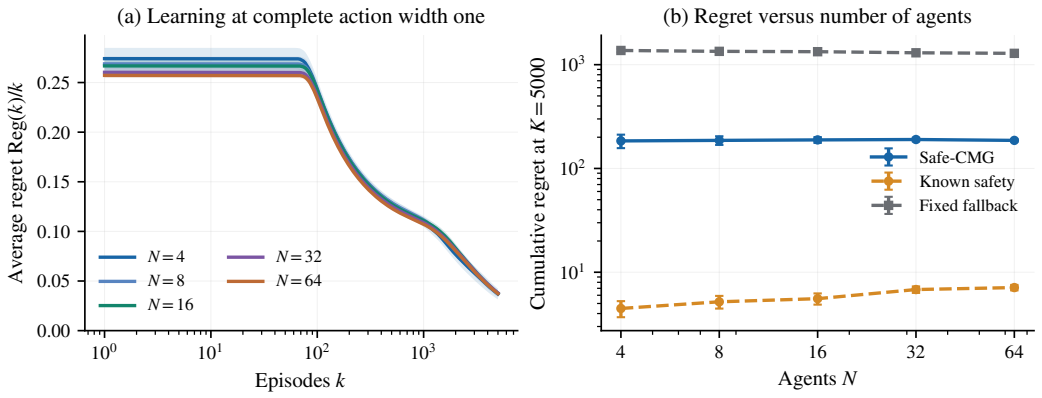


Figure 4: Learning on chains whose complete action-factor graph has induced width one. (a) Average regret decreases on the tested chain instances. (b) Cumulative regret at $K = 5000$ for the learned certificates, known-safety comparison, and fixed fallback. Bands and bars are pointwise approximate 95% intervals over 30 seeds. Rewards are normalized by N. Each step provides all local factor observations, so scalar feedback increases with N.

Table 5: Chain learning at $K = 5000$. Regret is mean $\pm$ approximate 95% interval half-width. The mean count B_K measures episodes with an uncertified comparator. All executed actions were safe.

N	$\mathrm{Reg}(K)$, learned safety	$\mathrm{Reg}(K)$, known safety	B_K
4	184.3 ± 27.4	4.48 ± 0.79	2385.9
8	186.4 ± 17.8	5.19 ± 0.73	2745.1
16	188.4 ± 12.2	5.56 ± 0.69	3254.8
32	190.1 ± 7.7	6.81 ± 0.49	3969.5
64	186.4 ± 5.1	7.12 ± 0.43	4233.5

Thus the team reward satisfies $r \in [0, 1]$, $d_{r,i} = 2$, and the common reward-parameter bound is $B_r = \sqrt{2}/N$. Reward and constraint noise have standard deviations $0.05/N$ and 0.02, respectively. Across values of N, each seed uses a prefix of the same 256-agent parameter draw. Distinct seeds remain independent. Every optimistic score factor, including its bonus, is unary. Pairwise chain constraints therefore give induced action width one.

We implement exact suffix maximization and conditional assignment recovery. All agents use one consistent action assignment across overlapping edges. The known-safety comparison uses the same reward estimator and optimizer with the true feasible set, isolating the cost of learning safety. The all-zero fallback provides a fixed safe-policy reference. The true optimal safe policy is computed by the same exact chain recursion using the true reward and constraints.

Figure 4 and Table 5 show cumulative regret between 184.3 and 190.1 over the tested sizes, compared with 1286.0–1370.4 for the fallback. The mean per-episode gap over the final 500 episodes is 0.0011–0.0019. There were zero violations among 17,850,000 executed constraint tests across the five learned-safety configurations. These results demonstrate effective reuse of local feedback on this family without establishing a dimension-independent worst-case bound. Increasing N supplies more factor observations per episode, and the reward normalization keeps the team return bounded. Moreover, B_K increases with N even though regret changes little. Full comparator inclusion requires every relevant local tuple to be certified, while a remaining blocked tuple can have a small value cost. The sufficient bound HB_K can therefore exceed the instance's actual certification loss.

K.3 Safe feedback and the certification bottleneck

To isolate information coverage, we retain the $N = 8$ chain models and set $\rho(s = 0) = 1$. Under ordinary trajectory feedback, the learner remains at the all-zero fallback in every run. Write $x(p) = (p, -1)^\top/\sqrt{5}$ for the feature at total pair power p. The observed safety features are collinear with

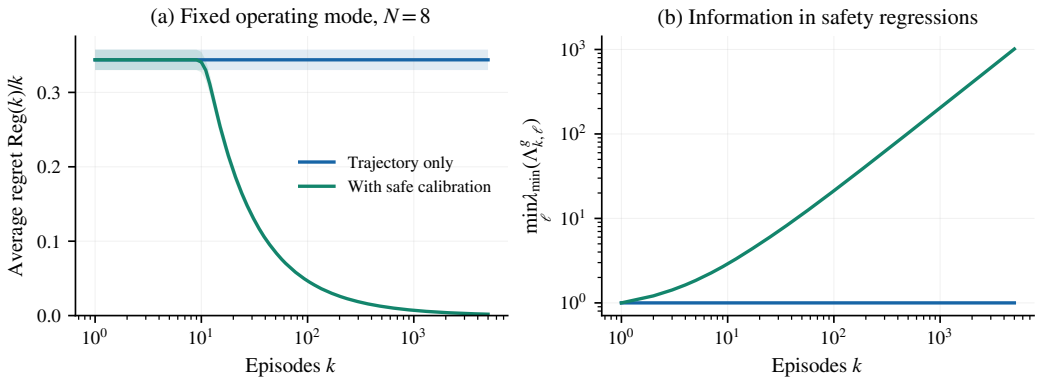


Figure 5: Coverage control on the fixed-mode $N = 8$ chain. The trajectory-only learner remains at the fallback, whereas eight additional joint safe-constraint queries per episode make both safety directions observable. Curves include all 30 seeds. The Gram eigenvalues include the identity regularizer. The calibration improvement uses nine times the total number of constraint readings.

$x(0.2)$, so the smallest eigenvalue of each regularized safety design remains one. For a fixed constraint, write c and b for its gain and threshold, $x_0 = x(0.2)$ for the fallback feature, $x_1 = x(1.1)$ for a mixed-action feature, and $g_0 = 0.2c - b$ for the fallback constraint value. After n noiseless fallback observations, the design is $\Lambda_n = I + nx_0x_0^\top$ and its ridge estimate satisfies

$$\widehat{\theta}_n = \frac{ng_0x_0}{1 + n\|x_0\|_2^2}, \qquad \lim_{n\to\infty} \|x_1\|_{\Lambda_n^{-1}} = \frac{1.1 - 0.2}{\sqrt{5(1 + 0.2^2)}}. \tag{131}$$

The parameter ranges give $g_0 \geq 1.05(0.2 - 1.78) = -1.659$. Since the prediction decreases toward its limit, the confidence norm is bounded below by its limit, and every radius is at least $B_g = 5$, the mixed-action upper certificate satisfies, for every $n \geq 0$ in this noiseless calculation,

$$U_n^g(x_1) \geq -1.659\,\frac{1 + 0.2(1.1)}{1 + 0.2^2} + \frac{5(1.1 - 0.2)}{\sqrt{5(1 + 0.2^2)}} > 0.027. \tag{132}$$

Here $U_n^g(x_1) = \langle x_1, \widehat{\theta}_n\rangle + \beta(\Lambda_n; B_g, R_g)\|x_1\|_{\Lambda_n^{-1}}$. Thus reducing noise along the observed direction does not resolve the unobserved safety direction.

We compare this regime with an explicit auxiliary safe-feedback oracle. After each episode it returns four fresh noisy constraint observations at the all-zero fallback in each of the two prescribed modes. These are additional simulator queries, not unobserved samples implicitly assigned to the learner. Each episode uses eight joint queries, or $8(N - 1) = 56$ additional scalar constraint readings at $N = 8$, in addition to the seven readings from the actual trajectory. This oracle requires access to the known-safe configuration at either mode. Ordinary trajectory feedback does not provide it.

The calibration coefficients α_p, β_p represent the feature of every comparator-safe total power $p \in [0.2, 1.5]$ as

$$\begin{aligned} x(p) = \alpha_p x(0.2) + \beta_p x(1), \quad \alpha_p = \frac{1 - p}{0.8}, \quad \beta_p = \frac{p - 0.2}{0.8}, \\ \alpha_p^2 + \beta_p^2 \leq 3.03125 < 4, \\ 4\big[x(0.2)x(0.2)^\top + x(1)x(1)^\top\big] \succeq x(p)x(p)^\top. \end{aligned} \tag{133}$$

The convex squared coefficient norm has endpoint maximum $97/32 = 3.03125$. Applying Cauchy–Schwarz to $v^\top x(p)$ for each $v \in \mathbb{R}^2$ gives the displayed matrix inequality. Thus the calibration samples provide the design growth in Assumption 5.3 at the stated feedback cost.

With ordinary feedback, mean cumulative regret is 1719.0 and $B_K = 5000$. With calibration, mean cumulative regret is 9.27 and mean B_K is 12.2. Both methods have zero executed violations. Figure 5 relates this difference to the safety-design eigenvalues. The comparison identifies the missing information that prevents exploration despite a safe fallback and a positive margin. The calibrated result requires the stated auxiliary oracle, whose availability depends on the application.

K.4 Finite-horizon learning with unknown dynamics

We evaluate sequential learning with $N = 4$ agents, horizon $H = 3$, and $K = 5000$ episodes. The state space is $\mathcal{S} = \{(0, \dots, 0), (1, \dots, 1)\}$. Every agent observes the common mode $s \in \{0, 1\}$, and the initial mode is uniform. We use 30 independent runs, with seeds 35001–35030 and paired random-number streams across methods. All environment parameters below are fixed across runs. Reported intervals follow Appendix K.1.

Rewards and constraints. Index agents by $i = 1, \dots, 4$, chain edges by $\ell = 1, 2, 3$, and stages by $h = 1, 2, 3$. Actions are binary. The local mean reward is

$$r_i(s, a_i) = \frac{0.05 + 0.65s + [0.25 + 0.13(-1)^{i-1+s}]a_i}{N}. \tag{134}$$

Let $u_1, \dots, u_4$ be the standard coordinate vectors of $\mathbb{R}^4$. The reward feature $\phi^r_{i,h}(s, a_i) = u_{2s+a_i+1}$ selects one of four parameter coordinates, whose values are $r_i(0, 0)$, $r_i(0, 1)$, $r_i(1, 0)$, and $r_i(1, 1)$, respectively. The mean team reward lies in $[0, 1]$, and each reward parameter has norm at most $B_r = 2/N$. Independent Gaussian factor noise has standard deviation $R_r = 0.05/N$.

In the power-constraint construction (129), low power is $p^0_s = 0.1 + 0.4s$ and high power is one. For $p_i = p^0_s + (1 - p^0_s)a_i$, set

$$\begin{aligned} c_{\ell,h} &= 1 + 0.04 \sin(0.7 + (\ell - 1) + 2(h - 1)), \\ b_{\ell,h} &= c_{\ell,h}[1.72 + 0.06 \cos(0.2 + 2(\ell - 1) + (h - 1))], \\ g_{\ell,h} &= c_{\ell,h}(p_\ell + p_{\ell+1}) - b_{\ell,h}, \qquad \phi^g_{\ell,h} = \frac{(p_\ell + p_{\ell+1}, -1)^\top}{\sqrt{5}}. \end{aligned} \tag{135}$$

Both coordinates of $\theta^g_{\ell,h} = \sqrt{5}(c_{\ell,h}, b_{\ell,h})^\top$ are unknown and estimated, with $B_g = 5$ and $\|\phi^g_{\ell,h}\|_2 \le 1$. Constraint measurements have independent Gaussian noise with $R_g = 0.05$. The true feasible actions have no adjacent active agents. Every all-low local tuple is initially certified, and the whole all-low action is the fallback at both modes and every stage.

Action-dependent transitions. Let $\bar{a} = N^{-1} \sum_i a_i$ be the fraction of agents using high power. With known weights $w_i = 1 + 0.3 \cos(2\pi(i-1)/N)$, the weighted fraction is $\bar{a}_w = \sum_i w_i a_i / \sum_i w_i$. Two known transition kernels assign probability

$$\begin{aligned} P^{(0)}(s' = 1 \mid s, \mathbf{a}) &= 0.75 + 0.15s - 0.65\bar{a}, \\ P^{(1)}(s' = 1 \mid s, \mathbf{a}) &= 0.55 + 0.30s - 0.45\bar{a}_w. \end{aligned} \tag{136}$$

The unknown kernel is $P_h = \nu_h P^{(0)} + (1 - \nu_h)P^{(1)}$, with true weights $(\nu_1, \nu_2, \nu_3) = (0.75, 0.45, 0.65)$. The probability of mode zero is the complement of the displayed probability. The known feature and unknown parameter are

$$\phi^P_h(s' \mid s, \mathbf{a}) = \big(P^{(0)}(s' \mid s, \mathbf{a}), P^{(1)}(s' \mid s, \mathbf{a})\big)^\top / \sqrt{2}, \qquad \theta^P_h = \sqrt{2}(\nu_h, 1 - \nu_h)^\top. \tag{137}$$

The known parameter domain is $\Theta^P_h = \{\sqrt{2}(v, 1 - v)^\top : 0 \le v \le 1\}$. It implies $B_P = \sqrt{2}$ and $\|\psi^V_h\|_2 \le \|V\|_\infty$, because each coordinate of $\sqrt{2}\psi^V_h$ is an expectation of V under a probability kernel. Strictly positive transition probabilities make both modes reachable at every stage.

Estimation and comparisons. We implement (31)–(32) with $\delta = 0.05$, $F = 4$, $m = 3$, and the normalized transition noise bound $R_P = 1$. Estimates and values are frozen before each episode. Transition data retain their original historical value functions (10). Planning maximizes the unclipped score with lexicographic tie-breaking and then clips its selected value. The complete transition bonus has a global action scope. We therefore enumerate all 16 joint actions. This experiment does not measure bounded-width planning scalability.

The known-safety learner receives the true feasible sets but otherwise uses the same reward and transition learning. The full-joint certification ablation uses the same reward and transition models and feedback access but estimates each $(h, s, \mathbf{a}, \ell)$ constraint mean independently. For an entry observed n times, its upper bound is the empirical mean plus $R_g\sqrt{2\log(2|\mathcal{S}||\mathcal{A}|K/\alpha)/n}$ for $n > 0$, and infinity for unobserved entries. The whole fallback remains available. This ablation isolates reuse of local constraint observations and does not represent all centralized algorithms. Other comparisons use the fixed fallback and unconstrained optimistic learning.

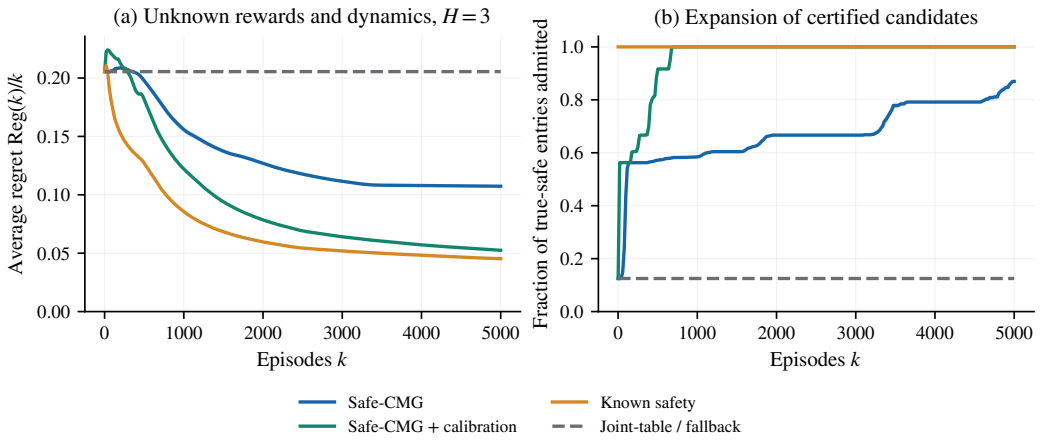


Figure 6: Finite-horizon learning with unknown rewards, dynamics, and coupled constraints. Curves use all 30 runs. Uncertainty bands are 95% Student-t pointwise intervals. Calibration has the additional feedback cost stated in the text.

Acquiring certification information. Alongside trajectory-only feedback, we evaluate an explicit safe calibration oracle. After each episode, it supplies four independent all-low constraint measurements in each mode, at every stage. These require 24 joint oracle calls, or 72 scalar constraint readings, per episode. All readings enter the regressions. Including trajectory feedback gives 405,000 scalar readings per calibrated run, compared with 45,000 without calibration. Writing $x(p) = (p, -1)^\top/\sqrt{5}$, the calibration increment satisfies

$$4x(0.2)x(0.2)^\top + 4x(1)x(1)^\top \succeq x(p)x(p)^\top, \qquad 0.2 \le p \le 1.5. \tag{138}$$

This increment covers every feasible local tuple and supplies the design growth in Assumption 5.3. Merely visiting both fallback modes does not guarantee this episode-wise condition, so we check coverage witnesses for every blocked episode and record B_K.

Results. Exact dynamic programming gives $V_1^{\Pi^\star}(\rho) = 1.7323861914$ and $V_1^{\Pi^0}(\rho) = 1.5268929688$. The safe optimum uses all-low actions at stage one and mode-dependent activation later, reflecting the effect of current actions on future modes. We evaluate every learned policy exactly under the true model. Figure 6 and Table 6 report these policy gaps and latent constraint violations, rather than violations of noisy measurements.

All certified methods had zero observed violations. Calibration first admitted every comparator action between episodes 639 and 680 ($B_K \in [638, 679]$), and every blocked episode had a coverage witness. Without calibration, $B_K = 5000$ in all runs, despite admitting on average 86.94% of true-safe state/action entries by the end. These trajectory-only results show an improvement over fallback, but do not establish sublinear regret. The calibrated learner approaches the known-safety learner's performance over the final 500 episodes at the stated additional observation cost. The unconstrained diagnostic incurred 597,995 violated scope–step tests over 238,840 steps, with violations in all 30 runs. Its signed gap relative to the safe comparator is retained and is not interpreted as safe regret.

K.5 Exact planning and interaction structure

We isolate the computational effect of action-factor structure using binary actions and the complete score $\widetilde{Q}(\mathbf{a}) = \sum_{i=1}^{N} w_i a_i$, where the weights are independent uniform integers in $\{1, \dots, 1000\}$. Each edge (i, j) imposes the exact hard constraint $a_i a_j = 0$, represented by a $0/-\infty$ table. The all-zero action is the known-safe fallback and is scored separately. We compare exhaustive joint-action enumeration with exact variable elimination and conditional backtracking. Both optimize the same materialized score and feasibility tables, with descending elimination order, lexicographic action ties, and fallback priority on a score tie. Thus this comparison changes the planning implementation while preserving its objective and feasible set. It does not compare different learning rules. Conditional backtracking gives every shared variable one action that satisfies all its constraint scopes.

Table 6: Mean gaps ± approximate 95% interval half-widths across 30 runs. B_K is a mean. Final gap averages episodes 4501–5000 within each run. Violations count all violated latent scope–step constraints across runs. The unconstrained row reports a signed safe-comparator gap.

Method	Gap G_K	Final gap	B_K	Violations
Safe-CMG	536.8 ± 3.2	0.1053 ± 0.0009	5000.0	0
Safe-CMG + calibration	262.4 ± 1.2	0.0331 ± 0.0008	658.4	0
Known safety	226.3 ± 1.2	0.0330 ± 0.0007	0.0	0
Joint-table certificate	1027.5	0.2055	5000.0	0
Fixed fallback	1027.5	0.2055	5000.0	0
Unsafe diagnostic	78.8 ± 1.7	0.0055 ± 0.0013	0.0	597,995

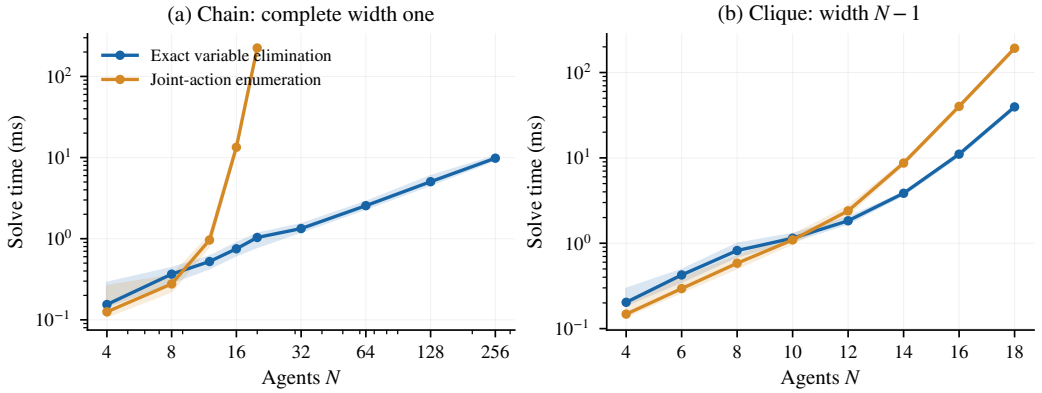


Figure 7: Exact planning on identical certified objectives. Measured runtime summaries use 25 independent instances per configuration. Enumeration above $N = 20$ is not run. The complete action-factor width is one for the chain and $N - 1$ for the clique.

The chain sizes are $N \in \{4, 8, 12, 16, 20, 32, 64, 128, 256\}$. The complete pairwise graph sizes are $N \in \{4, 6, 8, 10, 12, 14, 16, 18\}$. We generate 25 independent instances per size. Enumeration is limited in advance to $N \leq 20$ and uses chunks of at most 65,536 assignments. Larger enumeration cases are omitted, without extrapolated runtimes. All 325 paired instances agree exactly in optimal score, selected joint action, and fallback decision. All selected actions satisfy every hard constraint.

Figure 7 reports measured medians and interquartile ranges. At $N = 20$, chain elimination takes 1.035 ms (0.788–1.156 ms), compared with 224.322 ms (217.548–228.657 ms) for enumeration. Chain elimination takes 9.824 ms (9.339–10.389 ms) at $N = 256$. Its induced width remains one, with four entries in the largest elimination table and two in the largest separator message. For the $N = 18$ clique, the width is 17 and these counts become 262,144 and 131,072. Elimination takes 39.575 ms (38.216–40.599 ms). These counts describe materialized tables, not measured process memory. Width is computed from all score and feasibility factors. The clique's particular constraints admit a specialized $O(N)$ solver because at most one action can equal one. The measured growth therefore describes generic elimination on this representation and does not establish intrinsic hardness.

K.6 State inputs and physical routing

We evaluate the packet accounting of Propositions I.1 and I.2 on the same chain sizes. The reduced junction tree has $N-1$ bags $B_i = \{i, i+1\}$, rooted at its middle bag, and the physical communication graph is an N-agent path. Matched hosts assign bag B_i to agent i. Alternating hosts use the order $1, N, 2, N-1, \ldots$. Unary factor i and output action i are assigned to bag $\min(i, N-1)$, while pair factor $(i, i+1)$ is assigned to bag i. State symbols and fallback components each occupy one bit. Repeated requests for the same source–host pair are merged.

Local unary factors use state scope $\{i\}$, whereas global unary factors use $[N]$. Pair-factor state scopes remain $\{i, i+1\}$. With matched hosts, the resulting raw-state traffic is $N-1$ bit-hops locally and $N(N-1)(2N-1)/6$ bit-hops globally. At $N = 32$, these counts are 31 and 10,416

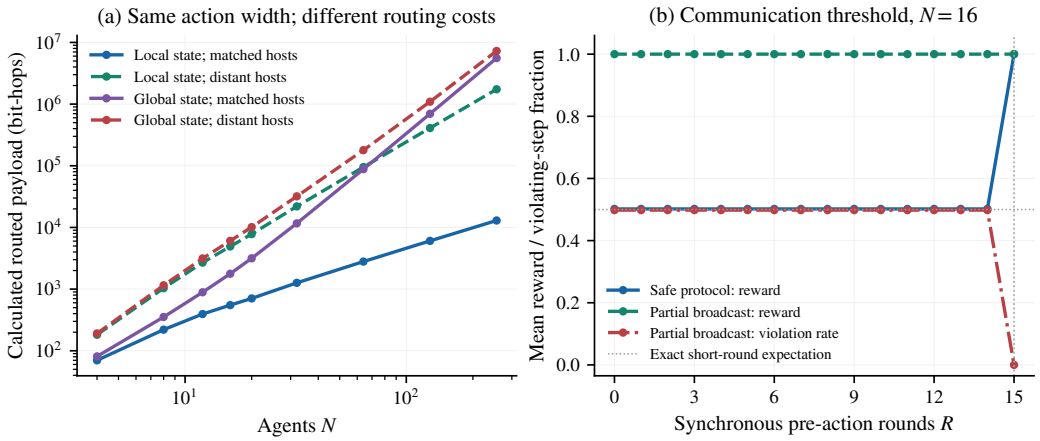


Figure 8: Communication costs and implementability. (a) Calculated total routed payload at action width one, varying state-input scopes and bag placement. These are protocol counts, not network timings. (b) Actual synchronous execution on the 16-agent path. Safe commitment has zero violations at every budget. Partial broadcast has high reward but violates equality constraints below 15 rounds. Means and approximate pointwise 95% intervals use 30 independent seeds. The dotted horizontal line marks the exact short-round expectation, rather than a fitted value.

despite identical action width one. Alternating hosts increase routing distances without changing the logical factorization. Figure 8 therefore separates action structure, state acquisition, and physical placement. These counts quantify traffic for the specified raw-input protocol. They are neither minimum communication requirements nor measurements of network latency. The plotted totals include state and fallback inputs, upward scores, downward assignments, and final outputs. At $N = 32$, matched hosts require 1264 bit-hops with local state inputs and 11,649 with global inputs.

For score traffic we use $|q_i| \le 1/N$, desired optimization-error bound $\epsilon = 0.01$, and score grid $\zeta = \epsilon/N$. The locally known fallback requires $R_0 = 0$ rounds. Each upward packet contains two separator scores and one fallback scalar. Downward packets and final outputs each contain a branch flag and one action bit. The data also record $r_{\text{in}} + 2D_G + r_{\text{out}}$ for the unrestricted vector-message model and a separate sufficient serial schedule with 64-bit edge capacity. The two bounds correspond to different edge-capacity models and do not measure network latency.

A limited-round protocol. We separately simulate the known-model path game in Proposition H.1, with $N = 16$. A fresh fair bit X is observed only by agent one. The reward is $\mathbf{1}\{a_1 = X\}$ and the hard constraints require all adjacent actions to agree. Information is forwarded synchronously by one edge per round, using only messages received in previous rounds. The safe protocol executes the whole all-zero fallback when the publicly known budget is $R < 15$. At $R = 15$ every agent executes the received bit. For comparison, an unsafe partial-broadcast protocol lets informed agents act on the bit while uninformed agents choose zero. We use 30 seeds, 10,000 independent episode bits per seed, and the same bits for both protocols and all $R \in \{0, \dots, 15\}$.

Below the threshold, the safe protocol has empirical mean reward 0.50191 and no violations, whereas partial broadcast has reward one and violates a constraint in $149{,}428$ of the $300{,}000$ episodes (fraction 0.49809). At $R = 15$ both achieve reward one without violations. The exact expectations below the threshold are $1/2$ for safe reward and $1/2$ for the unsafe protocol's violation probability. Figure 8(b) displays the sampled values separately from this expectation. The short-round fallback attains the optimal safe value within its admissible protocol class. Its shortfall relative to the full-state controller is therefore an information limitation even with a known model and exact action optimization.

K.7 Finite precision in score communication

On 25 independent $N = 32$ chains, normalize the unary scores to $q_i(a_i) = w_i a_i/(1000N)$. There are $m_Q = N$ score factors. For desired optimization-error bound ϵ, set the quantization step to

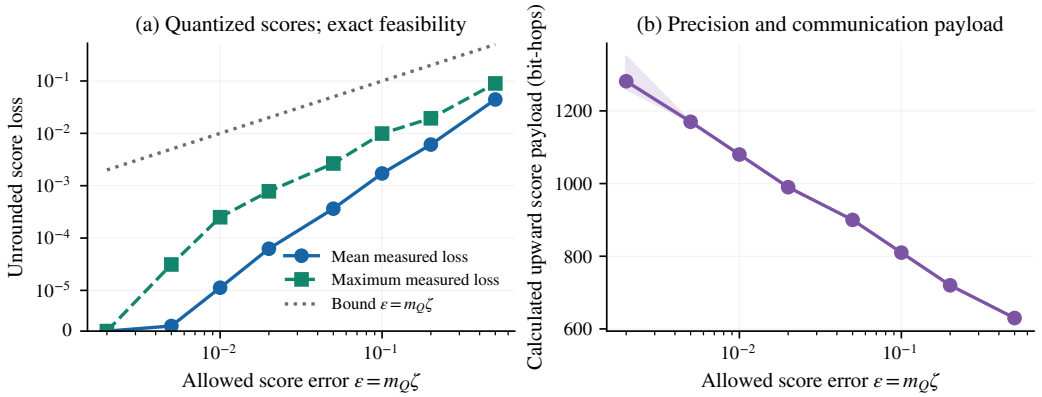


Figure 9: Original-score loss and calculated upward score payload under exact hard constraints. (a) Mean and maximum loss over 25 instances, with the deterministic bound $m_Q\zeta$. The loss axis is linear near zero and logarithmic above 10^{-5}, preserving exact zero losses. (b) Mean upward score payload with empirical minimum–maximum shading. Score rounding changes the selected value while all 200 plans remain feasible. Payload excludes input, downward-assignment, and output traffic.

$\zeta = \epsilon/N$, so $m_Q\zeta = \epsilon$. We sweep $\epsilon \in \{0.5, 0.2, 0.1, 0.05, 0.02, 0.01, 0.005, 0.002\}$, replace only soft scores by $z_i = \lfloor q_i/\zeta \rfloor$, and solve the integer objective exactly. The hard flags are unchanged, and the fallback is compared under the same integer objective. Rational arithmetic computes each floor, and integer weight sums give the original-score losses. All 200 plans remain feasible and lose at most ϵ in the original score.

Figure 9 shows mean losses of 0.0443975 at $\epsilon = 0.5$, 0.00171375 at 0.1, and 0.00001125 at 0.01. The corresponding maximum losses are 0.089875, 0.00990625, and 0.00025. No loss occurs on these instances at $\epsilon = 0.002$. Using the actual bounds $L_i = w_i/(1000N)$ in (113), the mean upward score payload is 630 bit-hops at $\epsilon = 0.5$ and 1,281.6 at 0.002. This counts $(N-2)3b_\zeta$ bit-hops on matched chain hosts, including fallback aggregates and excluding other packet types. The uniform bounds $L_i = 1/N$ in Appendix K.6 can require longer score words.

K.8 Reproducibility, resources, and empirical scope

The supplementary package contains the complete simulation and plotting code, all production seed records, model parameters, per-episode metrics, executed state/action traces, and raw timing observations. Chain learning uses seeds 24001–24030, finite-horizon learning uses 35001–35030, and path execution uses 46001–46030. Independent streams are explicitly indexed within each family. All listed runs enter the figures and summaries, with pilot seeds excluded.

The base seed is 20261105. NumPy's `SeedSequence([base,family,N,replicate])` generates each instance's seed from its first 64-bit output. Family identifiers 10, 11, and 20 denote chains, cliques, and precision trials, respectively, and replicate indices range from 0 to 24. This base-seed construction applies to the planning and precision instances. Measurements use Python 3.12.14 and NumPy 2.3.5 on an AMD EPYC 9V74, with an eight-CPU quota, 8 GiB memory limit, and BLAS/OMP thread counts set to one. One independent warmup precedes each method–size configuration. Method order alternates across replicates. Timing includes validation, optimization, backtracking, and fallback comparison. Factor construction is recorded separately. The 750 measured planning calls take 14.186 s in aggregate. The complete timing sweep, including warmups and measurement overhead, takes 15.235 s. Raw observations, omitted-case flags, source hashes, and all configurations accompany the implementation.

The twelve batched chain configurations take 84.70 s in aggregate, and the six finite-horizon configurations take 74.45 s. Each configuration contains 30 seeds and 5000 episodes. Precision, routing, and path execution take approximately 0.30 s, 0.20 s, and 1.03 s, respectively. These are sums of elapsed durations, not exclusive CPU-hour measurements. Timing measurements begin after learning jobs become idle. The shared runtime has no exclusive allocation and uses no GPU. Additional computation includes pilots, independent checks, plotting, and reconstruction of one truncated archive

with unchanged seeds. Every reconstructed summary except elapsed time matched exactly. The supplement documents the repair and validates all 19 raw archives.

These experiments use exactly specified linear models and small state spaces. The common-mode learning instances do not test distributed state acquisition. The routing and path studies use separately specified information models. Finite-window curves and zero observed violations complement the proofs but do not establish asymptotic rates or probability guarantees. No conclusion about physical deployment, misspecification, packet loss, or network latency is drawn.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The abstract and introduction state the conditions for instantaneous safety and sublinear regret, including the safe fallback, observation model, exact implementable planning, and certification coverage. The computational claim is restricted to per-state action optimization with bounded induced width of the complete action-factor graph.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations discussion addresses the globally available safe fallback, factor-level reward observations, certification coverage, exact planning and communication, and the costs of large state spaces and action graphs with large induced width. These assumptions restrict the scope of the theoretical guarantees.

   Guidelines:

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

- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

   Justification: The model and assumptions are stated before the results, and the proof appendix establishes simultaneous confidence bounds, zero violation, regret, and the lower bounds. The recursive-viability extension is distinguished from the main regret theorem and does not inherit its finite certification bound without additional conditions.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: Appendix K specifies the synthetic models, observation regimes, exact policy evaluation, comparison methods, confidence radii, seeds, and reporting conventions. The supplementary package supplies the executable implementations, all production records, and figure-generation code.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.

- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

   Answer: [Yes]

   Justification: The supplementary experimental package includes all simulation and plotting code, raw synthetic outputs, exact configurations, dependency versions, and reproduction commands. No external dataset or pretrained model is needed.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Appendix K states the model parameters, noise distributions, sample budgets, calibration costs, initialization, planning methods, and comparison conditions. Appendix K.8 and the supplement give the seeds and environment. No train/test split is used because episode policy values are evaluated exactly in the simulator.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: Learning figures report means with approximate pointwise Student-t 95% intervals over 30 independent seeds, and state the sources of variation. Timing figures show medians and interquartile ranges over 25 instances. Deterministic routing counts, precision bounds, and empirical ranges are labeled separately; no simultaneous-band or hypothesis-test claim is made.

   Guidelines:

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Appendix K.8 identifies the CPU, memory and CPU quotas, numerical thread settings, software versions, measured durations by experiment family, and additional pilot, verification, plotting, and archive-recovery computation. No GPU is used.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics. It involves no human-subject studies or personal data, and the paper states the assumptions and limitations of its safety guarantees.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [Yes]

    Justification: The broader-impact discussion identifies potential safety benefits for robotic and wireless coordination, risks from applying model-dependent certificates to unvalidated systems, and the absence of an automatic fairness guarantee.

    Guidelines:

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

11. **Safeguards**

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

    Answer: [N/A]

    Justification: The paper releases no datasets or trained models with elevated misuse risk.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.

- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

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

    Answer: [N/A]

    Justification: The experiments use original synthetic simulators and do not reuse external datasets, pretrained weights, or third-party research implementations. Standard numerical and plotting dependencies are identified by name and version in the supplement; their source code is not redistributed. Prior research is cited and language-model assistance is disclosed.

    Guidelines:

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

13. **New assets**

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

    Answer: [Yes]

    Justification: The supplementary package documents the new experimental software and synthetic outputs, including their purpose, configurations, seeds, formats, limitations, integrity checks, and reproduction commands. It contains no personal data.

    Guidelines:

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

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

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

Answer: [N/A]

Justification: The research involves no crowdsourcing or human participants.

Guidelines:

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

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

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

Answer: [N/A]

Justification: The research involves no human participants or studies requiring institutional review board approval.

Guidelines:

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

16. **Declaration of LLM usage**

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

Answer: [Yes]

Justification: ChatGPT assisted manuscript editing, mathematical and notational checks, proposed corrections and supplementary arguments, and implementation, checking, and analysis of the synthetic experiments. This assistance is disclosed in the manuscript. No LLM is a component of the proposed learning algorithm or simulated environments.

Guidelines:

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