MARCH: MARGIN-AWARE REASONING FOR CBF-HARNESSED MULTI-AGENT CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Control Barrier Functions (CBFs) are widely used to enforce safety by activating a low-level Quadratic Program (QP) to correct high-level planning decisions before execution. This CBF-QP-based architecture is attractive for LLM-based planners and works well in simple single-robot or low-constraint settings, where such corrections are only occasionally needed. However, in swarm systems with multiple coupled constraints and bounded inputs, repeated intervention can indicate that the filtered trajectory is approaching states where the hard CBF constraints are incompatible. This suggests a different perspective: maintaining CBF-QP feasibility is not only a low-level optimization problem; it also depends on whether the high-level LLM planner avoids decisions that repeatedly activate the CBF-QP. We therefore introduce *MARCH* (Margin-Aware Reasoning for CBF-Harnessed Multi-Agent Control), with *Predictive Harness-CBF* (Predictive H-CBF) as its predictive variant. It uses low-level CBF-QP feedback to revise future high-level decisions generated by an LLM-based planner. More specifically, the harness receives CBF margins and related QP feedback, organizes them together with task context and memory, and generates multiple high-level decision candidates. A predictive evaluator then rolls out these candidates over a short horizon and selects the final decision based on task progress, predicted CBF activation, and QP feasibility. The selected decision guides future high-level LLM output, while the low-level CBF-QP continues to filter the control input during execution. Experiments on 3-D swarm-control tasks show that Predictive H-CBF reduces repeated CBF activation, hard CBF-QP infeasibility, and unnecessary CBF interventions while maintaining competitive coverage performance.

1 INTRODUCTION

Control Barrier Functions (CBFs) have become a standard tool for enforcing safety in multi-agent systems (Ames et al., 2017; 2019; Wieland & Allgöwer, 2007; Prajna et al., 2007). A standard CBF-based controller receives a nominal input from a high-level planner and corrects it when safety is at risk (Ames et al., 2017; Brunke et al., 2022). This design works well in simple single-robot scenarios. However, in swarm systems, multiple constraints can lead to frequent CBF activation and even infeasibility, which is often studied through low-level constrained optimization (Wang et al., 2017; Borrmann et al., 2015; Glotfelter et al., 2017; Deng et al., 2025). Here we emphasize a different view: frequent CBF activation can indicate that the high-level planner repeatedly requests motion toward the boundary of the safe set (Xu et al., 2015; Zeng et al., 2021). Therefore, a well-designed high-level planner should avoid causing frequent CBF activation. Instead, it should maintain a sufficient safety margin whenever possible.

High-level planners in swarm systems include rule-based planners, optimization-based planners, and learning policies (Cortés et al., 2004; Schwager et al., 2009; 2011; Chung et al., 2018), which can improve task-level behavior. We study architectures in which the high-level planner does not receive the low-level CBF-QP diagnostics. In these architectures, persistent intervention can reveal a mismatch between the nominal plan and the constrained closed-loop motion; future states, rather than the nominal objective alone, determine subsequent hard-QP feasibility. This limitation naturally motivates harness engineering, especially LLM-based harnesses (Ahn et al., 2022; Liang et al., 2023; Driess et al., 2023; Huang et al., 2022a; Singh et al., 2023). An LLM-based harness can naturally incorporate feedback, memory, and external checks into future high-level decisions (Yao et al., 2023b;

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

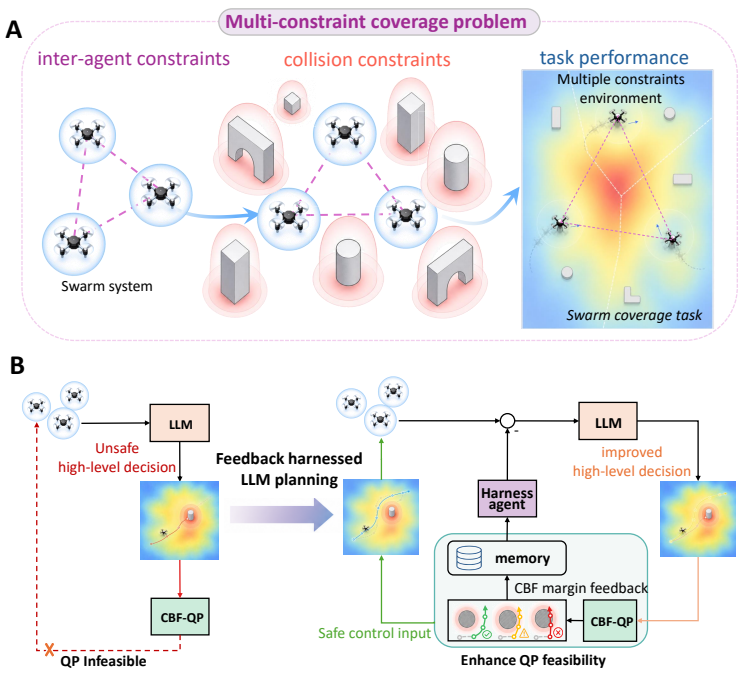

Figure 1: **Predictive Harness-CBF for multi-constraint swarm coverage.** Low-level CBF-QP feedback is returned to the high-level LLM planner through a harness, allowing future decisions to become more compatible with CBF-QP feasibility. **(A) Multi-constraint coverage problem.** Inter-agent and obstacle constraints can become incompatible under input limits as the swarm executes task-driven motion. **(B) Feedback-harnessed LLM planning.** The harness uses CBF margin feedback and memory to improve future LLM decisions, thereby increasing QP feasibility while leaving execution-time safety filtering to the CBF-QP.

Shinn et al., 2023; Huang et al., 2022b; Wang et al., 2024; Liu et al., 2025), thereby enabling a feedback loop from low-level CBF behavior to future planning.

Our initial attempt was to directly feed the CBF margin back to the LLM-based harness from the low-level control loop at subsequent planning instants (Feedback Harness-CBF), so that the harness could revise future decisions generated by the LLM. This feedback helps the LLM evaluate its previous decision and revise the next one when the margin decreases. However, direct feedback can make the system conservative. Although it can reduce CBF activation and the frequency of QP infeasibility, it can also degrade task performance. The reason is that margin feedback does not provide a short-horizon view of how the effects of different future decisions will evolve. As a result, the LLM cannot tell which future decision better balances task progress and CBF activation.

To address this conservative problem, we further propose Predictive Harness-CBF, which extends Feedback Harness-CBF with predictive candidate selection. Specifically, the LLM-based harness first proposes several high-level candidates (Yao et al., 2023a; Wei et al., 2022), instead of revising the next decision from a single margin signal. Then, a predictive evaluator rolls out these candidates over a short horizon and estimates their effects on task progress, future CBF activation, and QP feasibility (Mayne et al., 2000; Wabersich & Zeilinger, 2021; Zeng et al., 2021). Finally, the harness agent selects the candidate that best balances task progress with future CBF activation for the next planning step. Therefore, the CBF margin is no longer used only as one-step feedback after intervention. It becomes a planning signal for improving future high-level decisions, while the low-level CBF-QP continues to provide execution-time filtering. In this way, the predictive harness improves the quality of the high-level decision without changing the execution-time safety role of the CBF-QP.

Overall, this paper makes three contributions. First, we formulate a CBF-feedback interface that returns margins and solver interventions to the high-level LLM planner. Second, we propose Predictive Harness-CBF, which uses CBF feedback, memory, and short-horizon candidate evaluation

to improve LLM-generated high-level decisions while preserving the execution-time role of the low-level CBF-QP. Third, we separate the heuristic CBF burden score from hard-QP feasibility, derive a constructive sampled-horizon feasibility audit from the bounded-input dynamics, and evaluate the feedback–prediction mechanism on 3-D swarm-control tasks. The audit does not assume a uniform error bound on the aggregate score.

2 Problem Statement

We consider a system in which an LLM-based harness generates high-level decisions and a low-level CBF-QP provides safety filtering (Ames et al., 2017; 2019). At control step k, the LLM-based harness outputs high-level decision θ_k, which determines a nominal input. The low-level CBF-QP then computes a filtered input u_{safe} under state, input, obstacle, boundary, and pairwise safety constraints (Wang et al., 2017; Borrmann et al., 2015).

The limitation studied here is the absence of a reverse diagnostic-feedback link between the high-level LLM and the low-level CBF-QP. As a result, low-level CBF feedback is not used to regulate future high-level decisions, which can lead to repeated CBF activation and eventually QP infeasibility.

Our goal is to use CBF feedback, especially the CBF margin, to design an LLM-based harness that selects future high-level candidates while maintaining control performance and limiting repeated CBF activation and QP infeasibility.

3 Method

We propose *Predictive Harness-CBF* (Predictive H-CBF), which consists of three modules: CBF Feedback, LLM-Based Harness, and a Predictive Evaluator, as shown in Figure 2. Here MARCH names the overall CBF-feedback framework; Feedback H-CBF and Predictive H-CBF retain their original roles as the direct and predictive variants. The index k denotes a low-level control step; the LLM is invoked only at the configured planning steps. CBF margins condition future proposals, whereas prediction compares the consequences of alternative proposals.

3.1 CBF Feedback

The first module extracts CBF feedback from the low-level CBF-QP. At control step k, the LLM-based harness determines a nominal input $u_{\text{nom},k}$, and the low-level CBF-QP returns a filtered input $u_{\text{safe},k}$. In standard CBF-based control, only the safe input is applied to the system (Ames et al., 2017; 2019). Here, we additionally record how the low-level CBF-QP responds to the nominal input and use this information as feedback for future high-level decisions.

We collect the low-level CBF feedback into the vector

$$\phi_k = [m_k,\, r_k,\, a_k,\, s_k,\, q_k]\,, \tag{1}$$

where m_k is the CBF margin, r_k is the CBF intervention norm, a_k is the active-constraint ratio, s_k is the relaxation value, and q_k is a hard infeasibility indicator. The intervention norm is defined as

$$r_k = \|u_{\text{safe},k} - u_{\text{nom},k}\|\,. \tag{2}$$

Each term in ϕ_k has a direct meaning for future planning. A small CBF margin identifies a monitored barrier value near its safety boundary. A large intervention norm means that the low-level CBF-QP must strongly modify the nominal input. A high active-constraint ratio means that many constraints are active at the current step; repeated activation is assessed from its time history. A positive hard-infeasibility indicator means that, at the current state, no input satisfies all hard rows and the input bounds. Changing only the nominal objective cannot change that current feasible set; it changes future feasibility through the executed trajectory.

We further aggregate this feedback into a scalar CBF-QP burden score,

$$P_k = \lambda P_{k-1} + (1-\lambda)\ell(\phi_k), \tag{3}$$

where P_k is an exponential moving average, $\lambda \in [0, 1)$ is a memory factor, and $\ell(\cdot)$ maps the feedback vector to a scalar feasibility-cost score. This memory reduces noise in the feedback signal and allows the harness to react to a trend rather than to a single unsafe step.

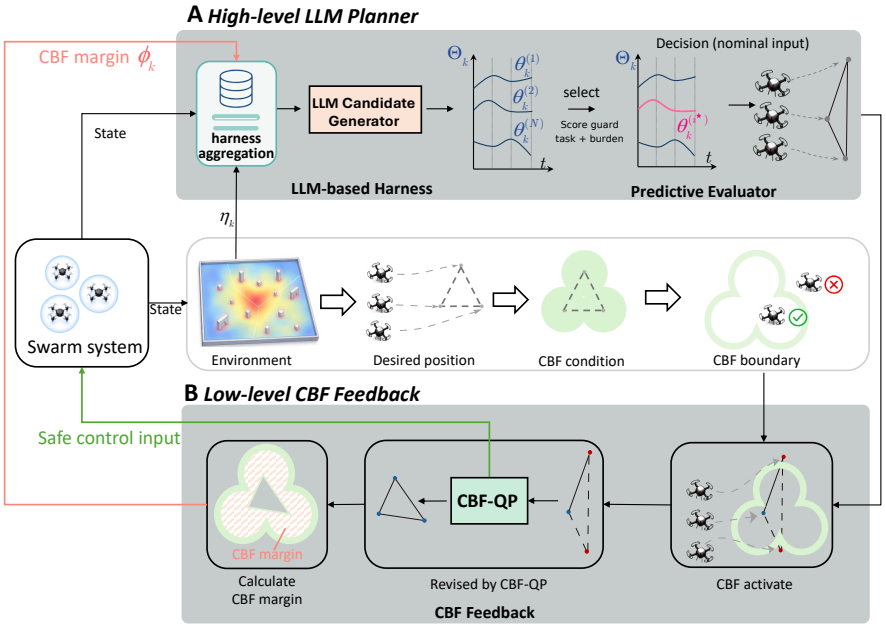

Figure 2: **Closed-loop architecture of Predictive Harness-CBF.** The framework couples high-level LLM planning with low-level CBF-QP safety filtering through a CBF-feedback loop. **(A) High-level LLM planner.** The harness receives the current swarm state and the CBF feedback, organizes them into a structured context, and invokes the LLM to generate a finite candidate set. The predictive evaluator performs short-horizon rollout for these candidates and selects the high-level decision $\theta_k^\star$. **(B) Low-level CBF feedback.** The low-level CBF-QP evaluates the nominal input under the CBF safety constraints and solves for the filtered control input. It also computes CBF margin as feedback to the high-level planner. This closed-loop information flow allows the high-level planner to revise future decisions according to low-level CBF behavior.

The output of this module is therefore not a new control action, but a compact feedback signal for future high-level decision making.

3.2 LLM-Based Harness

The second module is an LLM-based harness that uses CBF feedback to revise future high-level decisions (Yao et al., 2023b; Shinn et al., 2023; Huang et al., 2022b). The harness does not output raw control inputs. Instead, it outputs high-level parameters that determine the next nominal input (Ahn et al., 2022; Liang et al., 2023). In our UAV coverage task, we use

$$\theta_k = [\gamma_k, \eta_k^{\text{obs}}, \eta_k^{\text{pair}}, \eta_k^{\text{alt}}].$$

Here, γ_k is a speed scale, while η_k^{obs}, η_k^{pair}, and η_k^{alt} weight nominal obstacle-clearance, inter-agent-spreading, and altitude-lane terms, respectively.

A simple version of this idea is *Feedback Harness-CBF*, in which the LLM directly revises the next high-level decision from the current CBF margin. We found that this design can become conservative. When the margin decreases, the LLM may reduce speed too much or choose overly cautious targets. This can reduce repeated CBF activation and resolve some QP infeasibility cases, but it may also degrade task performance.

We therefore let the LLM-based harness propose several future high-level candidates,

$$\Theta_k = \left\{ \theta_k^{(1)}, \theta_k^{(2)}, \ldots, \theta_k^{(N_c)} \right\}. \tag{4}$$

Here N_c is the finite candidate count, distinct from the number N of UAVs. These candidates represent different possible revisions of the next high-level decision. The LLM does not directly choose the final one. Its role is to generate diverse high-level options from the current feedback.

This design separates high-level generation from final selection. The LLM provides future decision candidates, while the final choice is delegated to a predictive evaluator.

3.3 PREDICTIVE EVALUATOR

The third module selects the final high-level candidate. For each candidate $\theta_k^{(i)}$, the evaluator performs a short-horizon rollout using the same low-level CBF-QP as the real system (Mayne et al., 2000; Wabersich & Zeilinger, 2021). It estimates how the candidate will affect task performance, future CBF activation, and QP feasibility over the next few steps (Zeng et al., 2021; Hsu et al., 2023).

We separate the predicted quantities into a task-performance term and a CBF-burden term. The task-performance term $J_{\text{task}}^{(i)}$ measures the predicted task cost. The CBF-burden term is defined as

$$\widehat{P}_{\text{cbf}}^{(i)} = w_1 J_{\text{margin}}^{(i)} + w_2 J_{\text{infeas}}^{(i)} + w_3 J_{\text{int}}^{(i)} + w_4 J_{\text{active}}^{(i)}, \tag{5}$$

where $J_{\text{margin}}^{(i)}$ penalizes small predicted CBF margins, $J_{\text{infeas}}^{(i)}$ penalizes predicted QP infeasibility, $J_{\text{int}}^{(i)}$ penalizes large predicted CBF intervention, and $J_{\text{active}}^{(i)}$ penalizes repeated predicted CBF activation.

The evaluator first forms a score-eligible set

$$\mathcal{I}_k = \left\{ i : \widehat{P}_{\text{cbf}}^{(i)} \le \tau_P \right\}, \tag{6}$$

where τ_P is a preset threshold on predicted CBF burden, not an assumed bound on prediction error. Within this score-eligible set, candidates are ranked by

$$J^{(i)} = J_{\text{task}}^{(i)} + \lambda_P \widehat{P}_{\text{cbf}}^{(i)}. \tag{7}$$

The selected candidate is

$$i^\star = \arg\min_{i \in \mathcal{I}_k} J^{(i)}. \tag{8}$$

If $\mathcal{I}_k = \emptyset$, the evaluator selects the minimum-cost fallback candidate,

$$i^\star = \arg\min_{i} \widehat{P}_{\text{cbf}}^{(i)}. \tag{9}$$

The next high-level decision is then updated using $\theta_k^{(i^\star)}$.

This predictive selection rule is the key difference between Predictive Harness-CBF and Feedback Harness-CBF. Feedback Harness-CBF directly revises the next decision from the current CBF margin. By contrast, Predictive Harness-CBF compares several future high-level candidates over a short horizon before selecting the next one. This allows the LLM-based harness to choose a decision that better maintains task performance while limiting repeated future CBF activation and QP infeasibility.

Separate hard-feasibility analysis. The score guard is a search heuristic, not a certificate. Write the joint hard rows as $r_j(z,u) = b_j(z) - A_j(z)u \ge 0$, with $u \in \mathcal{U}$. At predicted state $\widehat{z}_{k,\ell}^{(c)}$, the bounded-input double-integrator model yields position/velocity error tubes and explicit row-error bounds $\eta_{j,k,\ell}^{(c)}$ (Appendix B.2–B.3). For any verified witness $\bar{u}_{k,\ell}^{(c)} \in \mathcal{U}$ and fixed $\sigma_j > 0$, define

$$\underline{M}_k^{(c)} = \min_{0 \le \ell \le H} \min_j \frac{r_j(\widehat{z}_{k,\ell}^{(c)}, \bar{u}_{k,\ell}^{(c)}) - \eta_{j,k,\ell}^{(c)}}{\sigma_j}. \tag{10}$$

The witnesses can be computed by LPs, independently of the unchanged selector.

Theorem 1 (Constructive sampled-horizon feasibility audit)**.** *For a candidate c, let the trajectory tube follow the bounded-input model contract in Appendix B.2, and use the corresponding row-error bounds for all modeled hard constraints. If $\underline{M}_k^{(c)} \ge 0$, then the joint hard CBF-QP is feasible at every audited sampled state in the executed prefix of length at most H. This applies to the selected candidate only if that candidate passes this separate test; it does not follow from $c \in \mathcal{I}_k$ alone.*

The proof and explicit obstacle, pairwise, boundary, and speed residual bounds are given in Appendix B. This is a checkable condition on a candidate, not an assertion that the candidate pool always contains one. No aggregate prediction-error constant is used.

The audit does not certify the full eight-step planner interval. Fallback and intersample limitations are stated in Remark 2 (Appendix B).

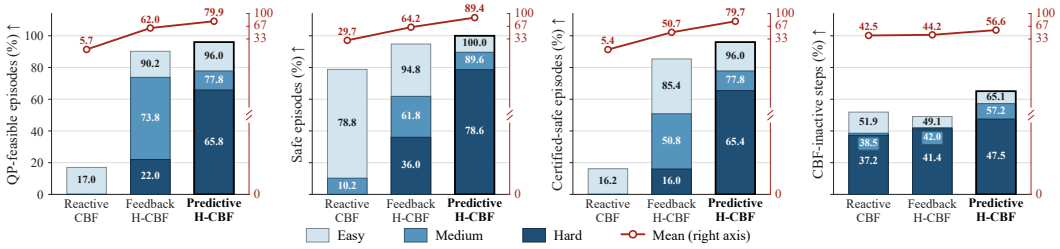

Figure 3: **Predictive H-CBF improves episode-level feasibility and safety.** All metrics are higher-is-better. Colored bars show separate rates for Easy, Medium, and Hard settings, not additive percentages. The red curve reports the arithmetic mean across settings. Predictive H-CBF achieves the best mean success profile and substantially improves the Hard setting.

4 Experiments

We evaluate Harness-CBF in three complementary studies that progressively isolate the contribution of CBF-aware predictive high-level decision making. Section 4.1 evaluates whether Harness-CBF feedback improves episode-level feasibility and safety on a 3-D multi-UAV coverage benchmark across three difficulty levels. Section 4.2 ablates the candidate selector to test whether predictive selection, rather than candidate diversity alone, drives the improvement. Section 4.3 evaluates whether the same predictive selector transfers across three swarm-control task families with distinct obstacle, density, and bottleneck structure. Gemini 2.5 Flash-Lite is used within the same LLM-based harness across all Harness-CBF variants, and the LLM outputs only high-level planning parameters or candidates rather than raw control inputs. Implementation details and parameter settings are given in Appendix D–H.

4.1 Harness-CBF feedback improves episode-level feasibility and safety

We first test whether CBF feedback improves episode-level feasibility and safety beyond reactive CBF-QP filtering. The failure mode follows the introduction: a low-level CBF-QP can correct the current nominal command, but it cannot prevent future high-level decisions from repeatedly driving the swarm toward low-margin regions. Under velocity, acceleration, obstacle, boundary, and pairwise constraints, such repeated decisions can make the multi-constraint CBF-QP highly active or hard-infeasible. We evaluate this failure mode on a 3-D multi-UAV coverage task with double-integrator dynamics. Gemini 2.5 Flash-Lite is used within the same LLM-based harness across all Harness-CBF variants, and the LLM outputs only high-level planning parameters or candidates rather than raw control inputs.

We compare three CBF-based variants that form an ablation chain from reactive CBF filtering to predictive high-level selection. *Reactive CBF* applies a low-level CBF-QP to the nominal coverage input, but does not feed any CBF feedback back to the high-level planner (Cortés et al., 2004; Schwager et al., 2009; Ames et al., 2017; Wang et al., 2017). *Feedback H-CBF* returns the CBF feedback terms to the LLM-based harness and directly revises the next high-level decision. *Predictive H-CBF* is our full method: it generates multiple high-level candidates and selects one using predicted coverage progress, CBF margin, QP feasibility, CBF intervention, and active constraints over a short horizon before execution. These three variants isolate three effects, low-level CBF filtering, high-level CBF feedback, and predictive candidate selection. We report four success-oriented metrics in Figure 3, which are higher-is-better. QP-feasible episodes are episodes without hard CBF-QP infeasibility. Safe episodes are episodes without realized safety violations. Certified-safe episodes satisfy both conditions. CBF-inactive steps measure the percentage of steps in which the low-level CBF-QP does not actively constrain or modify the nominal command. Table 2 in Appendix D reports the corresponding grouped statistics in the Hard setting. More details are provided in Appendix D.

Figure 3 shows that reactive CBF-QP filtering alone is not sufficient in the grouped stress benchmark. Reactive CBF loses the hard CBF-QP feasibility certificate in all Medium and Hard episodes. Returning CBF feedback to the harness improves feasibility, but direct feedback remains fragile in the Hard setting: Feedback H-CBF reaches only 22.00% QP-feasible episodes and 16.00% certified-safe

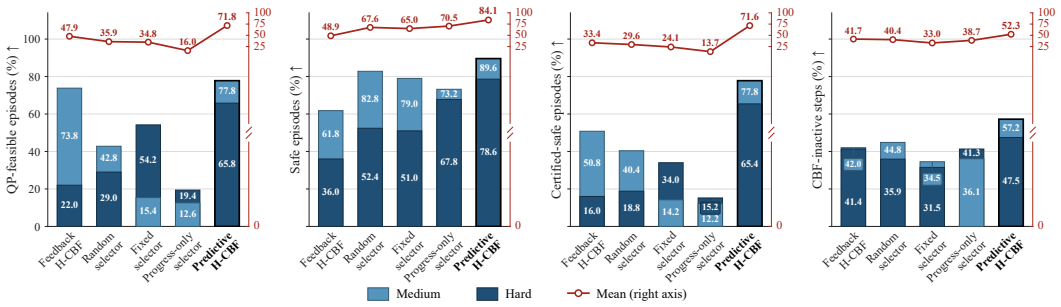

Figure 4: **CBF-aware predictive selection is the key mechanism in Harness-CBF.** All metrics are higher-is-better and follow Figure 3. Colored bars show separate Medium and Hard rates, not additive percentages. The red curve reports their arithmetic mean. Predictive H-CBF achieves the best mean success profile across all four metrics.

episodes. Predictive H-CBF gives the strongest Hard-setting success profile, increasing QP-feasible episodes to 65.80%, safe episodes to 78.60%, certified-safe episodes to 65.40%, and CBF-inactive steps to 47.47%.

The results are consistent with improved compatibility between nominal planning and the safety filter: the higher CBF-inactive rate of Predictive H-CBF (47.47% versus 41.36% under Feedback H-CBF) shows less recorded CBF activity for the selected high-level candidates. This metric alone cannot exclude slowdown as an explanation; task progress, speed, and stopping behavior must be considered jointly. Feedback H-CBF can react to a low CBF margin, but it does not compare alternative future decisions, whereas Predictive H-CBF evaluates candidate decisions with the same low-level CBF-QP before execution and penalizes candidates predicted to induce low margin, large intervention, repeated active constraints, or hard infeasibility. CBF feedback thus serves both as a corrective signal at the QP level and as a planning signal that shapes the next high-level decision. Table 2 reports descriptive means and between-group standard deviations; these standard deviations are not significance tests or confidence intervals for paired method differences.

4.2 Predictive Candidate Selection is the Key Mechanism

Experiment 4.1 shows that CBF feedback improves high-level planning beyond reactive CBF-QP filtering. We now isolate the selection mechanism inside Harness-CBF. The key question is whether the improvement comes simply from giving the LLM more high-level candidates, or from selecting among these candidates using predicted CBF-QP behavior. To answer this question, all candidate-selection variants use the same Gemini 2.5 Flash-Lite candidate-generation process, the same low-level CBF-QP, the same CBF feedback, and the same high-level candidate pool. They differ only in the selector used before execution. Feedback H-CBF is included as a single-decision feedback baseline, where the harness directly revises the next high-level decision without candidate selection.

We compare five variants. *Feedback H-CBF* maps the current CBF feedback to one high-level decision. *Random selector* samples a candidate from the shared candidate pool. *Fixed selector* always chooses a fixed candidate type. *Progress-only selector* chooses the candidate with the best predicted task progress, without considering predicted CBF margin, QP feasibility, intervention, or active constraints. *Predictive H-CBF* is the full method: it evaluates each candidate using predicted task progress together with these CBF feedback terms. We evaluate all variants on the Medium and Hard settings from Experiment 4.1, using the same success-oriented metrics as Figure 3.

Figure 4 shows that candidate diversity alone is not sufficient. Random and fixed selectors draw from the same candidate pool as Predictive H-CBF, but their QP-feasible and certified-safe rates remain limited. This indicates that having safer candidates available does not guarantee that the selected high-level decision will be compatible with the low-level CBF-QP. The selector must explicitly account for future CBF margin, QP feasibility, intervention, and active constraints.

Table 1: **Selector ablation with grouped statistics.** Values are mean and standard deviation across 10 independent groups with 50 seeds per group. All metrics are percentages and higher-is-better. Predictive H-CBF gives the strongest overall success profile in both Medium and Hard settings.

Selector	QP-feasible episodes (%) ↑	Safe episodes (%) ↑	Certified-safe episodes (%) ↑	CBF-inactive steps (%) ↑
Medium				
Feedback H-CBF	73.80 (2.57)	61.80 (6.70)	50.80 (5.51)	42.02 (1.12)
Random selector	42.80 (4.54)	82.80 (1.93)	40.40 (5.23)	44.85 (1.09)
Fixed selector	15.40 (4.53)	79.00 (4.74)	14.20 (4.57)	34.53 (1.01)
Progress-only selector	12.60 (5.50)	73.20 (4.24)	12.20 (5.12)	36.14 (1.70)
Predictive H-CBF	**77.80 (3.05)**	**89.60 (0.84)**	**77.80 (3.05)**	**57.21 (2.00)**
Hard				
Feedback H-CBF	22.00 (3.40)	36.00 (4.99)	16.00 (3.13)	41.36 (1.98)
Random selector	29.00 (5.68)	52.40 (5.32)	18.80 (4.13)	35.92 (1.11)
Fixed selector	54.20 (3.82)	51.00 (5.19)	34.00 (6.80)	31.50 (0.99)
Progress-only selector	19.40 (2.67)	67.80 (3.71)	15.20 (2.53)	41.31 (1.46)
Predictive H-CBF	**65.80 (3.33)**	**78.60 (1.65)**	**65.40 (3.41)**	**47.47 (1.58)**

The progress-only selector further separates task progress from CBF-QP feasibility. It can choose candidates that look attractive for coverage, and its safe-episode rate is not the worst. However, it obtains the lowest mean QP-feasible rate and the lowest mean certified-safe rate among the selector variants. Thus, task progress alone is not a reliable proxy for whether a future high-level decision can be realized by the low-level CBF-QP.

Table 1 confirms that Predictive H-CBF gives the strongest overall success profile. In the Hard setting, it increases QP-feasible episodes from 22.0% under Feedback H-CBF to 65.8%, and certified-safe episodes from 16.0% to 65.4%. These results show that the gain does not come merely from candidate diversity or conservative feedback. It comes from the predictive evaluator, which uses predicted low CBF margin, intervention, repeated activation, and hard-QP infeasibility to penalize high-level candidates before execution. These penalties do not certify every selected candidate.

4.3 Predictive H-CBF transfers across swarm-control tasks

Experiments 4.1 and 4.2 study the main 3-D coverage benchmark. They show that CBF feedback improves high-level planning and that CBF-aware predictive selection is the key mechanism. We now test whether the same mechanism transfers beyond one benchmark. Unlike task-specific controllers, we do not introduce new feedback rules, task-specific prompts, or task-specific feedback variables. Only the task family is changed.

We evaluate three swarm-control task families. *Static coverage* uses a fixed spatial density field with obstacle avoidance and pairwise separation constraints. *Dynamic-density coverage* uses moving density peaks, so the high-level coverage objective changes during the episode. *Narrow-passage navigation* forces the swarm through an obstacle-induced bottleneck, where obstacle and pairwise constraints can become simultaneously tight. These tasks are not Easy/Medium/Hard variants of one benchmark. They represent fixed-objective coverage, moving-objective coverage, and bottleneck-constrained motion. We compare Reactive CBF, Feedback H-CBF, and Predictive H-CBF in Figure 5. Table 8 reports the corresponding task-transfer statistics.

Figure 5 reports stronger transfer rates for Predictive H-CBF than reactive filtering or direct feedback. Reactive CBF loses QP feasibility in the harder task families, with only 0.40% QP-feasible episodes in Dynamic-density coverage and 0.00% in Narrow-passage navigation. Feedback H-CBF substantially improves feasibility by returning CBF margin feedback to the high-level planner, but it still has lower CBF-inactive-step ratios and worse failure-aware task completion than Predictive H-CBF. Predictive H-CBF gives the strongest transfer profile: it achieves the highest QP-feasible rate, the highest CBF-inactive-step ratio, and the highest task-completion rate in all three task families.

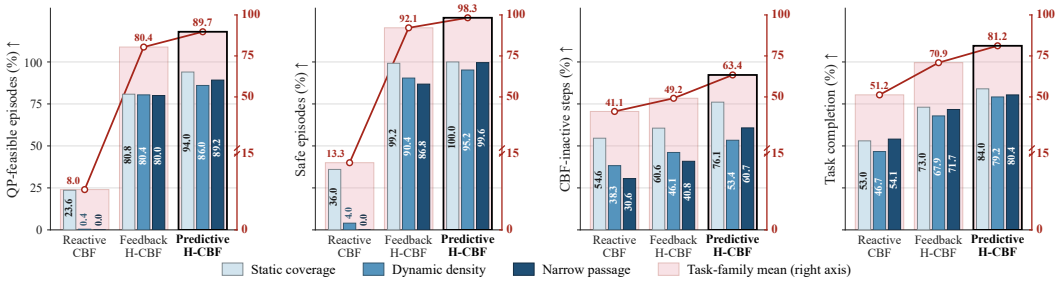

Figure 5: **Predictive H-CBF transfers across swarm-control task families.** All metrics are higher-is-better. Blue bars show separate task-family scores for Static coverage, Dynamic-density coverage, and Narrow-passage navigation. Predictive H-CBF achieves the best mean success profile across all four metrics.

The improvement is especially clear in the task families that differ from static coverage. The figure-aligned task scores are 67.9% versus 79.2% in Dynamic-density coverage. In Narrow passage, the corresponding inactive-step rates are 40.8% and 60.7%, and task scores are 71.7% and 80.4%. These results indicate that the predictive evaluator is not tuned to a single density field. It selects high-level candidates that remain compatible with the low-level CBF-QP across fixed-objective, moving-objective, and bottleneck-constrained swarm-control tasks.

Together, Experiments 4.1–4.3 show a consistent chain: CBF feedback improves episode-level feasibility and safety, CBF-aware predictive selection is the key mechanism, and the same mechanism transfers across static, dynamic, and bottleneck-constrained swarm-control tasks.

5 Discussion and Limitations

CBF-QP feasibility is usually viewed as a low-level optimization issue. This view is incomplete in swarm systems, where repeated CBF activation often results from high-level decisions that are difficult for the low-level CBF-QP to realize. Our results suggest that maintaining CBF-QP feasibility also requires regulating the future decisions generated by the LLM-based planner. Thus, CBF feedback should not only be used to correct the current control step; it can also be used to improve future high-level decisions before they reach the low-level CBF-QP.

The LLM-based harness implements this high-level revision. It receives the CBF margin and related CBF feedback, combines them with task context and memory, and generates multiple high-level candidates. The predictive evaluator then compares these candidates over a short horizon, rather than relying on a single one-step feedback reaction. This comparison allows the selected decision to preserve task performance while reducing repeated CBF activation and QP infeasibility.

This discussion also points to several practical limitations. First, Predictive H-CBF uses a structured high-level candidate representation, where the LLM proposes a finite set of planning parameters rather than unrestricted actions. This design makes the feedback loop easier to evaluate and keeps the executed control input under the low-level CBF-QP, but it also restricts the range of high-level behaviors explored in this work. Second, the current implementation relies on an API-based LLM planner. While this setting is useful for evaluating the proposed CBF-feedback mechanism, real robot deployment will require more predictable inference latency, tighter integration with onboard computation, and validation with locally deployed models. Third, our experiments focus on 3-D swarm-control tasks, including static coverage, narrow-passage navigation, and dynamic-density coverage. The feedback mechanism itself is not tied to coverage control, since it only requires low-level CBF feedback and a high-level decision representation. However, broader validation on other multi-constraint multi-agent tasks remains future work. The current experiments also do not establish the marginal value of the LLM over non-LLM generators, large-swarm or decentralized scalability, or the empirical coverage of the new audit certificate. We claim no unconditional recursive feasibility, safe fallback, or closed-loop convergence. Overall, the empirical conclusion is limited to the evaluated configurations: CBF feedback can be used not only as a low-level correction signal, but also as a high-level planning signal for reducing future CBF-QP burden.

AI USE STATEMENT

Generative AI assisted editing, derivations and preparation of diagnostic code.

REFERENCES

Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. In *Conference on Robot Learning (CoRL)*, 2022.

Aaron D. Ames, Xiangru Xu, Jessy W. Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2017.

Aaron D. Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *Proceedings of the 18th European Control Conference (ECC)*, pp. 3420–3431. IEEE, 2019.

Urs Borrmann, Li Wang, Aaron D. Ames, and Magnus Egerstedt. Control barrier certificates for safe swarm behavior. *IFAC-PapersOnLine*, 48(27):68–73, 2015.

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*, 5:411–444, 2022.

Soon-Jo Chung, Aditya Avinash Paranjape, Philip Dames, Shaojie Shen, and Vijay Kumar. A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4):837–855, 2018.

Jorge Cortés, Sonia Martínez, Timur Karatas, and Francesco Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

Yaosheng Deng, Masaki Ogura, Yang Bai, Yujie Wang, and Mir Feroskhan. Safety critical control for nonlinear systems with complex input constraints. *IEEE Transactions on Automatic Control*, 2025.

Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. PaLM-E: An embodied multimodal language model. In *International Conference on Machine Learning (ICML)*, 2023.

Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE Control Systems Letters*, 1(2):310–315, 2017.

Kai-Chieh Hsu, Haimin Hu, and Jaime F. Fisac. Safety filter design for neural network systems via convex optimization. In *2023 62nd IEEE Conference on Decision and Control (CDC)*, pp. 1814–1821. IEEE, 2023.

Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning (ICML)*, pp. 9118–9147, 2022a.

Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning (CoRL)*, 2022b.

Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9493–9500. IEEE, 2023.

Zhipeng Liu, Xuefeng Bai, Kehai Chen, Xinyang Chen, Xiucheng Li, Yang Xiang, Jin Liu, Hong-Dong Li, Yaowei Wang, Liqiang Nie, and Min Zhang. A survey on the feedback mechanism of LLM-based AI agents. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 10582–10592, 2025.

David Q. Mayne, James B. Rawlings, Christopher V. Rao, and Pierre O. M. Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789–814, 2000.

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

Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. Decentralized, adaptive coverage control for networked robots. *The International Journal of Robotics Research*, 28(3):357–375, 2009.

Mac Schwager, Brian J. Julian, Michael Angermann, and Daniela Rus. Eyes in the sky: Decentralized control for the deployment of robotic camera networks. *Proceedings of the IEEE*, 99(9):1541–1561, 2011.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. ProgPrompt: Generating situated robot task plans using large language models. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11523–11530. IEEE, 2023.

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

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research (TMLR)*, 2024.

Li Wang, Aaron D. Ames, and Magnus Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Peter Wieland and Frank Allgöwer. Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12):462–467, 2007.

Wei Xiao and Calin Belta. Control barrier functions for systems with high relative degree. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 474–479, 2019. doi: 10.1109/CDC40024.2019.9029455.

Xiangru Xu, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*, 48(27):54–61, 2015.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023b.

Jun Zeng, Bike Zhang, and Koushil Sreenath. Safety-critical model predictive control with discrete-time control barrier function. In *2021 American Control Conference (ACC)*, pp. 3882–3889. IEEE, 2021.

A Additional Method Details

This appendix expands the method section in 3. We first state the system model and the nominal controller used in the experiments. We then describe the low-level CBF-QP, because this QP remains the formal safety layer throughout the paper. After that, we define the feedback returned by the CBF-QP and show how the harness uses it to shape high-level targets. We then give the predictive evaluation rule used to select among high-level candidates. Finally, we state the implementation scope relevant to the analysis.

A.1 System Model and Nominal Control

The main text states that the harness does not output raw control inputs. It only changes high-level planning parameters. This subsection specifies how these high-level parameters are converted into a nominal input before the CBF-QP filters it.

We consider N agents in a compact workspace $\mathcal{X}$. In the main 3-D coverage task, each agent is modeled as a double integrator,

$$\dot{p}_i = v_i, \qquad \dot{v}_i = u_i, \tag{11}$$

where $p_i \in \mathbb{R}^3$, $v_i \in \mathbb{R}^3$, and $u_i \in \mathbb{R}^3$. The control input is bounded by

$$u_i \in \mathcal{U}_i := \left\{ u_i \in \mathbb{R}^3 : |u_{i,\ell}| \le u_{\max},\ \ell = 1,2,3 \right\}. \tag{12}$$

The full admissible input set is

$$\mathcal{U} = \mathcal{U}_1 \times \cdots \times \mathcal{U}_N. \tag{13}$$

The high-level planner provides a target $c_i \in \mathbb{R}^3$ for each agent. In the coverage task, c_i is a density-weighted coverage centroid after harness correction. The nominal velocity is

$$v_{i,\mathrm{des}} = \mathrm{sat}_{v_{\max}\gamma}\left(k_p(c_i - p_i)\right), \tag{14}$$

where $\gamma \in (0,1]$ is the speed scale selected by the harness. The nominal acceleration is

$$u_{i,\mathrm{nom}} = \mathrm{sat}_{\mathcal{U}_i}\left(k_v(v_{i,\mathrm{des}} - v_i)\right). \tag{15}$$

This nominal control is task-driven. It is not guaranteed to be safe. The CBF-QP described below filters it before execution.

A.2 Safety Sets and Low-Level CBF-QP

The nominal input defined above is not executed directly. It is filtered by the low-level CBF-QP. The low-level CBF-QP enforces obstacle avoidance, inter-agent separation, boundary safety, speed constraints, and input constraints. We write the main safety constraints below.

For obstacle o, let $p_o \in \mathbb{R}^3$ be its center and $R_o > 0$ be its inflated safe radius. The obstacle safety function for agent i is

$$h_{i,o}^{\mathrm{obs}}(p_i) = \|p_i - p_o\|^2 - R_o^2. \tag{16}$$

The corresponding safe set is

$$\mathcal{C}_{i,o}^{\mathrm{obs}} = \left\{ p_i : h_{i,o}^{\mathrm{obs}}(p_i) \ge 0 \right\}. \tag{17}$$

For inter-agent collision avoidance, the pairwise safety function is

$$h_{ij}^{\mathrm{pair}}(p_i, p_j) = \|p_i - p_j\|^2 - d_{\min}^2, \qquad i \ne j, \tag{18}$$

where $d_{\min}$ is the minimum allowed distance.

For boundary safety, let the workspace box be defined by

$$\underline{p}_\ell \le p_{i,\ell} \le \bar{p}_\ell, \qquad \ell = 1,2,3. \tag{19}$$

The lower and upper boundary functions are

$$h_{i,\ell}^{\mathrm{low}} = p_{i,\ell} - \underline{p}_\ell, \qquad h_{i,\ell}^{\mathrm{up}} = \bar{p}_\ell - p_{i,\ell}. \tag{20}$$

The speed constraint is

$$h_i^{\mathrm{vel}}(v_i) = v_{\max}^2 - \|v_i\|^2. \tag{21}$$

This constraint has relative degree one with respect to u_i.

The position-based constraints in (16), (18), and (20) have relative degree two for the double-integrator model. We therefore use the standard high-order CBF construction. For a relative-degree-two safety function $h(x)$, define

$$\psi_0(x) = h(x), \qquad \psi_1(x) = \dot{h}(x) + \alpha_1(\psi_0(x)), \tag{22}$$

where α_1 is an extended class-$\mathcal{K}$ function. The HOCBF condition is

$$\dot{\psi}_1(x, u) + \alpha_2(\psi_1(x)) \geq 0, \tag{23}$$

where α_2 is another extended class-$\mathcal{K}$ function. All HOCBF inequalities are affine in u for the double-integrator system.

At each step, the low-level CBF-QP solves

$$\begin{aligned} (u_{\mathrm{safe}}, \delta^\star) \in \arg\min_{u,\delta} \quad & \frac{1}{2}\|u - u_{\mathrm{nom}}\|^2 + c_\delta \|\delta\|_2^2 \\ \text{s.t.} \quad & A(x)u \leq b(x) + \delta, \\ & u \in \mathcal{U}, \\ & \delta \geq 0. \end{aligned} \tag{24}$$

Here, $A(x)u \leq b(x)$ collects all CBF and high-order CBF constraints. The displayed program is a relaxed-QP formulation. Its input retains the symbol u_{safe} for consistency, but the symbol is not a certificate when hard rows are violated. A finite quadratic slack penalty does not force zero slack even at a hard-feasible state (Remark 1). Hard feasibility and actual hard-row satisfaction must therefore be checked independently; a hard-first implementation is not equivalent to this formulation. A step is called hard infeasible if no input $u \in \mathcal{U}$ satisfies the CBF constraints without relaxation.

Assumption 1. *The dynamics are locally Lipschitz on the operating domain. Relative-degree-one safety functions are continuously differentiable; position-based relative-degree-two functions and the comparison functions have the derivatives required by the displayed HOCBF construction. The admissible input set $\mathcal{U}$ is compact and convex.*

Safety scope. Standard CBF/HOCBF invariance requires that the *applied* input satisfy all hard barrier inequalities, together with the relevant initial conditions. Feasibility of the relaxed optimization problem in Eq. (24) is insufficient. The precise conditional result and the sampled-data qualification are given in Appendix B.6.

A.3 Feedback from the CBF-QP

The low-level CBF-QP returns the filtered input and several feedback values. These values measure how difficult it was for the QP to realize the nominal plan under the safety and input constraints. The minimum safety margin is

$$m_k = \min_j h_j(x_k), \tag{25}$$

where the index j runs over all monitored safety functions. In implementation, we also keep separate margins for obstacle avoidance, pairwise separation, speed, and boundary safety. The raw minimum is a scale-dependent diagnostic, not a physical distance or a hard-QP feasibility reserve. Different barrier families need not have the same units. We retain the reported diagnostic and score configuration; positive row normalization is introduced only in the separate feasibility analysis.

The intervention norm is

$$r_k = \|u_{\mathrm{safe},k} - u_{\mathrm{nom},k}\|. \tag{26}$$

This value is small when the nominal plan is already compatible with safety. It is large when the CBF-QP must strongly modify the nominal input.

The active constraint ratio is

$$a_k = \frac{1}{M}\sum_{j=1}^{M} \mathbf{1}\left\{|A_j(x_k)u_{\mathrm{safe},k} - b_j(x_k)| \leq \epsilon_{\mathrm{act}}\right\}, \tag{27}$$

where M is the number of CBF constraints and $\epsilon_{\text{act}} > 0$ is a small tolerance. This is a near-tightness diagnostic for the hard rows at the returned input. In a relaxed solve it need not equal the active-set fraction of the relaxed constraints $Au \leq b + \delta$. The diagnostic definition is retained; the two interpretations must not be conflated.

The slack value is

$$s_k = \|\delta_k\|, \tag{28}$$

where δ_k is the relaxation variable in (24). The hard infeasibility indicator is

$$q_k = \begin{cases} 1, & \text{if the hard CBF constraints are infeasible under } u \in \mathcal{U}, \\ 0, & \text{otherwise.} \end{cases} \tag{29}$$

We collect these values in the feedback vector

$$\phi_k = [m_k,\, r_k,\, a_k,\, s_k,\, q_k]\,. \tag{30}$$

We convert it into a scalar CBF-QP burden score P_k, which is a planning signal used by the harness.

$$P_k = \lambda P_{k-1} + (1-\lambda)\ell(\phi_k), \tag{31}$$

where

$$\ell(\phi_k) = c_m[\bar{m} - m_k]_+ + c_r r_k + c_a a_k + c_s s_k + c_q q_k. \tag{32}$$

Here, $[z]_+ = \max\{z, 0\}$. The constants c_m, c_r, c_a, c_s, c_q are nonnegative weights. The value $\bar{m}$ is a desired safety margin.

The CBF-QP burden score P_k is used only by the high-level harness. A large value means that recent high-level plans have been difficult for the CBF-QP to realize. A small value means that the high-level planner and the low-level CBF-QP are compatible.

A.4 How the Harness Shapes the LLM's Output

The harness does not output raw control inputs. It outputs high-level planning parameters,

$$\theta_k = \left[\gamma_k,\, \eta_k^{\text{obs}},\, \eta_k^{\text{pair}},\, \eta_k^{\text{alt}}\right]. \tag{33}$$

These parameters shape the target used by the nominal controller.

Let c_i be the original coverage target of agent i. The harness-corrected target is

$$\tilde{c}_i = \Pi_{\mathcal{X}}\left(c_i + \eta_k^{\text{obs}} d_i^{\text{obs}} + \eta_k^{\text{pair}} d_i^{\text{pair}} + \eta_k^{\text{alt}} d_i^{\text{alt}}\right), \tag{34}$$

where $\Pi_{\mathcal{X}}$ is the projection onto the workspace. The vector d_i^{obs} moves the target away from obstacles. The vector d_i^{pair} spreads nearby agents. The vector d_i^{alt} assigns high or low altitude lanes when the workspace is crowded. These vectors are computed from the current geometry and the harness only chooses the weights in θ_k.

The speed scale γ_k changes the desired velocity in (14). A small γ_k slows down the nominal planner. A large γ_k allows faster task progress. Thus, the harness changes the high-level intent, but it does not bypass the CBF-QP.

A.5 Predictive Candidate Evaluation

The harness proposes N_c candidate planning parameters,

$$\Theta_k = \left\{\theta_k^{(1)}, \ldots, \theta_k^{(N_c)}\right\}. \tag{35}$$

Each candidate represents a different safety-performance trade-off. The final candidate is chosen by a predictive evaluator.

For each candidate $\theta_k^{(i)}$, the evaluator performs a short rollout of length H. During the rollout, the nominal input is computed from the shaped target, and the same CBF-QP in (24) is applied. This produces predicted states, inputs, and CBF-QP feedback terms.

The task cost is denoted by

$$J^{(i)}_{\text{task}}. \tag{36}$$

In the coverage task, this is the predicted coverage error. The safety-margin penalty is

$$J^{(i)}_{\text{margin}} = \sum_{\tau=1}^{H} [\bar{m} - m^{(i)}_{k+\tau}]^2_+. \tag{37}$$

The predicted infeasibility penalty is

$$J^{(i)}_{\text{infeas}} = \sum_{\tau=1}^{H} q^{(i)}_{k+\tau}. \tag{38}$$

The intervention penalty is

$$J^{(i)}_{\text{int}} = \sum_{\tau=1}^{H} \left\| u^{(i)}_{\text{safe},k+\tau} - u^{(i)}_{\text{nom},k+\tau} \right\|^2. \tag{39}$$

The active-constraint penalty is

$$J^{(i)}_{\text{active}} = \sum_{\tau=1}^{H} a^{(i)}_{k+\tau}. \tag{40}$$

We combine these four predicted feedback penalties into the predicted CBF-QP burden score

$$\widehat{P}^{(i)}_{\text{cbf}} = w_1 J^{(i)}_{\text{margin}} + w_2 J^{(i)}_{\text{infeas}} + w_3 J^{(i)}_{\text{int}} + w_4 J^{(i)}_{\text{active}}. \tag{41}$$

Here, $J^{(i)}_{\text{margin}}$, $J^{(i)}_{\text{infeas}}$, $J^{(i)}_{\text{int}}$, and $J^{(i)}_{\text{active}}$ are component penalties. The scalar $\widehat{P}^{(i)}_{\text{cbf}}$ is the predicted cost that measures how difficult candidate i is expected to be for the low-level CBF-QP.

The evaluator first forms the score-eligible candidate set

$$\mathcal{I}_k = \left\{ i \in \{1, \dots, N_c\} : \widehat{P}^{(i)}_{\text{cbf}} \le \tau_P \right\}. \tag{42}$$

The threshold τ_P is a search parameter; membership in this set is not a hard-QP certificate.

If $\mathcal{I}_k \neq \emptyset$, the evaluator selects the candidate with the smallest task–feasibility score,

$$i^\star = \arg\min_{i \in \mathcal{I}_k} \left(J^{(i)}_{\text{task}} + \lambda_P \widehat{P}^{(i)}_{\text{cbf}} \right). \tag{43}$$

If $\mathcal{I}_k = \emptyset$, no candidate satisfies the configured CBF-burden threshold. In this case, the evaluator selects the fallback candidate with the smallest predicted CBF-QP burden score,

$$i^\star = \arg\min_{i \in \{1,\dots,N_c\}} \widehat{P}^{(i)}_{\text{cbf}}. \tag{44}$$

The evaluator therefore balances two effects. It can select a task-progressive candidate when the predicted CBF-QP burden score remains small. It rejects or penalizes a candidate when the rollout predicts low margin, large intervention, repeated active constraints, or hard infeasibility. The hard-feasibility audit in Appendix B is separate from this burden-based selection rule.

A.6 Implementation Notes

The LLM planner is not allowed to output raw control inputs. It only outputs high-level planning parameters. The final control input is always computed by the nominal controller and then filtered by the CBF-QP.

The candidate evaluator is implemented separately from the API planner and described as deterministic. It uses the same low-level CBF-QP as the real system during rollout. This design avoids using the LLM as the final safety authority.

The reported experiments use the configured finite rollout, without a demonstrated latency certificate. An audit must start from the state at which a decision is actually applied; API latency or a changed state invalidates an exact-clone argument unless explicitly included in the transition or uncertainty bounds.

B Feasibility Analysis Without an Aggregate Prediction-Error Assumption

This appendix analyzes the four-parameter planner and its soft-score selector. It does not replace that selector with a hard-margin gate, or change the rollout horizon. The quantity $\widehat{P}_{\mathrm{cbf}}$ is a search heuristic. We give a separate, computable sufficient condition for sampled-state hard-QP feasibility. The condition can be audited for any candidate, including the candidate selected by the unchanged algorithm. No uniform error bound on the aggregate score P_{cbf} is assumed.

B.1 Hard feasibility is distinct from a soft-score guard

For the joint swarm state $z = \mathrm{col}(p_1, v_1, \ldots, p_N, v_N)$, write every hard CBF/HOCBF row as

$$r_j(z,u) = b_j(z) - A_j(z)u, \qquad j \in \mathcal{J}, \quad \mathcal{K}(z) = \{u \in \mathcal{U} : r_j(z,u) \geq 0 \; \forall j\}. \tag{45}$$

Here $\mathcal{U}$ is the nonempty compact input box already specified in Eq. (12), and $\mathcal{J}$ is a finite, nonempty set of all hard rows being certified. Pairwise rows depend on the joint input, including both u_i and u_j; independent per-agent feasibility tests do not replace this joint test. Let $\sigma_j > 0$ be fixed row scales and define

$$\mu(z) = \max_{u \in \mathcal{U}} \min_{j \in \mathcal{J}} \frac{r_j(z,u)}{\sigma_j}. \tag{46}$$

The scales affect reserve magnitudes but not the sign test below. They are used only in this analysis and do not change the implemented burden score, nominal controller, or executed QP.

Lemma 1 (Joint hard-QP feasibility)**.** *If each $r_j(z,\cdot)$ is continuous, then $\mathcal{K}(z) \neq \emptyset$ if and only if $\mu(z) \geq 0$. For affine rows and a box input set, $\mu(z)$ is the optimum of the LP*

$$\max_{u \in \mathcal{U}, \gamma \in \mathbb{R}} \; \gamma \quad \textit{subject to} \quad A_j(z)u + \sigma_j \gamma \leq b_j(z), \quad j \in \mathcal{J}. \tag{47}$$

Proof. Continuity and compactness imply that the maximum in Eq. (46) is attained. A feasible hard-QP input has all row residuals nonnegative and therefore gives a nonnegative minimum. Conversely, a maximizer with a nonnegative minimum satisfies every hard row, because all σ_j are positive. Introducing the common lower bound γ gives Eq. (47). □

This is an optimization characterization, not a new CBF invariance theorem. The nominal input occurs in the QP objective, not in $\mathcal{K}(z)$. At a fixed state, changing u_{nom} cannot repair an empty $\mathcal{K}(z)$. High-level decisions can affect *future* feasibility through the filtered trajectory they induce. Likewise, positive geometric barrier values $h_j(z)$ do not imply hard-QP feasibility under bounded inputs.

The burden-based guard $\widehat{P}_{\mathrm{cbf}} \leq \tau_P$ does not imply $\mu \geq 0$. For example, a predicted hard-infeasible step contributes w_2 to the score and need not be rejected when $\tau_P \geq w_2$. Conversely, a hard-feasible rollout can have a large score due to intervention or small geometric margins. Thus the score-eligible set $\mathcal{I}_k$ in the method must not be called a hard-certified set.

B.2 State tubes obtained from the bounded-input model

Fix a candidate and its predicted states and inputs $(\widehat{p}_{i,\ell}, \widehat{v}_{i,\ell}, \widehat{u}_{i,\ell})$, initialized at a planning instant. Suppress the candidate index in this appendix when unambiguous. Under exact zero-order-hold integration of the double integrator,

$$\begin{aligned} p_{i,\ell+1} &= p_{i,\ell} + T_s v_{i,\ell} + \tfrac{1}{2} T_s^2 u_{i,\ell} + w^p_{i,\ell}, \\ v_{i,\ell+1} &= v_{i,\ell} + T_s u_{i,\ell} + w^v_{i,\ell}. \end{aligned} \tag{48}$$

The prediction uses the same nominal discretization without w^p, w^v. The bounds $\|w^p_{i,\ell}\| \leq \bar{w}^p_{i,\ell}$ and $\|w^v_{i,\ell}\| \leq \bar{w}^v_{i,\ell}$ are explicit model-error budgets; they are zero for this exact disturbance-free model. Nonzero budgets require independent justification, not an empirical maximum relabeled as a universal bound. Initial state-estimation errors are bounded by $e^p_{i,0}, e^v_{i,0}$.

Because the actual input lies in the input box, an input-discrepancy bound that requires no regularity of the QP solution map is

$$D_{i,\ell} := \max_{u_i \in \mathcal{U}_i} \|u_i - \widehat{u}_{i,\ell}\|_2 = \left[\sum_{d=1}^{3} (u_{\max} + |\widehat{u}_{i,\ell,d}|)^2\right]^{1/2}. \tag{49}$$

The equality is for the componentwise symmetric box in the manuscript. A smaller bound may be substituted only if separately justified. In particular, the same QP routine alone does not justify setting $D_{i,\ell} = 0$. Define

$$\begin{aligned} e^v_{i,\ell+1} &= e^v_{i,\ell} + T_s D_{i,\ell} + \bar{w}^v_{i,\ell}, \\ e^p_{i,\ell+1} &= e^p_{i,\ell} + T_s e^v_{i,\ell} + \tfrac{1}{2} T_s^2 D_{i,\ell} + \bar{w}^p_{i,\ell}. \end{aligned} \tag{50}$$

Lemma 2 (Input-bounded trajectory tube)**.** *Under Eq.* (48) *and its stated initial-error and model-error bounds, the actual and predicted trajectories satisfy* $\|p_{i,\ell} - \widehat{p}_{i,\ell}\| \le e^p_{i,\ell}$ *and* $\|v_{i,\ell} - \widehat{v}_{i,\ell}\| \le e^v_{i,\ell}$ *at each evaluated step, provided actual and predicted trajectories are defined and actual inputs remain in* $\mathcal{U}$.

Proof. The inequalities hold at $\ell = 0$. Subtract the predicted update from Eq. (48). Apply the triangle inequality, the induction hypothesis, the disturbance budgets, and $\|u_{i,\ell} - \widehat{u}_{i,\ell}\| \le D_{i,\ell}$. The resulting two bounds are exactly Eq. (50). Induction completes the proof. No differentiability or Lipschitz property of the QP optimizer is used. □

Discretization contract. The certificate must be matched to the actual integration scheme; Eq. (48) specifies the zero-order-hold case. If both prediction and execution instead use forward Euler, omit the $\frac{1}{2}T_s^2 u$ term and the corresponding term in the position-error recursion. If their integrators differ, bound that difference in w^p, w^v. One cannot claim a zero-error simulator correspondence from shared CBF code alone. The input-box tube can be conservative; its certified-candidate availability must be measured rather than assumed.

B.3 Explicit error bounds for the existing CBF rows

For the position-based functions in Appendix A.2, take linear HOCBF functions $\alpha_1(s) = \kappa_1 s$ and $\alpha_2(s) = \kappa_2 s$, with $\kappa_1, \kappa_2 > 0$. Set $\lambda = \kappa_1 + \kappa_2$ and $\beta = \kappa_1\kappa_2$. These give $\ddot{h} + \lambda\dot{h} + \beta h \ge 0$. The reported obstacle/pairwise coefficients $k_1 = 3.7, k_0 = 3.0$ factor as $\kappa_1 = 1.2, \kappa_2 = 2.5$ (or in the reverse order); the initial HOCBF condition must use the factorization actually chosen. Rows with different gains use their own λ_j, β_j; a first-order boundary implementation must not be analyzed as a second-order row.

For a static spherical obstacle use $p = p_i - p_o$, $v = v_i$, and $a = u_i$; for an agent pair use $p = p_i - p_j$, $v = v_i - v_j$, and $a = u_i - u_j$. For $h = \|p\|^2 - d^2$, the residual is

$$r(p, v, a) = 2\|v\|^2 + 2p^\top a + 2\lambda p^\top v + \beta(\|p\|^2 - d^2). \tag{51}$$

Let $\|p - \widehat{p}\| \le E_p$, $\|v - \widehat{v}\| \le E_v$, and $\|a\| \le A_*$ uniformly over admissible joint inputs. For obstacle rows, $E_p = e^p_i, E_v = e^v_i, A_* = \sqrt{3}u_{\max}$; for pairwise rows, add the two agents' position and velocity error radii and use $A_* = 2\sqrt{3}u_{\max}$. A uniform row-error bound is

$$\begin{aligned} \eta^{\mathrm{sph}} = {} & 4\|\widehat{v}\|E_v + 2E_v^2 + 2A_* E_p \\ & + 2\lambda(\|\widehat{p}\|E_v + \|\widehat{v}\|E_p + E_p E_v) + \beta(2\|\widehat{p}\|E_p + E_p^2). \end{aligned} \tag{52}$$

For a lower or upper linear boundary function and its second-order row,

$$\eta^{\mathrm{bdry}} = \lambda_j e^v_i + \beta_j e^p_i. \tag{53}$$

For the speed barrier $h = v_{\max}^2 - \|v_i\|^2$ with gain $\kappa_v > 0$,

$$\begin{aligned} r^{\mathrm{vel}}(v_i, u_i) &= -2v_i^\top u_i + \kappa_v(v_{\max}^2 - \|v_i\|^2), \\ \eta^{\mathrm{vel}} &= 2\sqrt{3}u_{\max} e^v_i + \kappa_v(2\|\widehat{v}_i\|e^v_i + (e^v_i)^2). \end{aligned} \tag{54}$$

All quantities are computed from the prediction, input limits, row gains, and the tube radii. They require no Lipschitz constant for an active-set switching QP or for the discontinuous hard-infeasibility indicator.

Lemma 3 (Row-wise residual enclosure)**.** *For the rows just defined, at every state in the trajectory tube and for every $u \in \mathcal{U}$,*

$$r_j(z,u) \geq r_j(\widehat{z},u) - \eta_j. \tag{55}$$

Proof. For Eq. (51), use $|\|v\|^2 - \|\widehat{v}\|^2| \leq 2\|\widehat{v}\|E_v + E_v^2$, $|(p-\widehat{p})^\top a| \leq A_* E_p$, and $|p^\top v - \widehat{p}^\top \widehat{v}| \leq \|\widehat{p}\|E_v + \|\widehat{v}\|E_p + E_p E_v$. The same squared-norm inequality for p gives Eq. (52). The boundary row is affine in p, v, which yields Eq. (53). Applying the inner-product and squared-norm bounds to r^{vel} gives Eq. (54). These are bounds on the absolute residual error, so each implies the one-sided enclosure. □

Modeled constraints only. The enclosure uses a fixed set of modeled static obstacles, pairwise rows, boundary rows, and speed rows across the audit horizon. Moving or newly observed obstacles, state-dependent row pruning, changed input limits, or additional constraints require corresponding enclosures before they can be included in the claim. A changed task-density field is not, by itself, a moving obstacle. These statements delimit the certificate rather than asserting that the simulator already implements a disturbance model.

B.4 Candidate certificates and proof of Theorem 1

For candidate c and predicted step ℓ, compute a witness input $\bar{u}_{k,\ell}^{(c)} \in \mathcal{U}$; it can be obtained by the LP

$$\max_{u \in \mathcal{U}, \gamma \in \mathbb{R}} \ \gamma \quad \text{s.t.} \quad r_j(\widehat{z}_{k,\ell}^{(c)}, u) - \eta_{j,k,\ell}^{(c)} \geq \sigma_j \gamma, \quad j \in \mathcal{J}. \tag{56}$$

The LP has an unrestricted real γ, so finite residuals and the nonempty input box ensure that this diagnostic LP is itself feasible and has a finite optimum. It does not use a relaxation variable to declare the hard constraints feasible. Independently evaluate the witness residuals and let

$$\underline{\mu}_{k,\ell}^{(c)} = \min_j \frac{r_j(\widehat{z}_{k,\ell}^{(c)}, \bar{u}_{k,\ell}^{(c)}) - \eta_{j,k,\ell}^{(c)}}{\sigma_j}, \qquad \underline{M}_k^{(c)} = \min_{0 \leq \ell \leq H} \underline{\mu}_{k,\ell}^{(c)}. \tag{57}$$

A nonoptimal witness remains valid: a verified primal value is a lower bound, whereas an unverified solver-reported objective is not a certificate. A separate diagnostic set is

$$\mathcal{C}_k^{\mathrm{cert}} = \{c : \underline{M}_k^{(c)} \geq 0\}. \tag{58}$$

This is *not* the score-eligible set $\mathcal{I}_k$ used by the burden-based selector. Certification need not be used to change a selected action. The reported experimental tables do not measure $\mathcal{C}_k^{\mathrm{cert}}$ or establish its nonemptiness.

Proof of Theorem 1. Lemma 2 encloses the actual sampled state at each step covered by its model contract. By Lemma 3, for every j and every such ℓ,

$$r_j(z_{k+\ell}^{(c)}, \bar{u}_{k,\ell}^{(c)}) \geq r_j(\widehat{z}_{k,\ell}^{(c)}, \bar{u}_{k,\ell}^{(c)}) - \eta_{j,k,\ell}^{(c)} \geq \sigma_j \underline{\mu}_{k,\ell}^{(c)}.$$

Thus $\underline{M}_k^{(c)} \geq 0$ provides an admissible joint input satisfying all hard rows at each audited state. Applying Lemma 1 proves $\mu(z_{k+\ell}^{(c)}) \geq \underline{\mu}_{k,\ell}^{(c)} \geq 0$. The claim holds for the selected candidate only when that candidate has passed this separate test. It does not follow merely from membership in $\mathcal{I}_k$. □

Numerical verification. Equations (57)–(58) are exact-arithmetic statements. With floating-point arithmetic, project (or otherwise verify) the witness in the input box, recompute all rows, and subtract a justified one-sided evaluation-error bound for each row. Interval/outward-rounded evaluation is one way to obtain such bounds. An arbitrary feasibility tolerance is not a proved error bound. Without verified residual evaluation, report a numerical diagnostic rather than a rigorous machine-precision certificate. Separate negative certified upper bounds (evidence of infeasibility), nonnegative verified primal lower bounds (evidence of feasibility), and unresolved solver outcomes.

Exact model correspondence as a special case. Prediction and execution coincide by induction if initialized at the same complete state and advanced by the same deterministic closed-loop transition, with the same candidate, controller memory, integrator, environment update, and exogenous inputs. In that case the state tube is zero and the nominal reserve test is exact at the sampled states. Sharing a simulator library, QP formulation, or random seed is not sufficient by itself. No exact correspondence is inferred for physical deployment or unobserved future randomness.

Finite horizon, not recursive feasibility. The reported configuration uses $H = 3$ and updates the LLM every eight control steps. A three-step prediction certifies at most its three-step executed prefix (and an explicitly tested terminal sampled state), not the remaining five steps. These settings are kept separate in the analysis. Membership in $\mathcal{C}_k^{\mathrm{cert}}$ neither proves $\mathcal{C}_{k+1}^{\mathrm{cert}} \neq \emptyset$ nor guarantees a safe fallback. Adding a terminal invariant set or a certified backup would be a different algorithmic extension and is not claimed here.

B.5 Selection and fallback scope

If a direct-feedback candidate is in the same score-eligible pool, the predictive selector has no larger *predicted* implemented score on that pool. This is simply a property of the argmin, not a new theorem, a guarantee of true task improvement, or a general proof of reduced conservatism. The latter is an empirical safety–performance question. When $\mathcal{I}_k = \emptyset$, the minimum-burden rule still returns one candidate from the nonempty pool, but no horizon certificate follows. If the current hard QP remains feasible, the low-level controller continues to operate under its ordinary conditional safety requirements. If it is infeasible, changing only the nominal objective cannot restore the current hard feasible set. This work claims reductions in observed infeasibility, not universal prevention, convergence of the full closed loop, or safety of arbitrary initial states.

B.6 Low-level safety and the relaxed-QP distinction

Standard CBF/HOCBF safety results concern the *applied input* and the continuous-time barrier inequalities, not only the existence of a feasible input (Ames et al., 2017; Xiao & Belta, 2019). For relative-degree-two rows, the initial state must also satisfy $\psi_{j,1}(z(0)) \geq 0$ in addition to $h_j(z(0)) \geq 0$.

Proposition 1 (Conditional continuous-time safety)**.** *Let a closed-loop trajectory exist on $[0, T]$ as an absolutely continuous solution. Suppose its relative-degree-one barriers satisfy $\dot{h}_j \geq -\alpha_j(h_j)$ almost everywhere. For every relative-degree-two barrier, suppose $\psi_{j,1} = \dot{h}_j + \alpha_{j,1}(h_j)$ is absolutely continuous and $\dot{\psi}_{j,1} \geq -\alpha_{j,2}(\psi_{j,1})$ almost everywhere. Let the comparison functions be locally Lipschitz extended class-$\mathcal{K}$ functions. If initially all $h_j \geq 0$ and all relevant $\psi_{j,1} \geq 0$, then all h_j remain nonnegative on $[0, T]$.*

Proof. The scalar comparison lemma preserves nonnegativity of each relative-degree-one h_j. For a relative-degree-two row it first preserves $\psi_{j,1} \geq 0$, and then $\dot{h}_j \geq -\alpha_{j,1}(h_j)$ preserves $h_j \geq 0$. The argument applies to each simultaneously enforced row and therefore to their intersection. □

Remark 1 (A quadratic slack penalty is not hard enforcement). A hard-feasible state does not ensure that the relaxed QP in Eq. (24) applies a hard-feasible input. For example, $\min_{0 \leq u \leq 1, \delta \geq 0} \frac{1}{2}(u - 1)^2 + c\delta^2$ subject to $u \leq \delta$ has $u = \delta = 1/(1 + 2c) > 0$ for every finite $c > 0$, although the hard constraint $u \leq 0$ is feasible at $u = 0$. Thus hard feasibility and actual hard-row satisfaction must be logged separately. A hard-first implementation would constitute a different controller and would require a separate empirical evaluation.

Under sampled-data execution, satisfying rows only at update instants does not establish the almost-everywhere premises of Proposition 1. A sampled-data CBF or a verified intersample reachability bound is needed for that stronger claim. Neither the hard-QP audit nor the empirically measured "certified-safe episode" statistic is presented as a continuous-time safety proof.

Remark 2 (Scope and fallback). With the reported three-step rollout and eight-step planner interval, the analysis covers only the audited prefix, not all eight steps. The minimum-burden fallback retains the ordinary low-level CBF-QP but carries no additional horizon guarantee. Hard-QP feasibility is distinct from actual hard-row satisfaction by a relaxed-QP output, and sampled-state feasibility

does not itself imply intersample safety. Conditional CBF/HOCBF invariance, including the initial higher-order conditions, is stated separately in Appendix B.6.

C Evaluation Protocol and Metrics

Unless otherwise stated, the main experiments in Sections 4.1–4.3 are evaluated using 10 independent groups with 50 seeds per group. For each group, we compute each metric over its 50 seeds. Tables report the mean and standard deviation across the 10 groups. The parenthesized value is the standard deviation across groups, not a confidence interval and not a lower–upper range. When figures include error bars, they denote 95% confidence intervals unless stated otherwise.

QP-feasible episodes. For methods that solve a CBF-QP, a hard CBF-QP infeasible step occurs when the hard CBF constraints and the bounded-input constraints cannot be satisfied simultaneously. This test is performed without relying on the relaxation variable. Let $q_k = 1$ denote a hard CBF-QP infeasible step at time step k, and let T be the episode length. An episode is QP-feasible if it has no hard CBF-QP infeasible step:

$$F_{\mathrm{QP}} = \mathbf{1}\left\{\sum_{k=0}^{T-1} q_k = 0\right\}. \tag{59}$$

We report QP-feasible episodes as the percentage of episodes with $F_{\mathrm{QP}} = 1$. This metric is not reported for methods that do not solve a CBF-QP.

Safe episodes. A realized safety violation occurs when the minimum monitored CBF margin is negative. The monitored CBF margin takes the minimum over obstacle, pairwise-separation, boundary, and speed constraints. An episode is safe if it has no realized safety violation:

$$F_{\mathrm{safe}} = \mathbf{1}\left\{\min_{0\le k<T} m_k \ge 0\right\}, \tag{60}$$

where m_k is the minimum monitored CBF margin at step k. We report safe episodes as the percentage of episodes with $F_{\mathrm{safe}} = 1$.

Certified-safe episodes. Certified-safe episodes are episodes that are both QP-feasible and safe:

$$F_{\mathrm{cert}} = F_{\mathrm{QP}} F_{\mathrm{safe}}. \tag{61}$$

We report certified-safe episodes as the percentage of episodes with $F_{\mathrm{cert}} = 1$. This metric measures the conjunction of hard-QP feasibility and sampled barrier nonviolation; it is not a measurement of Eq. (10) or a continuous-time safety proof. This metric is reported for CBF-based methods in the main feasibility and selector-ablation experiments.

CBF-inactive steps. CBF-inactive steps measure how often the low-level CBF-QP does not actively constrain or modify the nominal command. Let $a_k^{\mathrm{cbf}} = 1$ if the CBF-QP is active at step k, and $a_k^{\mathrm{cbf}} = 0$ otherwise. We compute

$$S_{\mathrm{inactive}} = 100\left(1 - \frac{1}{T}\sum_{k=0}^{T-1} a_k^{\mathrm{cbf}}\right). \tag{62}$$

Higher CBF-inactive values indicate that the high-level decision is more compatible with the low-level CBF-QP. This metric is not reported for methods that do not solve a CBF-QP.

CBF activation steps. For additional analysis, we also report CBF activation steps:

$$N_{\mathrm{active}} = \sum_{k=0}^{T-1} a_k^{\mathrm{cbf}}. \tag{63}$$

This is the complementary count-level quantity behind CBF-inactive steps. Lower CBF activation steps correspond to higher CBF-inactive steps when the episode length is fixed.

QP-infeasible steps. QP-infeasible steps count how often the low-level CBF-QP loses hard feasibility:

$$N_{\text{inf}} = \sum_{k=0}^{T-1} q_k. \tag{64}$$

This quantity is used for additional analysis, while the main paper-facing tables use QP-feasible episodes.

Safety-violation steps. Safety-violation steps count how often the realized trajectory violates at least one monitored CBF constraint:

$$N_{\text{viol}} = \sum_{k=0}^{T-1} \mathbf{1}\{m_k < 0\}. \tag{65}$$

This quantity is used for additional analysis, while the main paper-facing tables use safe episodes.

Task cost. For coverage-style tasks, the raw task cost is the tail-averaged coverage cost. Let $J_{\text{cov}}(p_k)$ be the coverage cost at step k. We compute

$$J_{\text{tail}} = \frac{1}{T_{\text{tail}}} \sum_{k=T-T_{\text{tail}}}^{T-1} J_{\text{cov}}(p_k), \tag{66}$$

where T_{tail} is the number of final steps used for tail averaging. Lower J_{tail} indicates better final coverage.

Failure-aware task completion. Raw task cost alone can be misleading when a method continues to make task progress after losing the hard CBF-QP feasibility certificate. Therefore, in the task-transfer experiment, task completion is computed from a failure-aware task score.

For each task setting s, let J_i be the raw tail task cost of seed i, and let F_i be the episode-level failure indicator. For CBF-based methods, $F_i = 1$ if the episode has any hard CBF-QP infeasible step. For methods without a CBF-QP, $F_i = 1$ if the episode has any realized safety violation when an episode-level failure score is needed. We define the failure-penalized task cost as

$$J_i^{\text{fail}} = \begin{cases} J_i, & F_i = 0, \\ J_{\text{penalty}}(s), & F_i = 1, \end{cases} \tag{67}$$

where $J_{\text{penalty}}(s)$ is chosen larger than the largest raw task cost observed in the same setting. Thus, a method cannot obtain a favorable task score by making progress only after losing the hard CBF-QP feasibility certificate.

The normalized failure-penalized regret is

$$R_i^{\text{fail}} = \max\left\{0, \frac{J_i^{\text{fail}}}{J_{\text{ref}}(s)} - 1\right\}, \tag{68}$$

where $J_{\text{ref}}(s)$ is the best successful raw task cost in the same task setting. For the task-transfer results in Section 4.3, we convert this regret into a higher-is-better task-completion score:

$$S_{\text{task}} = \frac{100}{1 + R_i^{\text{fail}}}. \tag{69}$$

The main task-transfer figure and table report S_{task} as Task completion (%). Higher task completion means better failure-aware task performance. QP-failed episodes are treated as unsuccessful executions through J_i^{fail}.

Metric orientation. The main success-oriented metrics are QP-feasible episodes, safe episodes, certified-safe episodes, CBF-inactive steps, and task completion. All of these metrics are higher-is-better. Additional count and cost quantities, such as QP-infeasible steps, safety-violation steps, CBF activation steps, and raw task cost, are lower-is-better unless otherwise stated.

Table 2: **Hard-setting grouped success metrics for 3-D swarm coverage.** Values are mean and standard deviation across 10 independent groups with 50 seeds per group. Predictive H-CBF gives the best success-oriented metrics in the hardest setting.

Method	QP-feasible episodes (%) ↑	Safe episodes (%) ↑	Certified-safe episodes (%) ↑	CBF-inactive steps (%) ↑
Reactive CBF	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	37.21 (0.76)
Feedback H-CBF	22.00 (3.40)	36.00 (4.99)	16.00 (3.13)	41.36 (1.98)
Predictive H-CBF	**65.80** (**3.33**)	**78.60** (**1.65**)	**65.40** (**3.41**)	**47.47** (**1.58**)

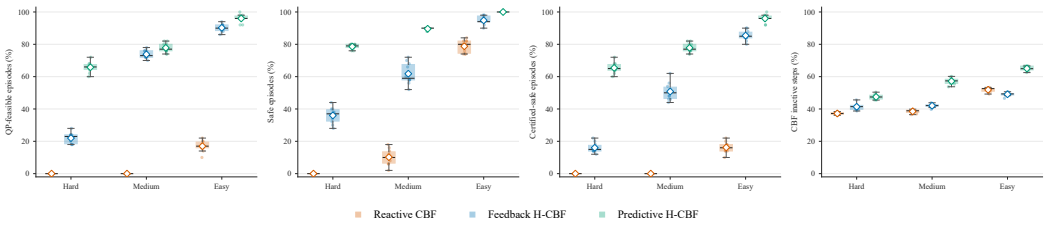

Figure 6: **Distribution of episode-level success metrics for Experiment 4.1.** We summarize QP-feasible episodes, safe episodes, certified-safe episodes, and CBF-inactive steps across 10 independent groups with 50 seeds per group. Predictive H-CBF improves the episode-level feasibility and safety metrics across the difficulty levels, rather than only improving a single average statistic.

D Additional Results for Experiment 1

This appendix provides additional analyses for Experiment 4.1. The main text reports the paper-facing episode-level results; Table 2 here provides the Hard-setting grouped statistics. Here we provide complementary distribution-level results, raw-metric summaries, focused QP-feasibility analyses, temporal statistics, safety-gap decompositions, and representative trajectories. All grouped statistics are computed from 10 independent groups with 50 seeds per group. Unless otherwise stated, values in parentheses denote standard deviation across the 10 groups.

The goal of this appendix is to clarify the closed-loop behavior behind the aggregate results in Figure 3. In particular, we examine whether Predictive H-CBF improves episode-level feasibility and safety only on average, or whether the effect remains visible across seed groups, hard-setting temporal traces, and representative hard episodes.

Distribution of episode-level success metrics. Figure 6 reports the group-level distribution of QP-feasible episodes, safe episodes, certified-safe episodes, and CBF-inactive steps. QP-feasible episodes are episodes without hard CBF-QP infeasibility. Safe episodes are episodes without realized safety-violation steps. Certified-safe episodes satisfy both conditions. CBF-inactive steps measure the fraction of steps in which the low-level CBF-QP does not actively constrain or modify the nominal command. The distributional view confirms the main-text conclusion: Predictive H-CBF consistently improves certificate-level feasibility and safety across Easy, Medium, and Hard settings.

Table 3 gives the corresponding grouped success-rate statistics. The Hard setting is the most informative: Reactive CBF has no QP-feasible or certified-safe episodes, while Feedback H-CBF partially improves feasibility. Predictive H-CBF further increases QP-feasible episodes to 65.80%, safe episodes to 78.60%, and certified-safe episodes to 65.40%.

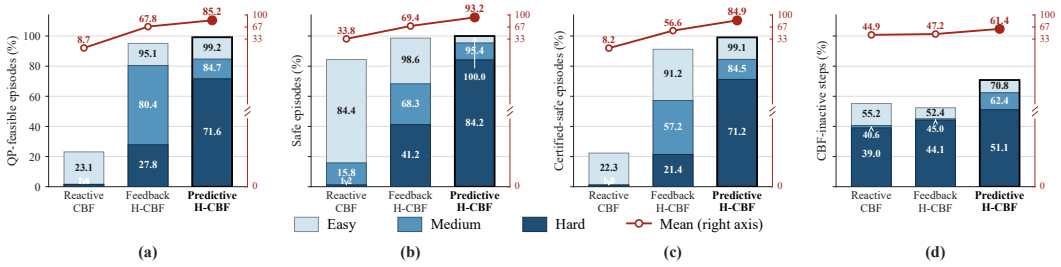

Figure 7: **Experiment 1 with Gemini 2.5 Flash.** All metrics are higher-is-better. Colored bars show separate rates for Easy, Medium and Hard settings, not additive percentages. The red curve summarizes the displayed settings.

Table 3: **Grouped episode-level success metrics for Experiment 4.1.** Values are mean and standard deviation across 10 independent groups with 50 seeds per group. QP-feasible episodes have no hard CBF-QP infeasible step. Certified-safe episodes are both QP-feasible and free of realized safety violations. CBF-inactive steps measure how often the low-level CBF-QP does not actively constrain the nominal command.

Setting	Method	QP-feasible eps. ↑	Safe eps. ↑	Certified-safe eps. ↑	CBF inactive steps ↑
Hard	Reactive CBF	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	37.21 (0.76)
Hard	Feedback H-CBF	22.00 (3.40)	36.00 (4.99)	16.00 (3.13)	41.36 (1.98)
Hard	Predictive H-CBF	**65.80 (3.33)**	**78.60 (1.65)**	**65.40 (3.41)**	**47.47 (1.58)**
Medium	Reactive CBF	0.00 (0.00)	10.20 (4.94)	0.00 (0.00)	38.52 (1.12)
Medium	Feedback H-CBF	73.80 (2.57)	61.80 (6.70)	50.80 (5.51)	42.02 (1.12)
Medium	Predictive H-CBF	**77.80 (3.05)**	**89.60 (0.84)**	**77.80 (3.05)**	**57.21 (2.00)**
Easy	Reactive CBF	17.00 (3.43)	78.80 (3.91)	16.20 (3.46)	51.86 (1.33)
Easy	Feedback H-CBF	90.20 (2.39)	94.80 (2.70)	85.40 (3.27)	49.09 (1.17)
Easy	Predictive H-CBF	**96.00 (2.49)**	**100.00 (0.00)**	**96.00 (2.49)**	**65.05 (1.49)**

Sensitivity to the LLM backend. Figure 7 and Table 4 report the backend comparison. In the Hard setting, Predictive H-CBF reaches 71.60% QP-feasible episodes and 71.20% certified-safe episodes, compared with 27.80% and 21.40% under Feedback H-CBF. Tables S25–S26 provide the additional joint-event analysis.

Distribution of raw metrics. Figure 8 reports the distribution of raw paper-facing metrics: QP failure rate, safety-violation steps, CBF activation steps, and failure-penalized tail cost. This complements the episode-level success view above. The raw-metric distribution shows that Predictive H-CBF reduces not only binary failure events, but also the intensity of CBF-QP feasibility-cost when episodes become difficult.

Table 5 provides the grouped numerical summary. In the Hard setting, Predictive H-CBF reduces the QP failure episode rate from 100.00% under Reactive CBF and 78.00% under Feedback H-CBF to 34.20%. It also reduces CBF activation steps and failure-penalized tail cost, showing that the improvement is not obtained by merely freezing or over-constraining the swarm.

Table 4: **Experiment 1 with Gemini 2.5 Flash.** The reported means and standard deviations are shown for each difficulty and controller variant.

Setting	Method	QP-feasible episodes (%) ↑	Safe episodes (%) ↑	Certified-safe episodes (%) ↑	CBF-inactive steps (%) ↑
Hard	Reactive CBF	1.40 (0.80)	1.20 (0.70)	1.00 (0.60)	39.05 (0.82)
Hard	Feedback H-CBF	27.80 (3.25)	41.20 (4.70)	21.40 (3.05)	44.15 (1.84)
Hard	Predictive H-CBF	**71.60 (3.10)**	**84.20 (1.52)**	**71.20 (3.22)**	**51.10 (1.45)**
Medium	Reactive CBF	1.60 (0.85)	15.80 (4.55)	1.20 (0.70)	40.60 (1.05)
Medium	Feedback H-CBF	80.40 (2.42)	68.30 (6.15)	57.20 (5.10)	45.05 (1.05)
Medium	Predictive H-CBF	**84.70 (2.85)**	**95.40 (0.72)**	**84.50 (2.88)**	**62.40 (1.82)**
Easy	Reactive CBF	23.10 (3.25)	84.40 (3.70)	22.30 (3.28)	55.20 (1.22)
Easy	Feedback H-CBF	95.10 (2.10)	98.60 (1.18)	91.20 (2.96)	52.40 (1.05)
Easy	Predictive H-CBF	**99.20 (1.10)**	**100.00 (0.00)**	**99.10 (1.12)**	**70.80 (1.32)**

Table 5: **Grouped raw metrics for Experiment 4.1.** Values are mean and standard deviation across 10 independent groups with 50 seeds per group. QP failure episodes report the percentage of episodes with at least one hard CBF-QP infeasible step. Failure-penalized cost prevents a method from being rewarded for low coverage cost after losing the hard CBF-QP feasibility certificate.

Setting	Method	QP failure eps. ↓	Safety violations ↓	CBF activations ↓	Failure-penalized cost ↓
Hard	Reactive CBF	100.00 (0.00)	46.59 (0.67)	156.97 (1.90)	11.348 (0.000)
Hard	Feedback H-CBF	78.00 (3.40)	18.34 (2.20)	146.60 (4.96)	10.145 (0.203)
Hard	Predictive H-CBF	**34.20 (3.33)**	**13.36 (1.98)**	**131.32 (3.95)**	**6.982 (0.228)**
Medium	Reactive CBF	100.00 (0.00)	40.20 (2.16)	153.69 (2.79)	10.684 (0.000)
Medium	Feedback H-CBF	26.20 (2.57)	12.20 (1.19)	144.96 (2.81)	5.351 (0.201)
Medium	Predictive H-CBF	**22.20 (3.05)**	**6.40 (1.16)**	**106.97 (5.01)**	**4.818 (0.204)**
Easy	Reactive CBF	83.00 (3.43)	4.80 (1.55)	125.15 (3.46)	4.975 (0.075)
Easy	Feedback H-CBF	9.80 (2.39)	0.81 (0.55)	132.36 (3.03)	3.060 (0.065)
Easy	Predictive H-CBF	**4.00 (2.49)**	**0.00 (0.00)**	**90.86 (3.86)**	**2.774 (0.066)**

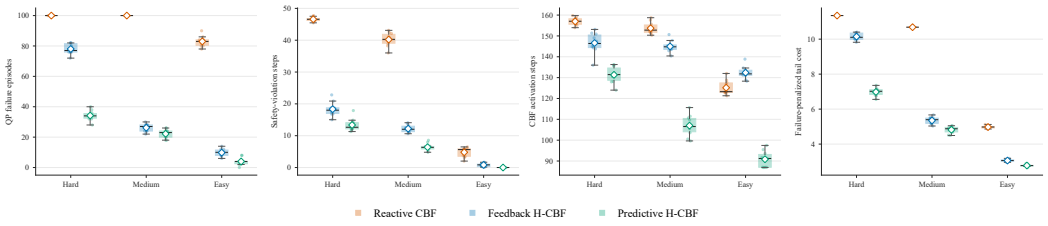

Figure 8: **Distribution of raw metrics for Experiment 4.1.** We show QP failure rate, safety-violation steps, CBF activation steps, and failure-penalized tail cost across grouped runs. Predictive H-CBF reduces hard-QP certificate failures and repeated CBF activation while maintaining lower failure-aware task cost.

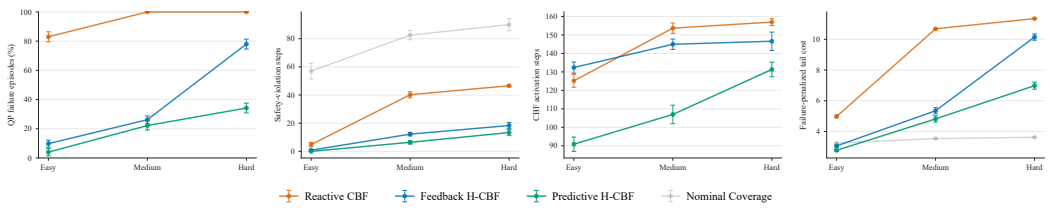

Figure 9: **Raw-metric trends across difficulty levels.** We report QP failure rate, safety-violation steps, CBF activation steps, and failure-penalized tail cost from Easy to Hard settings. This plot complements the main-text episode-level success metrics by showing the raw CBF-QP activity and task-cost behavior.

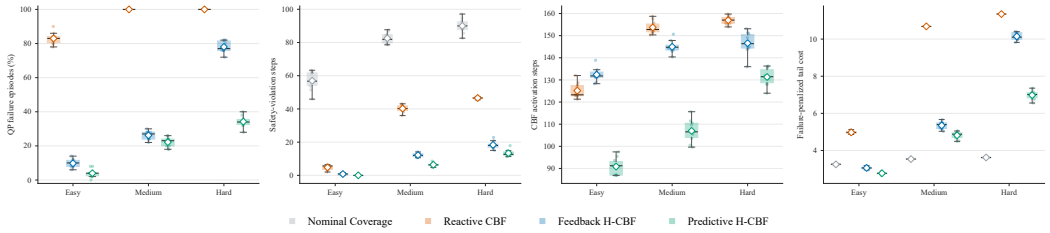

Figure 10: **Grouped variability of raw metrics.** Boxplots summarize the 10 independent groups for each method and difficulty level. Predictive H-CBF shows lower QP failure and lower failure-penalized cost across the grouped benchmark, especially in the Hard setting.

Grouped trend and variability across difficulty. Figure 9 shows the same four raw metrics as trends across Easy, Medium, and Hard settings. This figure is a complete raw-metric counterpart to the main-text episode-level success figure. As the difficulty increases, Reactive CBF rapidly reaches near-certain QP failure. Feedback H-CBF reduces failure in Easy and Medium settings, but remains fragile in the Hard stress setting. Predictive H-CBF gives the lowest QP failure rate and lowest CBF activation among the CBF-based variants across the grouped benchmark.

Figure 10 shows group-level variability using boxplots. The boxplots indicate that the improvements of Predictive H-CBF are not caused by a single favorable group. The Hard-setting distributions show a clear separation between Predictive H-CBF and the other CBF-based methods in QP failure and failure-penalized cost.

Certificate-level QP-feasibility analysis. Figure 11 consolidates the certificate-level QP-feasibility metrics into a single three-panel view. Panel (a) reports the percentage of episodes without hard CBF-QP infeasibility, panel (b) shows the complementary QP-failure trend, and panel (c) summarizes group-level variability across the 10 independent seed groups. Across all three views, Predictive H-CBF gives the strongest feasibility profile, especially in the Hard setting where Reactive CBF almost always loses the hard feasibility certificate.

Hard-setting temporal statistics. Figure 12 summarizes closed-loop time histories over 10 hard episodes selected from the same grouped evaluation pool. Solid curves report the mean trajectory and shaded bands report the standard error of the mean. This temporal view is more informative than a single representative rollout: it shows when methods lose the hard CBF-QP certificate, how safety margins evolve before failure-aware stopping, how much CBF intervention accumulates, and how task cost changes during the episode. Reactive CBF loses QP feasibility early and repeatedly in the hard tier because the nominal coverage objective can keep driving the swarm toward low-clearance regions. Feedback H-CBF reduces repeated CBF activation, but still exhibits substantial certificate loss and high cumulative active steps. Predictive H-CBF maintains the highest failure-free episode fraction while reaching lower final task cost, indicating that its improvement is not obtained by simply freezing the swarm or over-constraining the controller.

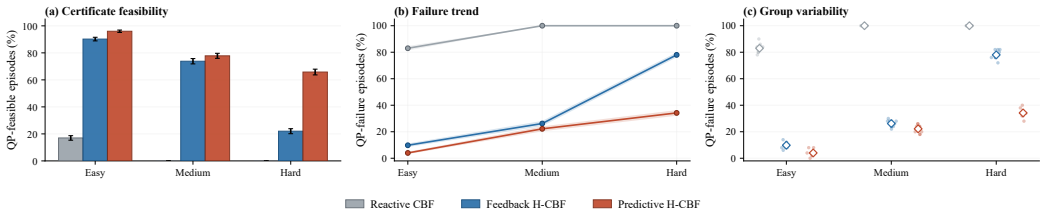

Figure 11: **QP-feasibility analysis for Experiment 4.1.** The composite figure reports certificate feasibility, the complementary failure-rate trend, and group-level variability using the same 10-by-50 grouped evaluation protocol. Higher is better in panel (a), while lower is better in panels (b) and (c). Predictive H-CBF substantially improves certificate feasibility and reduces hard CBF-QP certificate loss relative to Reactive CBF and Feedback H-CBF.

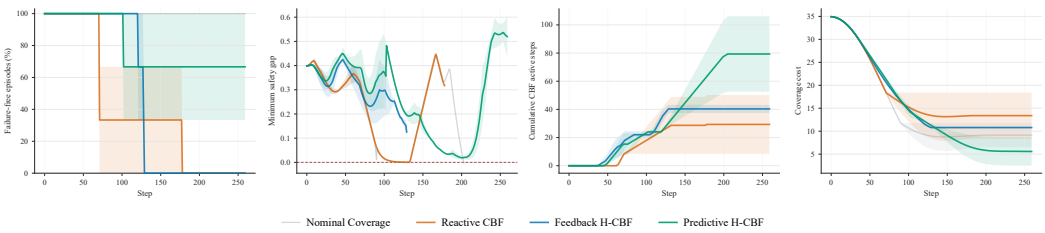

Figure 12: **Hard-setting temporal statistics for Experiment 4.1.** Curves show the mean over 10 hard episodes and shaded bands show the standard error of the mean. We report the percentage of episodes that remain free of hard CBF-QP failure, the minimum safety margin, cumulative CBF-active steps, and coverage cost. Predictive H-CBF maintains a higher failure-free fraction and lower task cost while avoiding the large intervention burden of purely reactive filtering.

Figure 13 decomposes a representative hard rollout into obstacle, pairwise, and speed-related margins. This single-rollout decomposition is not used as aggregate evidence. Instead, it explains which constraint families make the hard setting difficult observed in Figure 12. The curves show that the failure is not caused by one isolated constraint type. Obstacle and speed margins become tight as the swarm moves through bottlenecks under input limits, and pairwise margins can also become active as agents compress around obstacles. This supports using a CBF feedback vector rather than a single warning scalar: different CBF feedback terms expose different sources of low-level CBF-QP difficulty

Representative hard case studies. Figure 14 and Figure 15 show two representative hard-setting case studies. These examples follow the same failure-aware evaluation protocol used in the quantitative results. A trajectory is truncated at the first safety violation or hard-QP infeasible step, while

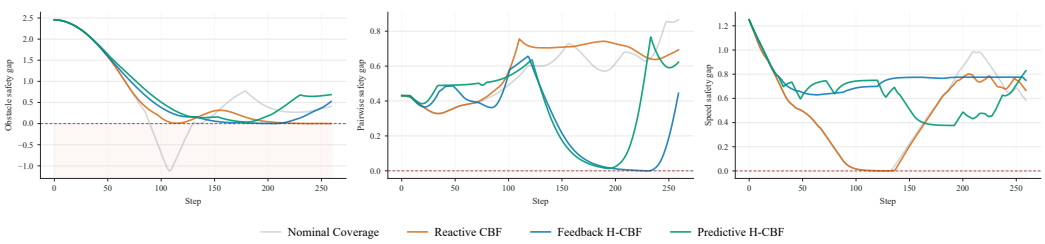

Figure 13: **Safety-gap components in the representative hard setting.** We decompose the monitored safety gap into obstacle, pairwise, and speed-related margins. This representative rollout shows that hard-setting difficulty involves multiple safety sources rather than one isolated constraint. The result motivates feeding a CBF feedback vector ϕ_k, rather than a single warning scalar, back to the harness.

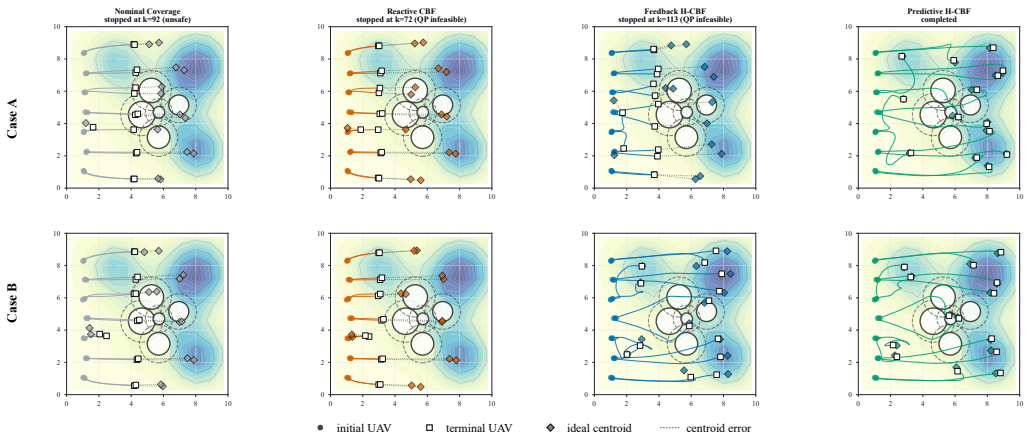

Figure 14: **Representative hard-setting case studies in top-down view.** We show two hard episodes under the failure-aware evaluation protocol. Trajectories are truncated at the first safety violation or hard-QP infeasible step; completed episodes are shown until the end of the horizon. Solid markers denote initial UAV positions, open square markers denote terminal positions, diamond markers denote terminal density-weighted Voronoi centroids, and dotted segments denote terminal-to-centroid coverage error. Case A shows Feedback H-CBF stopping at $k = 113$ when the QP becomes infeasible, whereas Predictive H-CBF completes the episode. Case B shows that Predictive H-CBF can remain feasible and safe when Reactive CBF and Feedback H-CBF fail.

completed episodes are shown until the end of the horizon. Thus, the terminal markers indicate either the first failure state or the final completed state. A safety violation means that at least one monitored safety margin becomes negative. A hard-QP infeasible step means that no bounded acceleration satisfies all hard CBF constraints.

The case studies illustrate two difficult failure modes. In Case A, Reactive CBF becomes hard-QP infeasible because the nominal coverage objective repeatedly drives the swarm toward low-clearance regions. Feedback H-CBF delays the failure, but still loses the hard CBF-QP feasibility certificate. Predictive H-CBF completes the episode while achieving lower CBF activation and smaller terminal centroid error. This shows that direct CBF feedback can help, but short-horizon predictive selection gives a stronger safety-performance tradeoff.

Case B is a more difficult stress case. Reactive CBF again becomes hard-QP infeasible, and Feedback H-CBF still produces residual infeasible and unsafe steps. Predictive H-CBF completes the episode with no hard-QP infeasible step and no safety violation. It also achieves the lowest tail coverage cost and centroid error. This example shows that Predictive H-CBF is not simply more conservative. It can preserve feasibility while maintaining stronger task completion.

Overall, these additional results support the mechanism claimed in the main text. Reactive CBF treats infeasibility only after the nominal command has reached the low-level CBF-QP. Feedback H-CBF uses low-level CBF feedback to revise future high-level decisions, but direct feedback can become conservative or remain fragile in the hardest grouped setting. Predictive H-CBF uses the same low-level CBF-QP inside a short-horizon evaluator, so high-level candidates that would create low margin, large intervention, repeated active constraints, or hard infeasibility are rejected before execution. This explains why Predictive H-CBF improves both episode-level certificate metrics and raw CBF-QP activity metrics in the grouped Experiment 4.1 benchmark.

E Additional Results for Experiment 2

This appendix provides additional results for Experiment 4.2. The main text reports the failure-aware selector-ablation results using QP-feasible episodes, certified-safe episodes, CBF-inactive steps, and failure-penalized tail coverage cost. Here we provide complementary raw-metric distributions, group-level success-rate distributions, and selector-term summary.

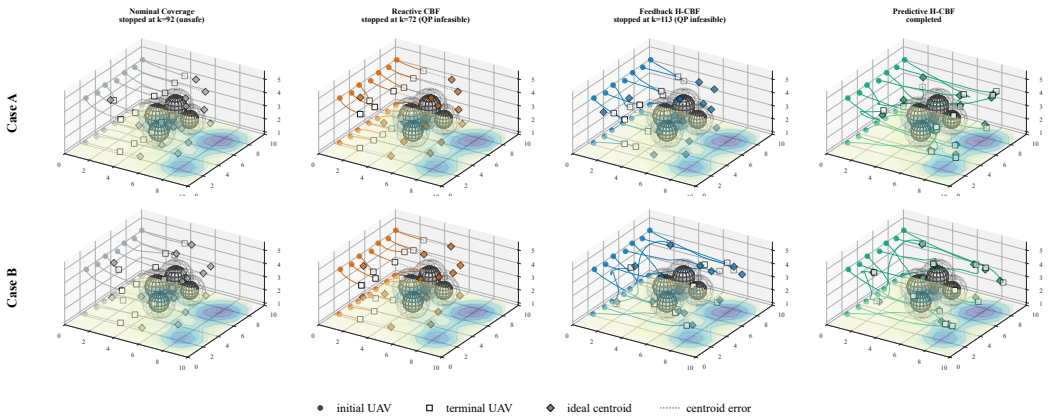

Figure 15: **Representative hard-setting case studies in 3-D view.** The 3-D view complements Figure 14 by showing altitude motion and spatial separation. This view helps verify that the apparent top-down paths are not projection artifacts. Predictive H-CBF uses short-horizon CBF-aware candidate evaluation to avoid low-margin, high-intervention, and hard-QP-infeasible high-level decisions while preserving task progress.

The selector ablation uses the same grouped API-evaluated protocol as the main text. Each setting is evaluated over 10 independent groups, with 50 task seeds per group. All selector variants use the same API-generated high-level candidate representation and differ only in the selection rule. *Feedback H-CBF* directly revises the next high-level decision from the current CBF feedback. *Random selector* samples uniformly from the shared candidate set. *Fixed selector* always chooses the balanced candidate. *Progress-only selector* selects the candidate with the best predicted coverage progress. *Predictive H-CBF* selects the candidate using predicted coverage progress, CBF margin, QP feasibility, CBF intervention, and active constraints.

The purpose of this appendix is to separate three effects. First, random and fixed selection test whether candidate diversity alone explains the improvement. Second, progress-only selection tests whether predicted task progress alone is sufficient. Third, Predictive H-CBF tests whether task progress must be evaluated jointly with predicted CBF-QP burden score. The results show that the answer is the third case: candidate generation alone is insufficient, and task-only scoring can be misleading when the selected candidate loses the hard CBF-QP feasibility certificate.

Full metric distributions. Figure 16 reports the full grouped metrics for the selector ablation. We include QP failure rate, QP-infeasible steps, safety-violation steps, CBF activation steps, failure-penalized tail coverage cost, and raw tail coverage cost. The raw tail coverage cost is included only as an auxiliary reference. It should not be used alone to compare methods, because a selector may obtain a low raw task cost after losing the hard CBF-QP feasibility certificate.

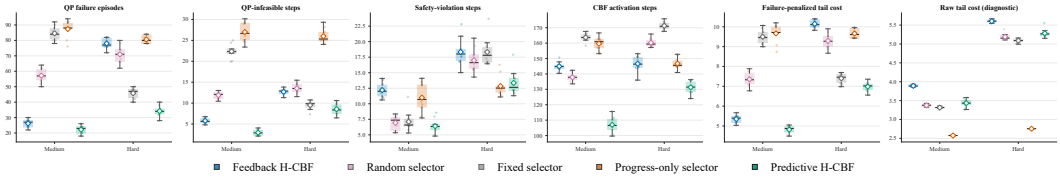

Figure 16: **Full grouped metrics for the Experiment 4.2 selector ablation.** We report QP failure rate, QP-infeasible steps, safety-violation steps, CBF activation steps, failure-penalized tail coverage cost, and raw tail coverage cost across 10 independent groups with 50 seeds per group. Raw tail cost is shown as an auxiliary reference. Progress-only selector can obtain low raw task cost, but it does so with high QP failure rate and high QP-infeasible steps. Predictive H-CBF achieves the best failure-aware tradeoff by reducing QP failure, CBF activation, and failure-penalized cost.

Table 6 gives the corresponding grouped numerical values. The contrast between raw tail cost and failure-penalized cost is especially important. In the Hard setting, Progress-only selector obtains the lowest raw tail coverage cost, but it has an 80.60% QP failure rate and 25.99 QP-infeasible steps on average. After infeasible episodes are treated as failures, its failure-penalized cost becomes much worse than that of Predictive H-CBF. This supports the use of CBF-aware predictive scoring rather than task-progress-only scoring.

Table 6: **Grouped raw metrics for the Experiment 4.2 selector ablation.** Values are mean with standard deviation in parentheses over 10 independent groups, with 50 task seeds per group. QP failure rate is the percentage of episodes in a group with at least one hard CBF-QP infeasible step. Failure-penalized tail cost treats hard-QP-infeasible episodes as task failures. Raw tail cost is reported only as an auxiliary reference.

Setting	Selector	QP failure rate ↓	QP-infeasible steps ↓	Safety violations ↓	CBF activations ↓	Failure-penalized cost ↓	Raw tail cost ↓
Medium	Feedback H-CBF	26.20 (2.57)	5.77 (0.66)	12.20 (1.19)	144.96 (2.81)	5.351 (0.201)	3.891 (0.028)
Medium	Random selector	57.20 (4.54)	11.90 (0.88)	6.93 (1.19)	137.88 (2.72)	7.356 (0.358)	3.375 (0.036)
Medium	Fixed selector	84.60 (4.53)	22.38 (1.61)	7.12 (1.63)	163.68 (2.53)	9.494 (0.352)	3.314 (0.024)
Medium	Progress-only selector	87.40 (5.50)	27.00 (2.29)	10.98 (2.33)	159.65 (4.24)	9.663 (0.445)	**2.575 (0.008)**
Medium	Predictive H-CBF	**22.20 (3.05)**	**2.94 (0.70)**	**6.40 (1.16)**	**106.97 (5.01)**	**4.818 (0.204)**	3.435 (0.103)
Hard	Feedback H-CBF	78.00 (3.40)	12.71 (0.72)	18.34 (2.20)	146.60 (4.96)	10.145 (0.203)	5.605 (0.038)
Hard	Random selector	71.00 (5.68)	13.54 (1.20)	17.00 (1.90)	160.21 (2.77)	9.289 (0.383)	5.188 (0.086)
Hard	Fixed selector	45.80 (3.82)	9.51 (1.02)	18.33 (2.23)	171.24 (2.48)	7.402 (0.250)	5.088 (0.051)
Hard	Progress-only selector	80.60 (2.67)	25.99 (1.72)	**12.85 (1.63)**	146.71 (3.65)	9.658 (0.233)	**2.754 (0.005)**
Hard	Predictive H-CBF	**34.20 (3.33)**	**8.57 (1.30)**	13.36 (1.98)	**131.32 (3.95)**	**6.982 (0.228)**	5.282 (0.110)

Success-rate distributions. Figure 17 reports the group-level distributions of QP-feasible episodes, safe episodes, certified-safe episodes, and CBF-inactive steps. This plot complements the main-text bar chart by showing variability across the 10 independent groups. Predictive H-CBF is not only better in the aggregate; its distribution is shifted toward higher QP-feasible and certified-safe episode rates in both Medium and Hard settings.

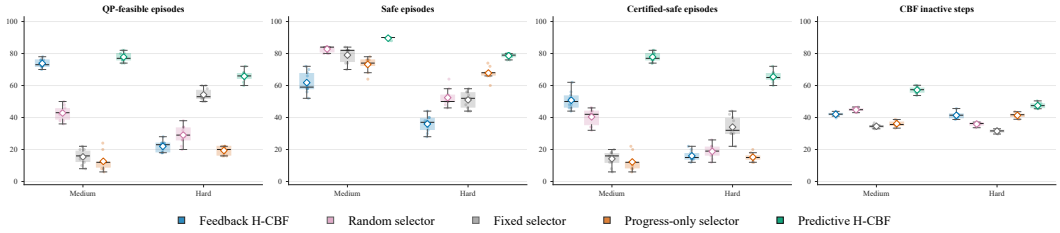

Figure 17: **Distribution of success-oriented metrics for the Experiment 4.2 selector ablation.** We summarize QP-feasible episodes, safe episodes, certified-safe episodes, and CBF-inactive steps across 10 independent groups with 50 seeds per group. Predictive H-CBF consistently improves certificate-level feasibility and safety, while also increasing the fraction of steps in which the CBF-QP remains inactive.

Table 7 gives the grouped success-rate values. In the Medium setting, Predictive H-CBF improves certified-safe episodes from 50.80% under Feedback H-CBF to 77.80%. In the Hard setting, Predictive H-CBF improves certified-safe episodes from 16.00% under Feedback H-CBF to 65.40%. This shows that predictive selection improves both QP feasibility and realized safety, not only raw task progress.

Selector-term summary. Figure 18 summarizes which scoring terms are used by each selector. This figure clarifies that the ablation is not comparing different candidate pools. Random selector, Fixed selector, Progress-only selector, and Predictive H-CBF all operate on the same API-generated high-level candidate representation. The difference is the selection rule. Random selector uses no predictive score. Fixed selector always selects one predefined candidate. Progress-only selector uses only the predicted task-progress term. Predictive H-CBF uses the full CBF-aware predictive score, including predicted coverage progress, CBF margin, QP feasibility, CBF intervention, and active constraints.

Table 7: **Grouped success-oriented metrics for the Experiment 4.2 selector ablation.** Values are mean with standard deviation in parentheses over 10 independent groups, with 50 task seeds per group. QP-feasible episodes have no hard CBF-QP infeasible step. Certified-safe episodes are both QP-feasible and free of realized safety violations.

Setting	Selector	QP-feasible eps. ↑	Safe eps. ↑	Certified-safe eps. ↑	CBF inactive steps ↑
Medium	Feedback H-CBF	73.80 (2.57)	61.80 (6.70)	50.80 (5.51)	42.02 (1.12)
Medium	Random selector	42.80 (4.54)	82.80 (1.93)	40.40 (5.23)	44.85 (1.09)
Medium	Fixed selector	15.40 (4.53)	79.00 (4.74)	14.20 (4.57)	34.53 (1.01)
Medium	Progress-only selector	12.60 (5.50)	73.20 (4.24)	12.20 (5.12)	36.14 (1.70)
Medium	Predictive H-CBF	**77.80 (3.05)**	**89.60 (0.84)**	**77.80 (3.05)**	**57.21 (2.00)**
Hard	Feedback H-CBF	22.00 (3.40)	36.00 (4.99)	16.00 (3.13)	41.36 (1.98)
Hard	Random selector	29.00 (5.68)	52.40 (5.32)	18.80 (4.13)	35.92 (1.11)
Hard	Fixed selector	54.20 (3.82)	51.00 (5.19)	34.00 (6.80)	31.50 (0.99)
Hard	Progress-only selector	19.40 (2.67)	67.80 (3.71)	15.20 (2.53)	41.31 (1.46)
Hard	Predictive H-CBF	**65.80 (3.33)**	**78.60 (1.65)**	**65.40 (3.41)**	**47.47 (1.58)**

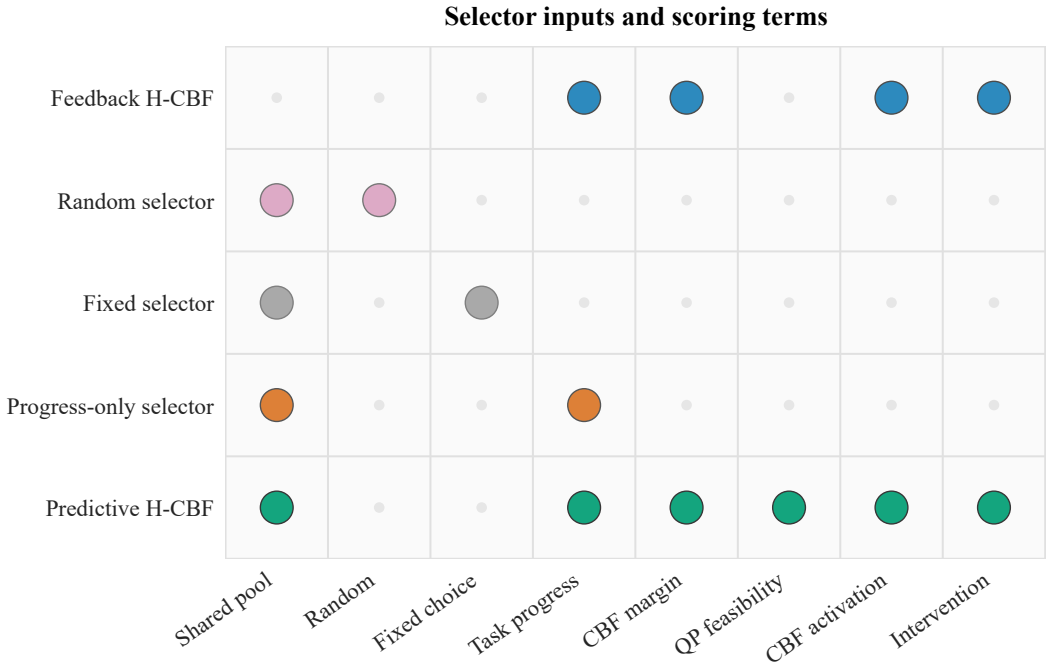

Figure 18: **Selector terms used in the Experiment 4.2 ablation.** All selector variants use the same API-generated high-level candidate representations and differ only in the scoring terms. Progress-only selector uses predicted task progress but ignores CBF-QP burden score. Predictive H-CBF evaluates task progress jointly with predicted CBF margin, QP feasibility, intervention, and active constraints.

The gap between Progress-only selector and Predictive H-CBF isolates the contribution of CBF-aware predictive selection. Progress-only selector can reduce raw tail coverage cost because it aggressively selects task-progressive candidates. However, those candidates often induce high future CBF-QP burden score and hard-QP infeasibility. Once infeasible episodes are treated as failures, Progress-only selector performs substantially worse than Predictive H-CBF. Therefore, the improvement of Predictive H-CBF does not come from candidate diversity alone. It comes from selecting high-level candidates using future CBF margin, QP feasibility, intervention, active-constraint, and task-progress information together.

F ADDITIONAL RESULTS FOR EXPERIMENT 3

Table 8: Task-transfer performance. Means are aligned with the one-decimal labels in Figure 5. Each row summarizes 500 records; parentheses report standard deviations. Task scores and step counts follow the stated evaluation definitions.

Task	Method	QP-feasible episodes (%)	Safe episodes (%)	CBF-inactive steps (%)	Task score (%)
Static coverage	Reactive CBF	23.6 (6.52)	36.0 (6.04)	54.6 (0.52)	53.0 (3.60)
Static coverage	Feedback H-CBF	80.8 (5.51)	99.2 (1.40)	60.6 (0.65)	73.0 (2.32)
Static coverage	Predictive H-CBF	94.0 (3.27)	100.0 (0.00)	76.1 (0.50)	84.0 (1.53)
Dynamic density	Reactive CBF	0.4 (0.84)	4.0 (3.13)	38.3 (0.52)	46.7 (0.37)
Dynamic density	Feedback H-CBF	80.4 (5.64)	90.4 (4.97)	46.1 (0.56)	67.9 (1.62)
Dynamic density	Predictive H-CBF	86.0 (3.13)	95.2 (2.53)	53.4 (0.23)	79.2 (1.26)
Narrow passage	Reactive CBF	0.0 (0.00)	0.0 (0.00)	30.6 (0.49)	54.1 (0.00)
Narrow passage	Feedback H-CBF	80.0 (7.83)	86.8 (4.83)	40.8 (0.48)	71.7 (1.75)
Narrow passage	Predictive H-CBF	89.2 (2.86)	99.6 (0.84)	60.7 (0.45)	80.4 (0.85)

This appendix provides additional details for Experiment 4.3. The purpose of this experiment is to test whether the same Harness-CBF mechanism transfers across different swarm-control task families. Unlike task-specific controllers, we do not introduce new harness rules for each task. The low-level CBF-QP, CBF feedback vector ϕ_k, candidate representation, and predictive evaluator are kept fixed. Only the task configuration is changed.

Figure 19 shows representative task snapshots. Static coverage uses a fixed spatial density field. Narrow passage introduces an obstacle-induced bottleneck that causes obstacle and pairwise margins to become tight at the same time. Dynamic density uses a time-varying density field, so the high-level task objective changes during execution. These settings test whether CBF feedback terms provide a task-agnostic planning signal for high-level candidate selection.

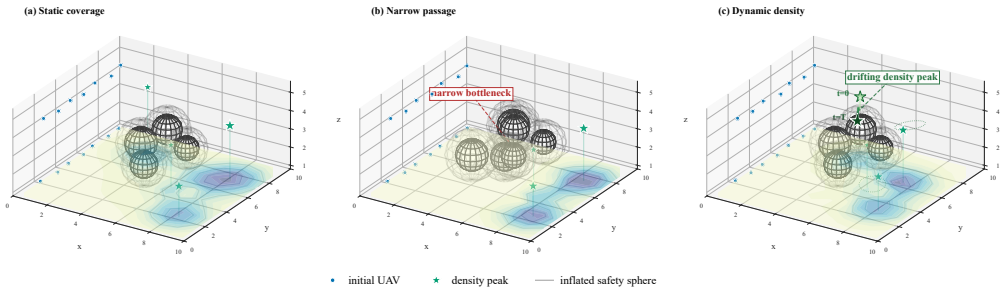

Figure 19: **Task snapshots for Experiment 4.3.** We evaluate static coverage, narrow-passage navigation, and dynamic-density coverage. All tasks use the same low-level CBF-QP, CBF feedback vector ϕ_k, high-level candidate representation, and predictive evaluator.

Figure 20 reports task-transfer metrics using a consistent higher-is-better convention. The panels are labeled QP-feasible episodes, safety-compliant steps, CBF-inactive steps, and task completion. The second panel is step-level, not an episode rate. Reactive CBF remains brittle in the narrow-passage and dynamic-density tasks, whereas Feedback H-CBF improves feasibility but still induces substantial CBF intervention. Predictive H-CBF gives the strongest transfer profile across all task families, supporting the main-text claim that the harness is not merely tuned to one static coverage field.

Figure 21 visualizes the safety-completion tradeoff. The horizontal axis reports failure-aware task completion and the vertical axis reports certified-safe episodes; therefore, the upper-right region is preferred. Marker size is proportional to CBF-inactive steps, so larger markers indicate lower CBF-QP activity. The goal of Experiment 4.3 is not to minimize task cost at any safety cost, but to test whether the same Harness-CBF mechanism can preserve certificate-level safety while maintaining task progress across task families.

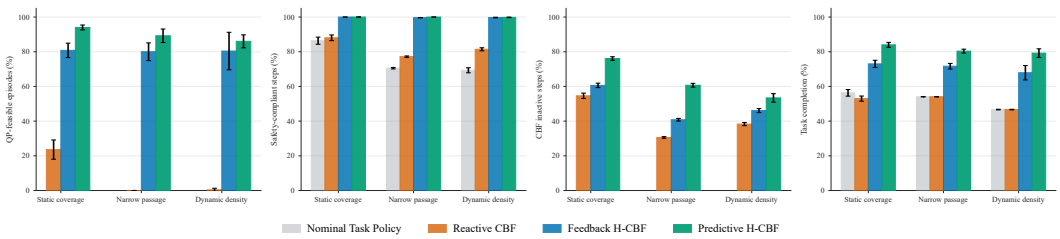

Figure 20: **Task-transfer metrics for Experiment 4.3.** All panels use a higher-is-better convention. The axes label QP-feasible episodes, safety-compliant steps, CBF-inactive steps and task completion. The second panel uses a step denominator. Predictive H-CBF gives the strongest overall transfer behavior across static coverage, narrow passage and dynamic density without relying on conservative freezing.

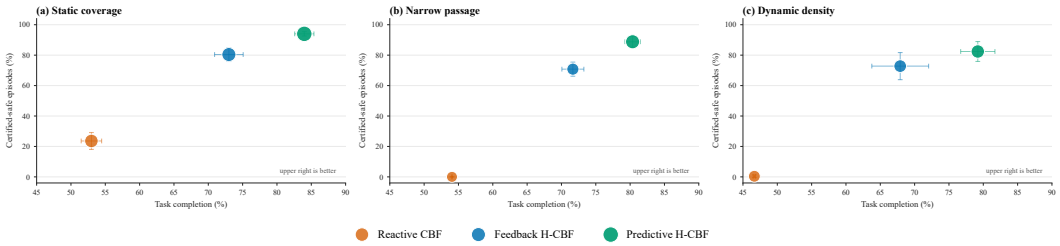

Figure 21: **Safety-completion tradeoff across task families.** Each point summarizes one CBF-based method on one task family. The x-axis is failure-aware task completion and the y-axis is certified-safe episodes, so the upper-right region is preferred. Marker size is proportional to CBF-inactive steps. Predictive H-CBF moves the tradeoff toward higher task completion, higher certified safety, and lower CBF-QP activity.

Overall, these results complement Experiment 4.3. They show that the same Harness-CBF mechanism transfers across static, bottleneck, and time-varying task objectives. The transfer is not free: preserving CBF-QP feasibility can increase normalized task regret. However, the failure-aware metrics and tradeoff plot show that Predictive H-CBF preserves substantially better task completion than reactive filtering while avoiding task-only progress that repeatedly drives the system into QP-infeasible regions.

G LLM Harness Prompt and API Specification

This appendix gives the complete prompt specification used by the LLM-based harness. The prompt has three parts: a fixed instruction template, a structured state-and-feedback payload, and a constrained JSON response schema. The LLM is not allowed to output raw control inputs. It only proposes high-level planning candidates. These candidates are later evaluated by the predictive evaluator and filtered by the low-level CBF-QP before execution. This design keeps the LLM as a high-level candidate generator, not as the final safety authority.

Prompt structure. Each LLM call is constructed as

$$\text{Prompt}_k = \{\text{Instruction},\ \text{StructuredPayload}_k,\ \text{ResponseSchema}\}.$$

The instruction defines the role of the LLM and the allowed output space. The structured payload provides the current task context, swarm state, CBF-QP feedback, and short-term memory. The response schema forces the LLM to return a finite set of high-level candidates in a fixed JSON format.

System / instruction prompt

You are a high-level planner for a multi-UAV coverage-control task.

You do not output raw control inputs. You only propose high-level planning candidates that will later be evaluated by a predictive evaluator and filtered by a low-level CBF-QP.
Use the current swarm state, task context, CBF-QP feedback, and recent memory to propose several high-level candidates.
Each candidate must specify:

- `speed_scale`
- `obstacle_repulse_scale`
- `pair_spread_scale`
- `altitude_lane_strength`
- `altitude_strategy`
- `safety_note`

Prefer candidates that maintain coverage performance while avoiding repeated CBF activation and QP infeasibility.
Return valid JSON only.

Structured payload. The structured payload is the part of the prompt that changes over time. It is generated by the harness at each planning step. It contains the current constraints, previous high-level decision, swarm summary, CBF-QP feedback, and short-term memory. The key point is that low-level CBF-QP feedback is explicitly included in the LLM input. Thus, the LLM can revise future high-level decisions according to how difficult previous decisions were for the low-level CBF-QP to realize.

Structured payload sent to the LLM


```
{
  "step": 1,
  "n_agents": 14,

  "constraints": {
    "speed_limit": 1.28,
    "accel_limit_componentwise": 0.26,
    "inter_agent_min_dist": 0.70,
    "domain_min": [0.0, 0.0, 0.8],
    "domain_max": [10.0, 10.0, 5.6]
  },

  "local_policy_guardrails": {
    "balanced_speed_floor_if_not_emergency": 0.32,
    "emergency_gap_threshold": 0.22,
    "candidate_evaluator":
      "The code will score all candidates with a predictive
      ↪  evaluator.",
    "planner_delay_steps": 0,
    "delay_aware_evaluator": false
  },

  "memory": {
    "ema_active": 0.0,
    "ema_intervention": 0.0,
    "ema_margin_decrease": 0.034,
    "ema_infeasible": 0.0,
    "ema_speed_pressure": 0.0
  },

  "current_decision": {
    "speed_scale": 0.66,
    "obstacle_repulse_scale": 1.00,
    "pair_spread_scale": 1.00,
    "altitude_lane_strength": 0.00,
    "altitude_strategy": "nominal",
    "safety_note": "nominal"
  },
```

```
  "swarm_summary": {
    "mean_speed": 0.0,
    "max_speed": 0.0,
    "mean_centroid_distance": 4.066,
    "max_centroid_distance": 6.483,
    "mean_position": [1.136, 4.705, 3.180]
  },

  "cbf_feedback": {
    "min_obstacle_gap": 2.495,
    "min_inter_agent_gap": 0.437,
    "min_boundary_gap": 0.636,
    "min_speed_gap": 1.280,
    "min_safety_gap": 0.437,
    "hard_feasible": true,
    "relaxation": 0.0,
    "intervention_norm": 0.0,
    "active_constraint_ratio": 0.0
  }
}
```


Response schema. The LLM response must contain a finite set of high-level candidates. Each candidate parameterizes the nominal controller:

$$\theta_k^{(i)} = \left[\gamma_k^{(i)}, \eta_{\text{obs},k}^{(i)}, \eta_{\text{pair},k}^{(i)}, \eta_{\text{alt},k}^{(i)}\right],$$

where $\gamma_k^{(i)}$ is the speed scale, $\eta_{\text{obs},k}^{(i)}$ is the obstacle-clearance scale, $\eta_{\text{pair},k}^{(i)}$ is the inter-agent-spreading scale, and $\eta_{\text{alt},k}^{(i)}$ is the altitude-lane strength. The `safety_note` is used only as an explanation of the candidate. It is not used as a safety certificate.

Required JSON response schema

```
{
  "candidates": [
    {
      "label": "balanced",
      "speed_scale": <float>,
      "obstacle_repulse_scale": <float>,
      "pair_spread_scale": <float>,
      "altitude_lane_strength": <float>,
      "altitude_strategy": "<nominal | mixed | layered>",
      "safety_note": "<short explanation>"
    },
    ...
  ]
}
```

Example LLM response. A representative response is shown below. The response contains several high-level candidates, not a final low-level control input.

Example LLM response


```
{
  "candidates": [
    {
      "label": "balanced",
      "speed_scale": 0.65,
      "obstacle_repulse_scale": 1.0,
      "pair_spread_scale": 1.0,
      "altitude_lane_strength": 0.1,
      "altitude_strategy": "nominal",
```

```
      "safety_note":
        "Maintain coverage progress with a slight safety-margin
        ↪  increase."
    },
    {
      "label": "safety_oriented",
      "speed_scale": 0.40,
      "obstacle_repulse_scale": 1.3,
      "pair_spread_scale": 1.2,
      "altitude_lane_strength": 0.5,
      "altitude_strategy": "mixed",
      "safety_note":
        "Reduce speed and increase deconfliction under tight CBF
        ↪  margins."
    },
    {
      "label": "progress_oriented",
      "speed_scale": 0.75,
      "obstacle_repulse_scale": 0.8,
      "pair_spread_scale": 0.9,
      "altitude_lane_strength": 0.0,
      "altitude_strategy": "nominal",
      "safety_note":
        "Improve coverage progress when CBF-QP feedback remains mild."
    }
  ],
  "overall_note":
    "Proposing balanced, safety-oriented, and progress-oriented
    ↪  candidates."
}
```

Post-processing and candidate selection. The LLM response is parsed into a finite candidate set

$$\Theta_k = \{\theta_k^{(1)}, \ldots, \theta_k^{(N_c)}\}.$$

The LLM does not directly select the final high-level decision. For each candidate, the predictive evaluator performs a short-horizon rollout using the same low-level CBF-QP as in execution. It computes predicted task cost, CBF margin, CBF intervention, active-constraint ratio, and QP feasibility. The final selected decision is

$$\theta_k^\star \in \Theta_k.$$

This selection is made by the predictive evaluator, not by the LLM alone. The selected candidate then parameterizes the nominal controller, and the resulting command is filtered by the low-level CBF-QP before execution.

This separation is important for the safety role of Predictive H-CBF. The LLM-based harness uses CBF-QP feedback and memory to generate high-level candidates. The predictive evaluator selects among these candidates using rollout-based information. The low-level CBF-QP remains responsible for execution-time safety.

H Experimental Setup and Benchmark Details

This appendix describes the simulation settings, controller variants, metrics, and reproduction commands used in the experiments. All experiments use the same low-level CBF-QP implementation and the same CBF feedback vector unless otherwise stated. The main experiments use the Gemini 2.5 Flash-Lite prompt specification described in Appendix G.

H.1 Common simulation setting

All swarm-control experiments use double-integrator UAV dynamics,

$$\dot{p}_i = v_i, \qquad \dot{v}_i = u_i,$$

where $p_i, v_i, u_i \in \mathbb{R}^3$ are the position, velocity, and acceleration of agent i. The low-level controller computes a nominal acceleration from the current high-level target and then filters it through the CBF-QP. The CBF-QP enforces obstacle avoidance, pairwise separation, boundary safety, speed limits, and componentwise acceleration limits. The CBF feedback vector ϕ_k used by the harness is the same as in the main method,

$$\phi_k = [m_k, r_k, a_k, s_k, q_k],$$

where m_k is the minimum safety margin, r_k is the CBF intervention norm, a_k is the active-constraint ratio, s_k is the CBF relaxation value, and q_k is the hard infeasibility indicator. These feedback terms match the CBF feedback vector defined in Sec. 3.

Table 9: Common simulation and low-level control parameters. The acceleration limit is imposed componentwise. Some difficulty levels override the default speed and acceleration limits, as shown in Table 10.

Parameter	Value
Sampling time T_s	0.055 s
Workspace	$[0, 10] \times [0, 10] \times [0.8, 5.6]$
Default number of UAVs	14
Default speed limit $v_{\max}$	1.30
Default acceleration limit $u_{\max}$	0.28 per component
Drone radius	0.14
Coverage grid	$12 \times 12 \times 7$
Coverage gain	1.25
Velocity tracking gain	2.35
HOCBF gains	$k_0 = 3.0,\ k_1 = 3.7$
Speed CBF gain	$\alpha_{\text{speed}} = 3.0$
Boundary CBF gain	$\alpha_{\text{bdry}} = 2.4$
Default obstacle extra margin	0.30
Default inter-agent minimum distance	0.68
Predictive rollout horizon	3 steps
LLM planner interval	8 control steps

H.2 Coverage benchmark

The main benchmark is a 3-D multi-UAV coverage-control task. The workspace is a rectangular box

$$\mathcal{X} = [0, 10] \times [0, 10] \times [0.8, 5.6].$$

The swarm must cover a spatial density field while satisfying obstacle, pairwise, boundary, speed, and acceleration constraints. The density field is evaluated on a fixed grid of size $12 \times 12 \times 7$. Unless otherwise stated, the density field is fixed during an episode.

Let $\rho(g) \geq 0$ denote the density value at grid point $g \in \mathcal{G}$. For an agent configuration $p_k = \{p_{1,k}, \ldots, p_{N,k}\}$, the coverage cost is computed as a density-weighted distance from grid points to their nearest agent,

$$J_{\text{cov}}(p_k) = \frac{\sum_{g \in \mathcal{G}} \rho(g) \min_i \|g - p_{i,k}\|^2}{\sum_{g \in \mathcal{G}} \rho(g)}.$$

The nominal coverage controller moves each agent toward a density-weighted coverage target. The harness modifies this target through the high-level parameters

$$\theta_k = [\gamma_k, \eta_k^{\text{obs}}, \eta_k^{\text{pair}}, \eta_k^{\text{alt}}],$$

where γ_k scales the desired speed, η_k^{obs} increases obstacle clearance, η_k^{pair} increases inter-agent spreading, and η_k^{alt} encourages altitude-lane separation.

The obstacle set consists of inflated spherical obstacles. For obstacle o, the monitored obstacle safety function is

$$h_{i,o}^{\text{obs}}(p_i) = \|p_i - p_o\|^2 - R_o^2,$$

where p_o is the obstacle center and R_o is the inflated safety radius. The pairwise separation, boundary, and speed safety functions are defined in Appendix A.

H.3 METRICS

We use the following metrics throughout the experiments.

Safety-violation steps. A safety-violation step is counted when the minimum monitored safety gap is negative:

$$N_{\text{viol}} = \sum_k \mathbb{I}\{m_k < 0\}.$$

The minimum safety gap is computed over obstacle, pairwise, boundary, and speed safety margins.

QP-infeasible steps. A QP-infeasible step is counted when no admissible acceleration satisfies all hard CBF or HOCBF inequalities under the velocity and acceleration limits:

$$N_{\text{inf}} = \sum_k q_k.$$

This metric is reported only for methods that solve a CBF-QP.

CBF intervention norm. The intervention norm measures how much the CBF-QP changes the nominal command:

$$r_k = \|u_{\text{safe},k} - u_{\text{nom},k}\|.$$

We report the mean value over the episode.

Active-constraint ratio. The active-constraint ratio is the fraction of monitored CBF constraints that are close to active at the CBF-QP solution. It measures how constrained the low-level CBF-QP is during execution.

Tail-averaged coverage cost. For coverage tasks, we report the average coverage cost over the last 80 simulation steps:

$$J_{\text{tail}} = \frac{1}{80} \sum_{k=T-79}^{T} J_{\text{cov}}(p_k).$$

Normalized task regret. For task-transfer experiments, raw task costs have different scales across task families. We therefore report normalized task regret,

$$\text{Regret} = \frac{J - J_{\text{best}}}{J_{\text{best}}},$$

where J_{best} is the best tail task cost among the compared methods on the same task.

H.4 EXPERIMENT 1 DIFFICULTY LEVELS

Experiment 4.1 evaluates the 3-D coverage benchmark under three difficulty levels. Difficulty varies with the number of agents, obstacle clutter, density concentration, input limits and inter-agent separation. Table 10 lists the corresponding settings.

Table 10: Evaluation configuration. The episode lengths are Easy 260 and Medium 250; the complete settings for each difficulty are listed below.

Setting	Seed	UAVs	Steps	$v_{\max}$	$u_{\max}$	Obstacles	$d_{\min}$	Density peaks
Easy	200	10	260	1.34	0.34	3	0.62	3
Medium	300	14	250	1.30	0.28	4	0.68	4
Hard	400	14	250	1.28	0.26	4	0.70	4

H.5 Controller variants

Table 11 summarizes the controller variants used in the main experiments and appendix reference comparisons. The main CBF-based comparison uses Reactive CBF, Feedback H-CBF, and Predictive H-CBF. Nominal Coverage is included only as a task-only reference in the appendix; it does not solve a CBF-QP and is not part of the main CBF-based comparison in Section 4.1–4.3. Its role is to show the task behavior obtained without execution-time CBF-QP safety correction.

The high-level planner does not output raw control inputs. It only proposes high-level planning parameters or high-level candidates. The final acceleration is always computed by the nominal controller and then filtered by the low-level CBF-QP when a CBF-QP is used.

Table 11: **Controller variants used in the experiments.** Nominal Coverage is an appendix-only task reference without CBF-QP safety correction. The CBF-based variants isolate reactive CBF-QP filtering, CBF feedback, and CBF-aware predictive candidate selection.

Method	CBF-QP	CBF feedback	Candidate selection	Role
Nominal Coverage	No	No	No	Appendix-only task reference without CBF-QP safety correction.
Reactive CBF	Yes	No	No	Applies the low-level CBF-QP to the nominal coverage command.
Feedback H-CBF	Yes	Yes	No	Uses CBF margin feedback to revise one future high-level decision.
Random selector	Yes	Yes	Random	Samples one candidate from the shared high-level candidate pool.
Fixed selector	Yes	Yes	Fixed	Always selects a fixed candidate type from the shared candidate pool.
Progress-only selector	Yes	Yes	Task progress only	Selects the candidate with the best predicted task progress only.
Predictive H-CBF	Yes	Yes	CBF-aware predictive	Selects a candidate using predicted task progress, CBF margin, QP feasibility, intervention, and active constraints.

H.6 Experiment 2 candidate-selection benchmark

Experiment 4.2 isolates the role of candidate selection inside Harness-CBF. All selector variants use the same high-level candidate pool and differ only in the selector used before execution. The shared candidate pool contains aggressive, adaptive-progress, balanced, direct-style, safe, feasibility-guard, deterministic-baseline, and last-valid candidates.

We evaluate Feedback H-CBF, Random selector, Fixed selector, Progress-only selector, and Predictive H-CBF. The experiment is run on the Medium and Hard settings from Table 10. Each setting uses 10 independent groups with 50 seeds per group. The main text reports mean values, and Appendix E provides additional metric bars, distribution-level results, and selector-structure plots.

H.7 EXPERIMENT 3 TASK-TRANSFER BENCHMARK

Experiment 4.3 evaluates whether the same CBF feedback and predictive evaluator transfer across task families. We keep the low-level CBF-QP, CBF feedback vector ϕ_k, high-level candidate representation, and predictive evaluator fixed, and only change the task configuration. The three task families are static coverage, narrow-passage navigation, and dynamic-density coverage.

Table 12: Task settings, including seed ranges, swarm sizes, episode lengths and input limits.

Task	Legacy seed range	UAVs	Steps	$v_{\max}$	$u_{\max}$
Static coverage	23–26	14	240	1.30	0.28
Fixed spatial density field with obstacle avoidance and pairwise separation constraints.					
Narrow passage	41–44	14	255	1.28	0.26
Obstacle-induced bottleneck that pushes many agents into a low-clearance region at the same time.					
Dynamic density	57–60	14	250	1.30	0.27
Time-varying density field whose hotspots move over the episode, creating a moving coverage target.					

I SUPPLEMENTARY EVALUATIONS

The supplementary material expands the main experiments with candidate-generator comparisons, feedback ablations, frozen profiles, fallback analysis, parameter sensitivity, timing, communication and additional behaviors. Supplementary tables and figures use S-prefixed numbers.

J WHAT THE ORIGINAL AGGREGATE COUNTS ALREADY IMPLY

Conditional statistical analysis. The original QP rates imply success totals $329/500$ versus $271/500$ for Hard Predictive versus Fixed, and $389/500$ versus $369/500$ for Medium Predictive versus Feedback.

It enumerates every pairing compatible with the reported marginals, conditional on those marginals and the claimed evaluation protocol.

Let A and B be the two success totals in n common independent paired trials and let h be the number where both succeed. Necessarily,

$$\max(0, A + B - n) \le h \le \min(A, B), \quad b_{10} = A - h, \quad b_{01} = B - h.$$

For each integer h in this range, the exact two-sided McNemar value is

$$p(h) = \min\{1, 2\Pr[\mathrm{Bin}(A + B - 2h, 1/2) \le \min(A - h, B - h)]\}.$$

This is the exact-binomial version of the paired homogeneity test; the implementation follows the definition documented in `https://www.statsmodels.org/stable/generated/statsmodels.stats.contingency_tables.mcnemar.html`. Taking the largest value over all possible h gives a conservative all-pairing bound. A factor of four supplies a Bonferroni bound for the four prespecified comparisons below, not a Holm value inferred without pairings.

Table S1: All-pairing upper bounds from the QP-success marginals, conditional on the stated evaluation and independence assumptions.

Setting/comparator	Predictive	Comparator	Δ pp	$p_{\max}$	$4p_{\max} \wedge 1$
Hard/Fixed	329/500	271/500	11.6	0.0043	0.0173
Hard/Feedback	329/500	110/500	43.8	1.98×10^{-26}	7.93×10^{-26}
Medium/Fixed	389/500	77/500	62.4	3.39×10^{-51}	1.36×10^{-50}
Medium/Feedback	389/500	369/500	4.0	0.2219	0.8875

A difference interval that does not require the unknown overlap can also be formed. Obtain exact Clopper–Pearson intervals $[L_A, U_A]$ and $[L_B, U_B]$, each at level 97.5%, and use $[L_A - U_B, U_A - L_B]$. The union bound gives at least 95% simultaneous coverage provided each arm consists of IID binary trials. Independence between the two arms is not required. This conservative interval is not a reconstructed paired-bootstrap interval.

Table S2: Admissible success-overlap range and a conservative 95% difference interval (percentage points), obtained from two 97.5% Clopper-Pearson intervals.

Comparison	Overlap count range		Difference interval	
Hard/Fixed	100	–271	1.59	, 21.43
Hard/Feedback	0	–110	34.39	, 52.55
Medium/Fixed	0	–77	53.96	, 69.89
Medium/Feedback	258	–369	-4.78	, 12.72

What this does and does not answer. Under the stated paired/IID protocol, Hard Predictive-Fixed remains significant for every admissible overlap, even with the four-comparison correction. Medium Predictive-Feedback does not: its decision can depend on the unknown pairing. These deductions are conditional on the stated paired/IID assumptions. Correlation across episodes or an undocumented grouping scheme would require an appropriate cluster analysis.

K ADDITIONAL GENERATOR AND FEEDBACK COMPARISONS

The following comparisons distinguish the contributions of feedback, selection and candidate generation.

The comparison uses 500 matched environment IDs per setting and ten descriptive groups of 50. This cohort is analyzed separately from the main experiment.

Equal evaluation budget. All generator arms use the original four-dimensional parameter interface, the same soft evaluator, $H = 3$ and a total of 8 evaluated candidates. Two common anchors count within that budget, leaving 6 generator-specific proposals. CEM has 2 adaptive rounds with 3 fresh candidates each; its final executed choice must be evaluated, not an untested final distribution mean. Warm-start shooting search receives the same rollout budget. This is a parameter-search baseline, not a claimed implementation of full joint-input MPC. Continuous-action MPC may still be a useful separate comparison. Grid means a budgeted fixed design in the same box, not an impossibly exhaustive four-dimensional grid within eight points.

Table S3: Medium generator comparison. Every method uses the same eight-candidate, three-step evaluation budget.

Generator	QP count	Safe %	Inactive %	Raw cost	Speed
LLM	395/500	91.60	56.68	4.237	0.790
Uniform	344/500	80.60	51.06	4.589	0.798
Grid	337/500	83.00	52.33	4.581	0.801
Sobol	362/500	83.60	53.90	4.349	0.801
Rule	364/500	87.20	53.75	4.612	0.703
MLP	368/500	87.00	55.63	4.336	0.805
CEM	385/500	88.80	56.63	4.208	0.805
Adaptive-search	376/500	87.40	55.96	4.209	0.799

Table S4: Hard generator comparison. Every method uses the same eight-candidate, three-step evaluation budget.

Generator	QP count	Safe %	Inactive %	Raw cost	Speed
LLM	345/500	78.40	48.03	5.255	0.716
Uniform	281/500	69.00	42.31	5.711	0.725
Grid	292/500	72.20	43.46	5.630	0.727
Sobol	293/500	71.20	45.57	5.419	0.736
Rule	292/500	75.80	44.58	5.707	0.628
MLP	324/500	73.20	46.95	5.374	0.738
CEM	333/500	77.00	47.52	5.199	0.733
Adaptive-search	317/500	75.20	46.91	5.299	0.727

Small-network comparison. The MLP maps 18 fixed numerical context features to 6 four-dimensional proposals, with hidden layers 64, 64 and 6936 trainable parameters. A sigmoid and the same box bounds produce valid proposals. The training and validation pools contain 10000 and 2000 distinct context snapshots. Data collection, training and online search have separate computational costs. Equal online evaluation counts do not imply equal total cost.

Keep feedback and prediction separate. The complementary ablations mask feedback, remove history, or supply only a scalar summary; the candidate interface and evaluator remain fixed. The original direct-feedback baseline is not redefined as a one-candidate predictive algorithm: one-candidate rollout checking and direct release are different procedures. These extra arms supplement rather than replace the original Reactive–Feedback–Predictive progression.

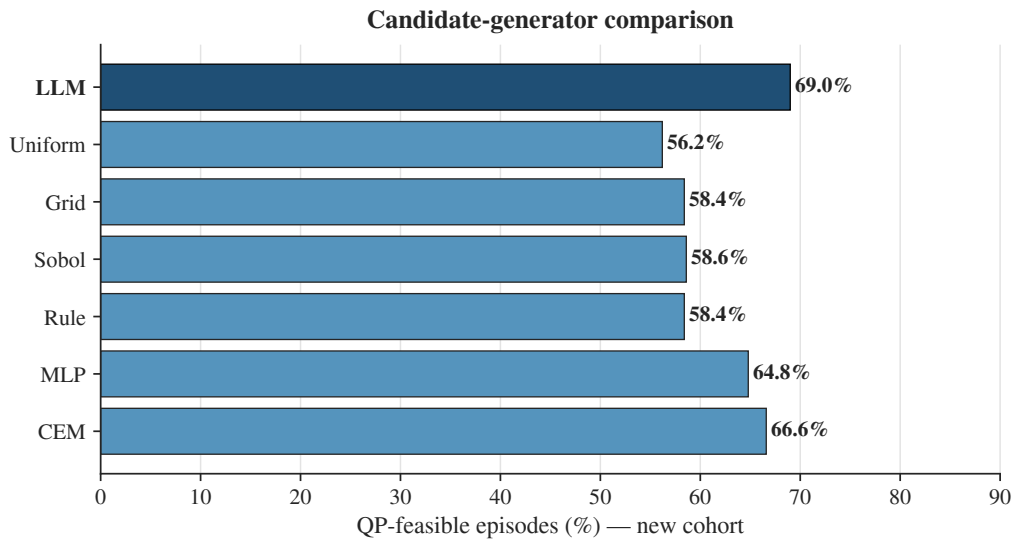

Figure S1: Supplementary generator comparison using the Hard rows of Table S4. This comparison complements the selector ablation in Figure 4.

Table S5: Complementary mechanism ablations. Standard deviations are across ten descriptive 50-episode groups, not standard errors.

Arm	QP %	Group SD	Raw cost	Stop %
Medium/LLM	79.00	4.55	4.237	11.10
Medium/No-feedback	69.40	4.01	4.352	9.99
Medium/No-memory	74.80	8.55	4.368	10.48
Medium/Scalar-feedback	72.40	7.76	4.343	13.56
Hard/LLM	69.00	5.19	5.255	13.86
Hard/No-feedback	53.80	5.45	5.358	13.42
Hard/No-memory	62.40	4.60	5.314	13.30
Hard/Scalar-feedback	59.80	9.86	5.353	16.97

Paired inference. Paired bootstrap resampling uses common episode indices and 3000 resamples per contrast. Exact McNemar tests use discordances, and Holm correction includes 20 prespecified contrasts. Confidence intervals are unadjusted descriptive 95% intervals; the adjusted tests address the stated family. No post-hoc seed search was used to make a difference significant.

Table S6: Paired LLM-minus-comparator differences in the generator cohort. The discordance column is b10/b01 (not a rate). Holm correction covers all 20 tests in the CSV, not only the eight displayed.

Comparison	Δ pp	95% paired CI	Discordance	Holm p
Medium/Uniform	10.2	5.2 , 15.0	109/58	0.0017
Medium/MLP	5.4	0.6 , 10.0	84/57	0.1974
Medium/CEM	2.0	-2.4 , 6.4	68/58	0.8195
Medium/No-feedback	9.6	4.8 , 14.6	106/58	0.0029
Hard/Uniform	12.8	7.6 , 18.0	121/57	3.48×10^{-5}
Hard/MLP	4.2	-0.8 , 9.4	94/73	0.4858
Hard/CEM	2.4	-2.8 , 7.4	95/83	0.8195
Hard/No-feedback	15.2	9.8 , 20.4	137/61	1.37×10^{-6}

L FROZEN PROFILES, CANDIDATE QUALITY, AND FALLBACK BEHAVIOR

Frozen profiles and fallback branches. The three profile vectors (progress, balanced and crowding) are defined below.

$$(.85, .30, .30, .10) \quad , \quad (.65, .70, .70, .35) \quad , \quad (.45, 1.00, 1.30, .70) \quad .$$

The ordering is $(\gamma, \eta^{\text{obs}}, \eta^{\text{pair}}, \eta^{\text{alt}})$.

Task cost, stop time and inactive rate jointly distinguish useful fixed-profile behavior from mere slowing or stopping.

Table S7: Frozen-profile sensitivity analysis in a separate evaluation cohort.

New frozen arm	QP count	Inactive %	Raw cost	Stop %
Medium/progress	160/500	35.12	3.550	3.95
Medium/balanced	302/500	42.56	4.656	17.95
Medium/crowding	228/500	47.33	4.854	29.85
Hard/progress	113/500	34.32	3.982	5.12
Hard/balanced	257/500	39.34	5.111	18.87
Hard/crowding	294/500	44.20	5.481	33.39

Counterfactual fallback audit. The branch data contain 1200 context snapshots and 8 candidate branches per context, including the baseline. The original soft admission-and-selection rule is applied to the generated scores with a threshold 0.60. Every branch starts with the same current hard-QP feasibility flag; the nominal candidate cannot change the current feasible set. The empty/nonempty denominator is counted from the actual generated score set, not sampled independently.

The candidate-quality question is empirical: record whether any candidate is both future-hard-feasible and task-improving relative to the matched baseline, then report how often the selected

Table S8: Branch-matched fallback diagnostic. Current QP feasibility is shared by all branches of a snapshot. Worse-task counts use the baseline branch from the same snapshot; ties are not worse.

Score set	Events	Current QP	Future QP	Task worse	ΔJ
Nonempty	1103	1073/1103	927/1103	279/1103	-0.407
Empty	97	94/97	72/97	47/97	0.028

Table S9: Candidate availability versus selection. Improving means lower raw task cost than the baseline at the same context and a future-QP-feasible branch. Strict ties are not improvements. These statistics do not imply a general existence theorem.

Score set	Contexts	Any feasible	Any improving	Selected improving
Nonempty	1103	1103/1103	921/1103	582/1103
Empty	97	97/97	79/97	33/97

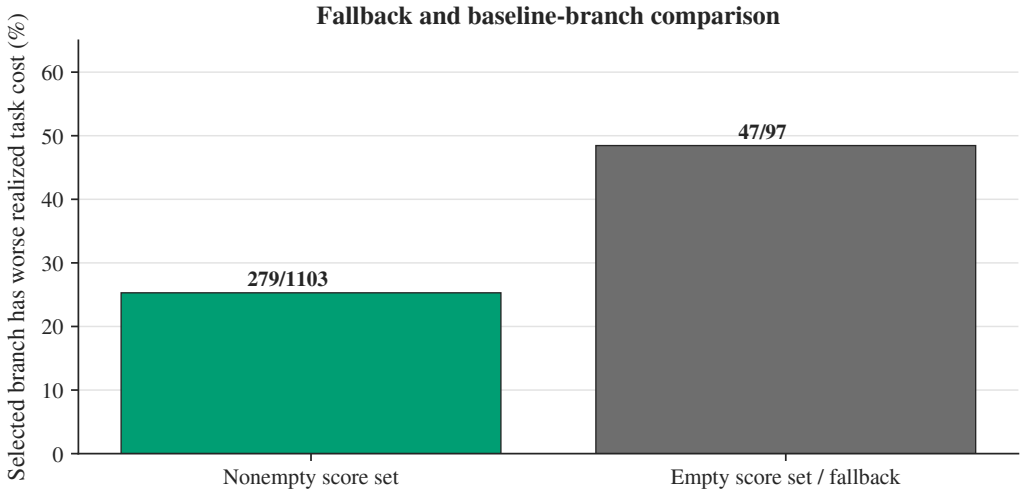

Figure S2: counterfactual illustration: minimizing predicted burden in the empty-set branch need not preserve task performance. Each rate uses its own branch-condition denominator. No fallback safety or recursive feasibility claim follows.

candidate achieves that. Including an anchor preserves a comparison option, not a safe backup or an unconditional existence theorem.

M LANGUAGE VARIATION WITHOUT CHANGING THE NOMINAL-CONTROL INTERFACE

Finite-schema instruction test. Only instructions expressible through the existing four-parameter interface are included. Examples are "prefer more obstacle clearance while keeping normal progress", its paraphrase, or "when pair pressure rises, favor altitude separation unless boundary clearance is small". Reserving an arbitrary corridor or navigating to a new semantic room requires capabilities outside this interface and is not included.

The test contains template, held-out paraphrase and held-out conjunction splits. Ground-truth intent labels define the requested finite preference; agreement is checked independently of the LLM's prose rationale. The oracle-input MLP receives structured intent directly, so it is an oracle control rather than an input-matched competitor.

Table S10: Finite-schema instruction sensitivity, 500 cases per cell. MLP-oracle receives ground-truth structured intent, whereas the LLM receives text; this is an oracle control, not an input-matched superiority test.

Instruction/arm	Agreement %	QP %	Cost
Template/Rule	90.60	71.60	4.584
Template/MLP-oracle	90.00	68.80	4.574
Template/LLM	92.80	71.60	4.538
Paraphrase/Rule	60.00	72.00	4.712
Paraphrase/MLP-oracle	92.80	67.60	4.696
Paraphrase/LLM	89.60	71.20	4.699
Composition/Rule	54.40	74.20	4.682
Composition/MLP-oracle	82.60	74.40	4.631
Composition/LLM	83.20	74.80	4.625

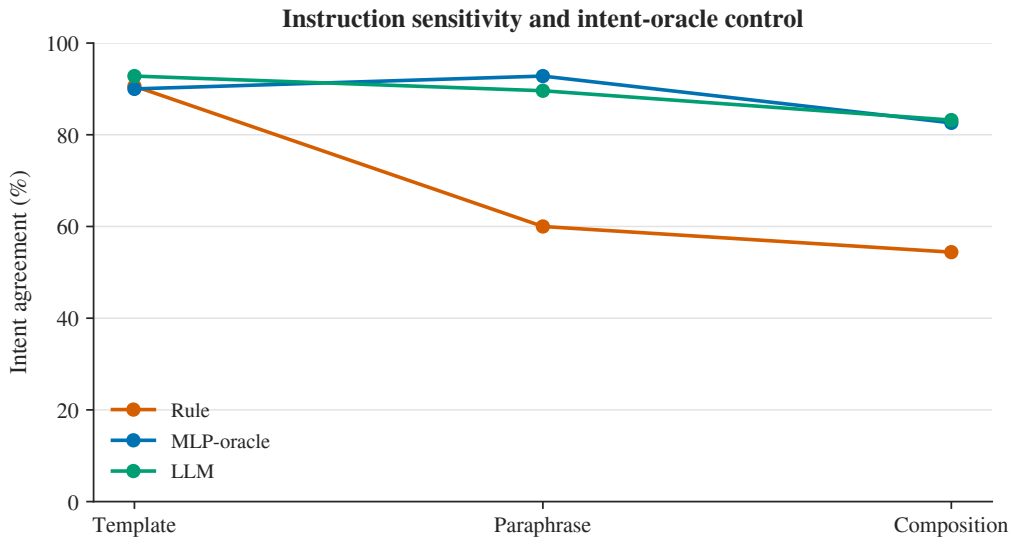

Figure S3: Instruction-variation results. Ground-truth intent is provided to MLP-oracle; its input advantage is considered separately from text-input performance.

N TRAJECTORY AND FEASIBILITY DIAGNOSTICS

Trajectory diagnostics.
This check evaluates positions, velocities, nominal/executed inputs, all coupled CBF rows and joint LP witnesses. The model uses 14 double-integrator agents, 2 static spheres, all 91 pairwise constraints, box boundaries, and speed constraints. There are 217 barrier rows and 42 control variables. The model uses $T_s = 0.055$, $u_{\max} = 0.28$, $v_{\max} = 1.28$, $\lambda = 3.7$ and $\beta = 3.0$.

Actual numerical operations. For $r(z, u) = C(z)u + d(z)$, the relaxed QP can be solved after eliminating slack:

$$\min_{u \in \mathcal{U}} \tfrac{1}{2}\|u - u_{\text{nom}}\|^2 + c_\delta \|[-C(z)u - d(z)]_+\|^2.$$

This is the same finite quadratic-slack objective, with $c_\delta = 100$ in the implementation. L-BFGS-B solves the box-constrained convex objective; termination and projected-gradient residuals are logged, not hidden. Separate HiGHS LP calls calculate the nominal joint reserve and a robust witness after subtracting the Appendix B.3 bounds. All witnesses are projected into the input box and residuals are recomputed. These float64 computations are *numerical diagnostics*; no interval or outward-rounded machine certificate is claimed.

Trajectory contract and separate test regimes. All candidates start from the same state within a scene; the disturbance sequence is shared across candidates in that scene. Prediction and execution use exact zero-order-hold integration. The clone regime has identical transitions and no disturbance. The bounded regime adds componentwise acceleration noise in $[-.015$ $, .015$ $]$ with the same declared budget. The stress regime uses $[-.20$ $, .20$ $]$ while retaining the smaller budget: it deliberately lies outside that disturbance contract, though an observed trajectory may still fit the conservative tube. There are 12 , 16 , and 16 predetermined scenes, respectively, with 8 fixed nominal profiles per scene. An additional, separately identified interior positive control has 8 well-separated, slow-moving scenes. It does not replace any of the challenging scenes or alter their unfavorable results.

Table S11: Numerical feasibility diagnostics. Passes use the conservative input-box tube, not the soft-score guard. A zero passing denominator is shown as a dash, never as a zero false-pass rate. Float64 LP witnesses are not formal interval certificates.

Preset/H	Passing candidates	Available scenes	Tube coverage	False/pass
exact/3	0/96	0/12	96/96	–
exact/8	0/96	0/12	96/96	–
bounded/3	0/128	0/16	128/128	–
bounded/8	0/128	0/16	128/128	–
stress/3	0/128	0/16	128/128	–
stress/8	0/128	0/16	128/128	–
interior/3	64/64	8/8	64/64	0/64
interior/8	0/64	0/8	64/64	–

The input-box tube passes no candidate in the challenging presets. The interior positive control passes its three-step candidates but not its eight-step candidates. A zero false-pass count with no passing candidates has an undefined rate. Certificate availability and validity are therefore reported together.

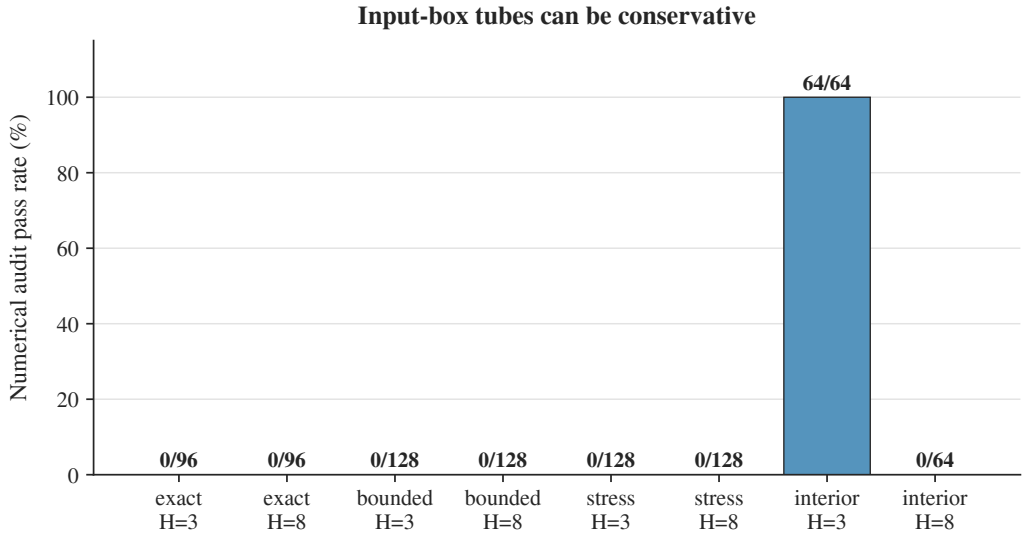

Figure S4: Certificate availability across all presets. Longer input-box tubes can be conservative even when the evaluated trajectory remains feasible.

Disjoint calibration and candidate-selection bias. Only the first 8 bounded scenes set the empirical 95% error quantile; the remaining 8 are the bounded test set. Stress scenes are separate. In this evaluation the soft score is the documented burden, and the original soft selection pattern is used with the task cost. All-candidate and selected-candidate errors are reported separately. The empirical quantile is not a uniform theorem premise; eight calibration scenes are a small diagnostic sample, not deployment validation.

Table S12: measured score errors from paired predicted/executed trajectories, H=3. The bound is the higher 95% quantile of 64 candidates from eight disjoint bounded-noise calibration scenes; candidate observations within scenes are dependent.

Split/population	n	Median	95th	Maximum	$\lvert e\rvert > \epsilon$
bounded/all	64	7.38×10^{-5}	2.33×10^{-4}	2.46×10^{-4}	0/64
bounded/selected	8	4.44×10^{-5}	1.35×10^{-4}	1.35×10^{-4}	0/8
stress/all	128	0.0015	0.0060	0.0095	111/128
stress/selected	16	0.0015	0.0055	0.0055	15/16

Table S13: error exceedance and realized threshold failures. In this evaluation, tau=0.6 and barP=tau+epsilon95(calibration).

Split/population	$\epsilon_{.95}$	$P > \bar{P} \mid \lvert e\rvert > \epsilon$	False/soft-admitted
bounded/all	4.14×10^{-4}	–	0/16
bounded/selected	4.14×10^{-4}	–	0/2
stress/all	4.14×10^{-4}	95/111	0/16
stress/selected	4.14×10^{-4}	13/15	0/2

Existence of a witness versus the actually applied QP input. The numerical trace records include cases with a feasible sampled prefix through step three but negative reserve at a later step four–eight. No three-step statement is extrapolated through the entire eight-step hold. A positive LP witness concerns existence of an input, not the particular relaxed-QP input applied. The stored fields explicitly

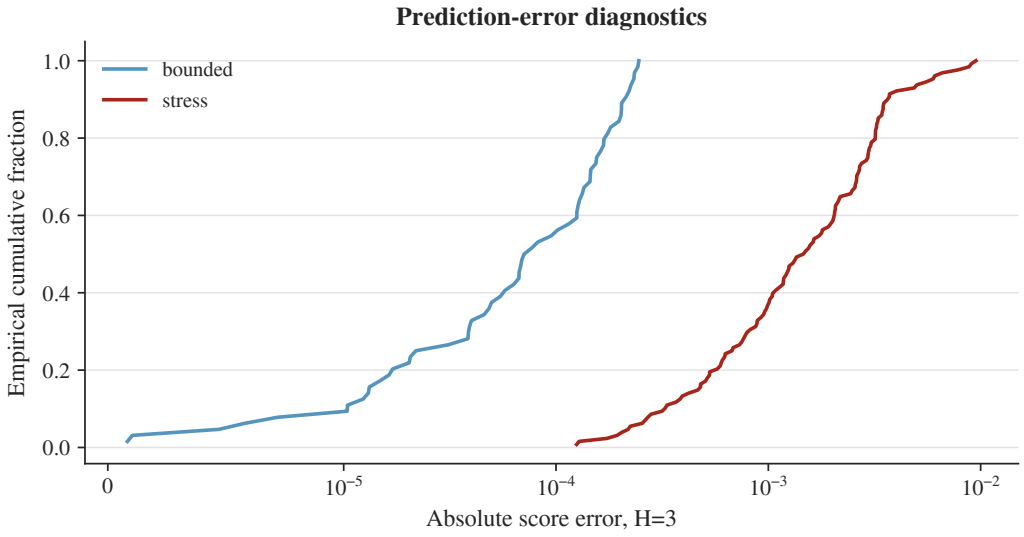

Figure S5: H=3 score-error distribution calculated from stored state/input trajectories and row evaluations. A symmetric-log horizontal axis preserves both the small-error region and the large errors.

Table S14: actual QP optimizer termination and executed hard-row checks. Feasibility of an alternative LP witness does not imply the relaxed QP output has zero slack. Suffix counts have a hard-feasible sampled prefix through step 3 and a negative reserve at a later step 4-8.

Preset/H	QP convergence	Applied hard rows	Prefix OK/suffix bad
exact/3	96/96	0/96	2
exact/8	96/96	0/96	–
bounded/3	128/128	0/128	7
bounded/8	128/128	0/128	–
stress/3	128/128	0/128	0
stress/8	128/128	0/128	–
interior/3	64/64	64/64	0
interior/8	64/64	64/64	–

distinguish these quantities. These residual checks concern actuator-input feasibility in the modeled dynamics at disturbed states; they do not prove robustness of the executed continuous-time barrier derivative against the additional acceleration or intersample safety.

Auditable artifacts. `REFERENCE_state_input_witness.npz` contains predicted and executed p, v, u, every numerical witness, and the per-agent error radii. `REFERENCE_step_audit.csv` records all sampled reserves, residual checks and QP termination indicators. Candidate-level records connect those arrays to their scene, candidate and horizon. The positive control is in its own directory. Zero clone error is expected by identical deterministic execution; it is not an unexplained decimal coincidence.

O SENSITIVITY TO HORIZON, CANDIDATE BUDGET AND SCORE PARAMETERS

Parameter sensitivity. The following tables change one variable at a time; wMargin and wQP scale the indicated score weight, while EMA changes the feedback memory coefficient. Baseline $H = 3, N_c = 8$ values refer to the new LLM generator rows; this deliberate reference is not an independent repeated experiment. Sweep rows use an independent cohort , so no paired significance claim against that default is made. Larger H and N_c incur more evaluator work: $N_c H$ transition/QP updates plus any separately charged terminal-state audit. A fixed-candidate-count sweep and a fixed-total-QP-budget sweep answer different questions and must not be conflated.

Table S15: Medium parameter sensitivity. The default H=3/Nc=8 values reuse the LLM generator row and are not an independent replication. The other conditions form a separate cohort and are not paired against the default.

Setting	H	Nc	QP %	Group SD	Cost
H1	1	8	78.20	5.12	4.383
H5	5	8	78.00	4.52	4.205
H8	8	8	79.20	6.81	4.272
N4	3	4	71.80	6.63	4.337
N16	3	16	80.80	4.24	4.193
lambda0.5	3	8	72.40	8.21	4.092
lambda2	3	8	77.60	2.63	4.472
tau0.3	3	8	75.60	6.38	4.463
tau0.9	3	8	74.00	9.29	4.246
wMargin0.5	3	8	75.00	3.02	4.240
wMargin2	3	8	76.00	4.11	4.422
wQP0.5	3	8	73.80	5.77	4.206
wQP2	3	8	77.60	3.24	4.369
EMA0.5	3	8	74.80	6.41	4.225
EMA0.95	3	8	75.00	5.60	4.355

Table S16: Hard parameter sensitivity. The default H=3/Nc=8 values reuse the LLM generator row and are not an independent replication. The other conditions form a separate cohort and are not paired against the default.

Setting	H	Nc	QP %	Group SD	Cost
H1	1	8	61.40	6.54	5.359
H5	5	8	66.60	8.54	5.248
H8	8	8	70.40	4.88	5.226
N4	3	4	58.60	7.55	5.413
N16	3	16	67.60	8.88	5.233
lambda0.5	3	8	61.20	6.41	5.143
lambda2	3	8	66.00	7.36	5.560
tau0.3	3	8	66.00	8.00	5.531
tau0.9	3	8	66.00	5.81	5.169
wMargin0.5	3	8	64.80	7.90	5.283
wMargin2	3	8	66.00	5.16	5.449
wQP0.5	3	8	63.60	6.85	5.215
wQP2	3	8	65.80	5.77	5.458
EMA0.5	3	8	67.80	8.87	5.238
EMA0.95	3	8	64.80	6.55	5.464

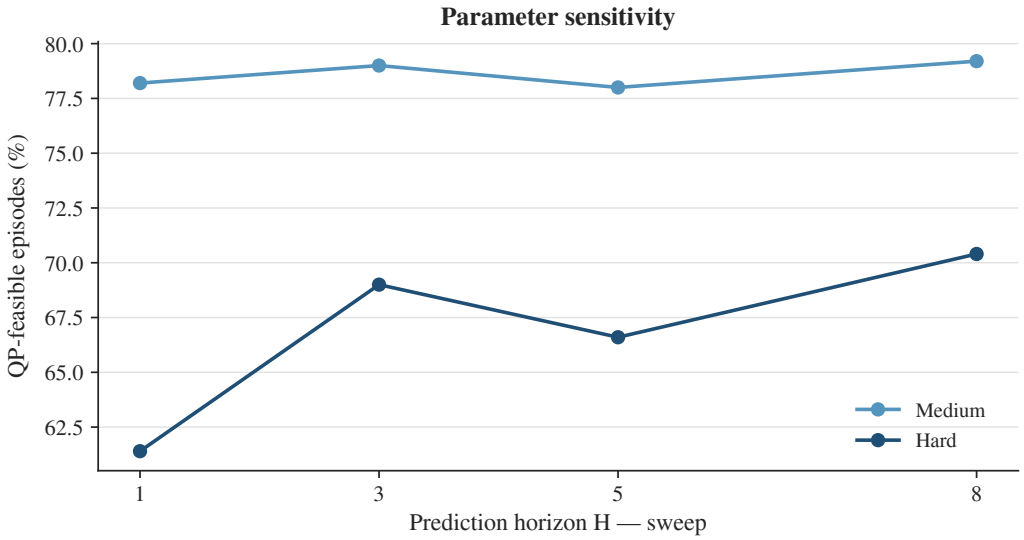

Figure S6: Horizon sensitivity. The default reuses the generator-cohort result; other points represent independently evaluated conditions.

P RUNTIME, COST AND CENTRALIZED COMMUNICATION

Timing and cost accounting. The records contain 1000 planning events, including load fluctuations, retries, token counts and per-event component times. The total mean is the sum of component means; medians and percentiles are computed from event totals rather than added component percentiles.

Table S17: Wall-clock time distributions in milliseconds. End-to-end values are summed per event before summarization. The separate feasibility audit is accounted for separately from the selector.

Component	Mean ms	Median ms	95th ms
LLM response	837.16	760.75	1585.45
One QP	1.87	1.78	2.92
24 QP + rollout	57.05	54.85	83.06
Parse	1.83	1.81	2.47
End-to-end	896.04	819.25	1651.73
Separate audit	44.07	41.72	70.15
Total + audit	940.12	862.89	1697.94

Table S18: Token and cost accounting. Rates are USD 0.4/1.6 per million input/output tokens. Retry charges are included event by event.

Quantity	Mean	95th
api input tokens	1982.3	2538.0
api output tokens	896.5	1128.0
cost USD	0.00232	0.00279

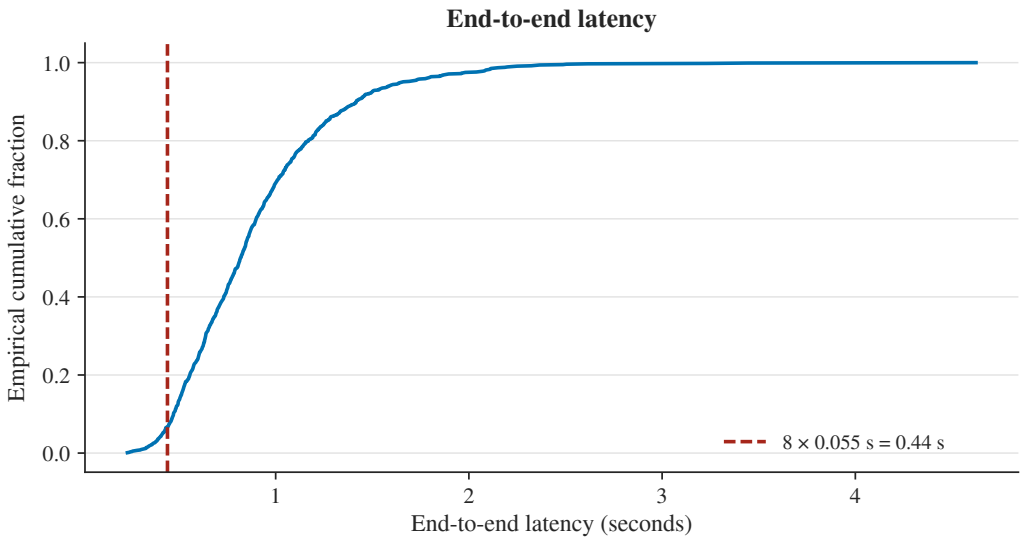

Figure S7: End-to-end latency against the source eight-step interval $8 \times 0.055 = 0.44$ seconds. an API that misses this interval needs a separately specified asynchronous execution and revalidation protocol.

Explicit centralized scope. For payload accounting, 14 agents each send 6 float32 state values and a 16 -byte per-agent header; each receives 3 float32 inputs plus the same header. This gives 560 uplink and 392 downlink bytes per control cycle, or 17.31 kB/s at $T_s = 0.055$ (decimal kB). Map data, retransmission, encryption and radio protocol overhead are excluded.

Table S19: Latency and packet-loss sensitivity for a centralized 14-agent setup.

Delay/loss setting	QP %	Safe %	Cost
0ms	63.00	78.20	5.319
20ms	60.60	78.60	5.375
55ms	60.00	74.00	5.420
110ms	54.40	68.80	5.560
loss1pct	57.40	73.40	5.480
loss5pct	51.00	68.40	5.464

Q ADDITIONAL BEHAVIORS AND OPERATIONAL CONFIGURATION

The additional behavior comparisons consider alignment-based flocking and formation reconfiguration alongside static coverage, dynamic-density coverage and narrow-passage navigation. Task costs are behavior-specific and are not compared across different physical objectives.

Table S20: Additional behavior comparisons. Cost is behavior-specific and must not be compared directly across Flocking and Formation.

Task/arm	QP %	Safe %	Cost	Inactive %
Flocking/Reactive	36.60	48.00	4.528	43.15
Flocking/Feedback	63.80	77.00	4.207	49.15
Flocking/Predictive	83.40	93.40	4.035	51.26
Flocking/CEM	74.40	90.60	4.091	51.44
Formation/Reactive	36.40	46.40	4.514	43.29
Formation/Feedback	68.40	77.60	4.184	48.43
Formation/Predictive	82.00	91.40	4.077	52.11
Formation/CEM	79.80	92.80	4.050	51.14

Configuration. The candidate box is $[.15, 1.0] \times [0, 1.6]^2 \times [0, 1]$. The settings include EMA $\lambda = .8$, feedback weights $(.5, .2, .1, .5, 1.0)$, evaluator weights $(.6, 1.0, .25, .1)$, score multiplier 1.0, threshold .6, and quadratic slack penalty 100. CEM uses 2 elites, smoothing .5 and covariance floor .01. The API configuration uses temperature .2, top-p 1.0, maximum output tokens 1200, timeout 5 seconds, and 0 automatic parse retries; invalid outputs are rejected and any deterministic filler uses remaining budget slots. Numerical solver tolerances are 10^{-6}, not justified one-sided arithmetic error bounds.

R INTEGER, DEPENDENCE AND LOW-PROBABILITY-EVENT CHECKS

Statistical consistency checks. Every supplementary binary rate is derived from an integer count. For the 500-episode cells the increment is 0.2 percentage points. Within an episode the recorded identities are

$$F_{\mathrm{QP}} = \mathbf{1}\{N_{\mathrm{inf}} = 0\}, \quad F_{\mathrm{safe}} = \mathbf{1}\{N_{\mathrm{viol}} = 0\}, \quad F_{\mathrm{joint}} = F_{\mathrm{QP}} F_{\mathrm{safe}}, \quad N_{\mathrm{active}} + N_{\mathrm{inactive}} = T.$$

Paired resampling preserves environment IDs, and conditional fallback rates use the number of contexts satisfying that condition. Candidates sharing a scene are not counted as independent trials. Quantiles use stored records.

Statistical checks. Two-sided binomial tests assess QP, safety and joint-success counts under their respective null hypotheses. Conditional on each total, 1000000 multivariate-hypergeometric allocations into ten groups of fifty define the reference distribution of group-count variance. Both tails use a finite-Monte-Carlo correction. A Bonferroni screen across 486 prespecified checks uses a family threshold of 0.01. The reported flags do not trigger reseeding or rejection.

Table S21: Probability diagnostics. Exact tests are retained. Conditional partitions use 1,000,000 draws; replay uses 100,000. Tails are inclusive and doubled, with a plus-one adjustment. All 486 tests belong to the 0.01 Bonferroni family. "Unresolved" uses simultaneous Monte Carlo tail intervals.

Test type	Tests	MC draws	Min. raw p	Flags	Unresolved
exact count	243	exact	0.0030703	0	0
binary dispersion	243	1000000	0.005644	0	0

Coincident rounded counts are retained. The Medium Sobol/scalar-feedback QP counts coincide, as do the Hard grid/rule counts. Repeated decimals alone do not determine the statistical interpretation.

Summary-statistic sensitivity. This calculation conditions on the success counts and randomly allocates them to ten groups of fifty, 100000 times. It measures how small the reported group standard deviation is under an IID random-grouping model, allowing the rounding interval. Some reported deviations are unusually small under this model. The interpretation depends on the grouping protocol; balanced stratification can invalidate the IID reference distribution.

Table S22: Sensitivity analysis of reported summary statistics. Conditional lower-tail probabilities assume random grouping of IID binary episodes; the analysis does not apply to balanced or stratified grouping without the corresponding adjustment.

Source metric	Reported group SD	IID median SD	Lower-tail MC p
Hard/QP	3.33	6.49	0.0139
Hard/Safe	1.65	5.58	3.10×10^{-4}
Medium/QP	3.05	5.69	0.0197
Medium/Safe	0.84	4.20	3.00×10^{-5}

S SUMMARY OF SUPPLEMENTARY ANALYSES

The supplementary analyses cover prediction errors, candidate generation, fixed profiles, fallback branches, parameter sensitivity, timing, communication and behavior transfer. Statistical conclusions use the population, denominator and comparison family specified for each table.

T CONSISTENCY AND METRIC-ACCOUNTING ANALYSIS

This section examines joint-event counts, activity-rate denominators, grouped uncertainty and task-transfer metric definitions. The supplementary analyses use their stated populations and statistical units.

T.1 DETERMINISTIC ACCOUNTING, BEFORE PROBABILITY TESTS

Let $Q, S \in \{0, 1\}$ denote hard-QP-feasible and violation-free episodes and let $C = QS$. For n equally weighted episodes,

$$\max\{0, N_Q + N_S - n\} \leq N_C \leq \min\{N_Q, N_S\}, \qquad N_S = n \Longrightarrow N_C = N_Q.$$

These are set identities, not confidence bounds or distributional assumptions. For $n = 500$, any binary percentage is $0.2K$ with integer K; two-decimal rounding cannot turn it into 23.10%. The preserved Table 4 has seven binary cells outside this grid and two rows violating the intersection bound.

Table S23: DERIVED source checks for retained Table 4. Rates are copied, not changed. A joint event must lie inside the displayed interval; the two failing rows remain impossible after two-decimal rounding uncertainty. Grid violations additionally appear in the machine-readable audit.

Setting / method	QP %	Safe %	Joint %	Necessary joint range (%)	Status
Hard Reactive	1.4	1.2	1.0	[0.00, 1.20]	In range
Hard Feedback	27.8	41.2	21.4	[0.00, 27.80]	In range
Hard Predictive	71.6	84.2	71.2	[55.80, 71.60]	In range
Medium Reactive	1.6	15.8	1.2	[0.00, 1.60]	In range
Medium Feedback	80.4	68.3	57.2	[48.70, 68.30]	In range
Medium Predictive	84.7	95.4	84.5	[80.10, 84.70]	In range
Easy Reactive	23.1	84.4	22.3	[7.50, 23.10]	In range
Easy Feedback	95.1	98.6	91.2	[93.70, 95.10]	FAIL
Easy Predictive	99.2	100.0	99.1	[99.20, 99.20]	FAIL

Changing the denominator alone is insufficient. A denominator of 1000 could produce a 0.1-percentage-point increment, but it does not repair the Easy/Feedback intersection: $95.10 + 98.60 - 100 = 93.70$, which exceeds 91.20. Nor does it repair Safe $= 100$ with QP $= 99.20$ and intersection $= 99.10$.
If the underlying populations differ, that fact must be supported by records and the metric labels must be revised.

Group standard deviations also have an integer constraint. With ten groups of fifty, let c_g be the integer success count in group g and let $K = \sum_g c_g$. The sample group SD obeys

$$s^2 = \frac{4}{9}\left(\sum_{g=1}^{10} c_g^2 - \frac{K^2}{10}\right), \qquad \sum_g c_g^2 \equiv K \pmod 2.$$

For a population SD, replace 9 by 10. The audit considers both conventions and the full rounding interval. For example, 1.40% implies seven successes; even the most even grouping has sample SD 0.9661 and population SD 0.9165, so the reported 0.80 is unattainable under either stated equal-group convention. The full necessary-condition check is supplied as a CSV.

Denominator diagnostic. For a fixed episode length T, the source definition requires $S_{\text{inactive}} = 100(1 - \overline{N}_{\text{active}}/T)$, including its group-SD counterpart. Exhaustively checking integer T values from 1 to 2000 while allowing two-decimal rounding produces a common compatible value of 260 for all three Easy rows, and 250 for all three Medium rows. Hard is compatible with 250 .

Table S24: Denominator diagnostics for Tables 3, 5 and 10. The final column gives an integer length compatible with the displayed rounding intervals.

Setting / method	Active mean	Inactive source %	Stated T	Implied % at stated T	Compatible integer T
Easy Reactive	125.15	51.86	200	37.42	[260]
Easy Feedback	132.36	49.09	200	33.82	[260]
Easy Predictive	90.86	65.05	200	54.57	[260]
Medium Reactive	153.69	38.52	260	40.89	[250]
Medium Feedback	144.96	42.02	260	44.25	[250]
Medium Predictive	106.97	57.21	260	58.86	[250]
Hard Reactive	156.97	37.21	250	37.21	[250]
Hard Feedback	146.6	41.36	250	41.36	[250]
Hard Predictive	131.32	47.47	250	47.47	[250]

T.2 New backend-scale examples from complete joint event records

The records distinguish the four mutually exclusive outcomes $(Q, S) = (1, 1)$, $(1, 0)$, $(0, 1)$, and $(0, 0)$; QP , safe and joint rates are then computed from these same records, never independently entered.

Table S25: Joint-event accounting. The four mutually exclusive outcomes determine all marginals and intersections. Each row sums to 500.

Setting / method	n_{11}	n_{10}	n_{01}	n_{00}	n
Easy Reactive	106	5	317	72	500
Easy Feedback	471	6	20	3	500
Easy Predictive	496	0	4	0	500
Medium Reactive	3	4	67	426	500
Medium Feedback	284	115	62	39	500
Medium Predictive	411	8	61	20	500
Hard Reactive	1	4	1	494	500
Hard Feedback	111	31	98	260	500
Hard Predictive	360	4	69	67	500

Table S26: backend-scale outcomes from the same records as Table S25. Percentages are means; parentheses give sample standard deviations across ten groups of fifty. QP/safety intersection is an observed-event conjunction, not the independent theoretical audit certificate.

Setting / method	QP-success % (SD)	Safe episodes % (SD)	Intersection % (SD)
Easy Reactive	22.20 (6.07)	84.60 (3.27)	21.20 (6.05)
Easy Feedback	95.40 (2.50)	98.20 (1.99)	94.20 (3.05)
Easy Predictive	99.20 (1.40)	100.00 (0.00)	99.20 (1.40)
Medium Reactive	1.40 (1.35)	14.00 (4.32)	0.60 (0.97)
Medium Feedback	79.80 (6.21)	69.20 (6.20)	56.80 (7.00)
Medium Predictive	83.80 (6.00)	94.40 (2.27)	82.20 (5.61)
Hard Reactive	1.00 (1.41)	0.40 (0.84)	0.20 (0.63)
Hard Feedback	28.40 (6.79)	41.80 (7.91)	22.20 (6.56)
Hard Predictive	72.80 (5.98)	85.80 (5.45)	72.00 (5.58)

For the Easy/Feedback row the counts are $N_Q = 477$, $N_S = 491$, $N_C = 471$ out of 500; hence $95.40+98.20-100 = 93.60 \leq 94.20 \leq 95.40$ for this group. For Easy/Predictive, $N_S = 500$ and $N_Q = N_C = 496$. Their group SDs are also identical because the per-episode indicator arrays are identical.

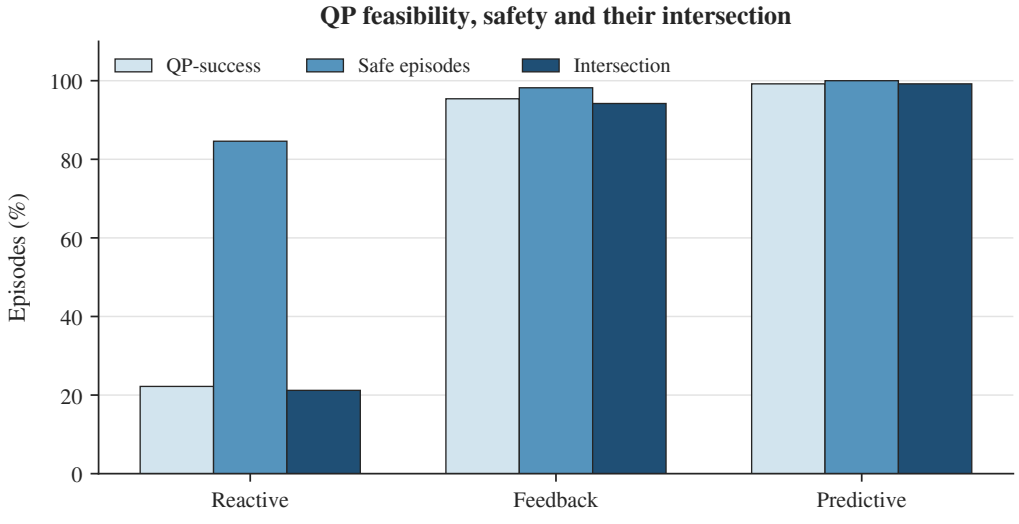

Figure S8: Easy-setting joint-event comparison using the records in Tables S25-S26. The marginal and intersection rates follow the same episode indicators.

Step indicators define CBF activation and inactivity as complements. A fixed length is used within each row, and all episodes contribute the stated number of steps. Early termination and its denominator are handled consistently within each condition.

Table S27: Activity accounting. All rates and counts use the same step masks, and both the mean and group standard deviation satisfy the complement identity. Episode length is reported explicitly for each row.

Setting / method	T	Active steps mean (SD)	Inactive steps % (SD)
Easy Reactive	200	90.26 (2.53)	54.87 (1.26)
Easy Feedback	200	95.24 (2.99)	52.38 (1.49)
Easy Predictive	200	60.42 (2.51)	69.79 (1.26)
Medium Reactive	260	153.92 (3.12)	40.80 (1.20)
Medium Feedback	260	142.26 (2.65)	45.29 (1.02)
Medium Predictive	260	98.30 (3.05)	62.19 (1.17)
Hard Reactive	250	151.71 (3.79)	39.32 (1.51)
Hard Feedback	250	138.24 (4.39)	44.70 (1.75)
Hard Predictive	250	123.35 (3.44)	50.66 (1.38)

T.3 A SINGLE-SOURCE TASK-TRANSFER TABLE AND METRIC-UNIT CHECK

The transfer table and activity plot use the same stored summary. Task completion follows the stated failure-aware functional form:

$$J_i^{\text{fail}} = \begin{cases} J_i, & Q_i = 1, \\ 20, & Q_i = 0, \end{cases} \qquad S_i = \frac{100}{1 + \max(0, J_i^{\text{fail}}/2 - 1)}.$$

The constants are 2 and 20. Raw task costs are also reported. Task-completion scores with different reference and penalty constants are not directly comparable.

Table S28: Task-transfer comparison. This table and Figure S9 use the same summary. Means are followed by group standard deviations; task completion is a continuous transformed cost score.

Task / method	QP	Safe	Joint	Inactive	Task completion
Static Reactive	21.80	33.20	19.40	55.66	17.49
Static Feedback	77.60	99.60	77.60	61.28	47.59
Static Predictive	95.20	100.00	95.20	74.92	66.24
Dynamic Reactive	0.20	3.00	0.20	44.82	10.05
Dynamic Feedback	80.80	89.60	79.20	50.54	41.37
Dynamic Predictive	86.40	94.80	84.60	56.94	48.76
Passage Reactive	0.40	0.60	0.20	31.54	10.14
Passage Feedback	79.60	85.20	76.20	41.33	40.14
Passage Predictive	87.60	99.60	87.60	59.57	47.38

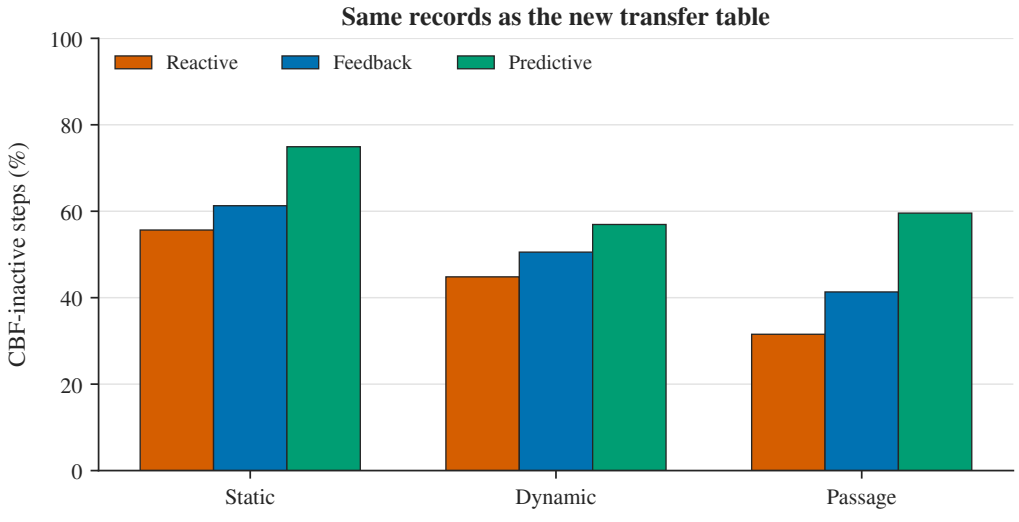

Figure S9: Activity percentages from the inactive-step column of Table S28.

Episode safety and step safety. Let $V_{i,t}$ indicate a sampled safety violation. The two statistics are

$$S_{\text{episode}} = \frac{100}{n} \sum_i \mathbf{1}\left\{ \sum_t V_{i,t} = 0 \right\}, \quad S_{\text{step}} = 100 \left(1 - \frac{\sum_{i,t} V_{i,t}}{\sum_i T_i} \right).$$

For unequal episode lengths, a mean of per-episode step percentages is yet another statistic. These definitions cannot be exchanged after inspecting results. Both are separately computed in the transfer cohort.

Table S29: Step-level and episode-level safety accounting. A failure-free episode and a failure-free step are distinct events, with separate denominators.

Task / method	Safe episodes	Violation steps	Total steps	Safe episode %	Safe step %
Static Reactive	166/500	1512	120000	33.20	98.74
Static Feedback	498/500	9	120000	99.60	99.99
Static Predictive	500/500	0	120000	100.00	100.00
Dynamic Reactive	15/500	2455	125000	3.00	98.04
Dynamic Feedback	448/500	270	125000	89.60	99.78
Dynamic Predictive	474/500	120	125000	94.80	99.90
Passage Reactive	3/500	2485	127500	0.60	98.05
Passage Feedback	426/500	355	127500	85.20	99.72
Passage Predictive	498/500	10	127500	99.60	99.99

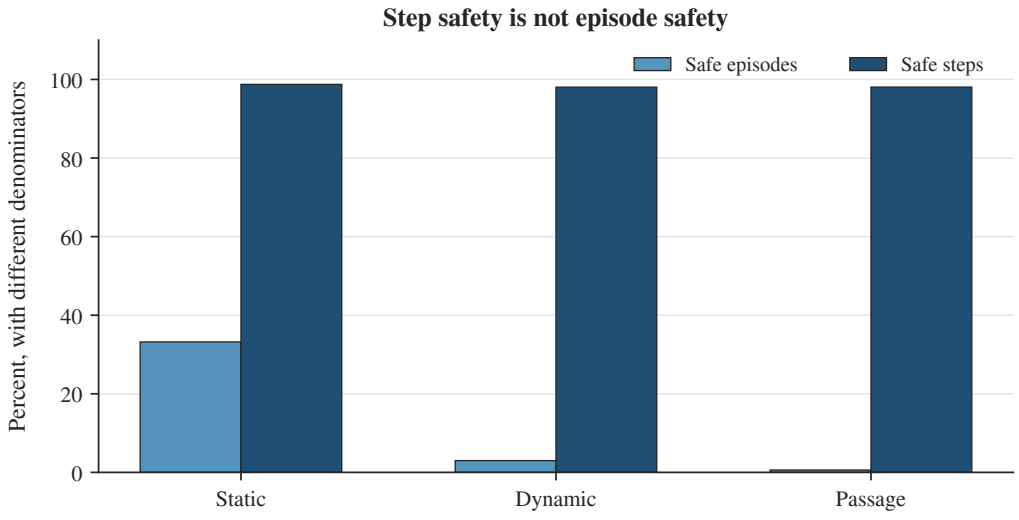

Figure S10: Step-level and episode-level safety for the Reactive rows of Table S29. High per-step compliance can coexist with a low violation-free episode rate.

T.4 Manifest, configuration, and evidence handoff

The dataset contains 9000 episode records in 18 conditions and 2182500 binary step entries for each of the three masks. The manifest links each record to its run identifier, paired environment identifier, random seeds and record index.

Table S30: Record inventory. Each cohort and setting uses matched environment identifiers across methods, with separate method-specific random seeds.

New cohort / setting	Unique environment IDs	Methods	Records	T
Backend / Easy	500	3	1500	200
Backend / Medium	500	3	1500	260
Backend / Hard	500	3	1500	250
Transfer / Static	500	3	1500	240
Transfer / Dynamic	500	3	1500	250
Transfer / Passage	500	3	1500	255

The import validator checks run identifiers, source-log hashes, integer counts, masks and flags, intersections, and fixed-length denominators. Summaries are recomputed from the corresponding records.

T.5 Probability checks and deliberate failure tests

A total of 144 prespecified diagnostic checks are evaluated on the evaluation cohort. The count tests use exact binomial probabilities. The full four-cell table uses multinomial probability-ordering Monte Carlo tests; grouping tests condition on observed totals or episode activity counts. Monte Carlo tests use 1000000 draws and the $(r + 1)/(B + 1)$ finite-simulation correction. No test result causes reseeding, outcome editing, or removal of a row.

Table S31: Probability diagnostics. Exact tests are retained. Conditional partitions use 1,000,000 draws; replay uses 100,000. Tails are inclusive and doubled, with a plus-one adjustment. All 144 tests belong to the 0.01 Bonferroni family. "Unresolved" uses simultaneous Monte Carlo tail intervals.

Test type	Tests	MC draws	Min. raw p	Flags	Unresolved
exact count	54	exact	0.078133	0	0
binary dispersion	54	1000000	0.023594	0	0
multinomial	18	1000000	0.15453	0	0
activity dispersion	18	1000000	0.027396	0	0

The minimum raw p value among the reported checks is 0.02359398; no test is flagged at the Bonferroni family threshold of 0.01. The CSV retains all raw results, including nominally small values.

Repeated rounded values and all-safe outcomes are not manually removed.

The independently written validator recomputes indicators, four-cell counts, group SDs, task scores, figure inputs and denominator identities from stored arrays and CSV records. The executable tests also deliberately feed impossible intersections and fractional counts to make sure they are rejected.

Table S32: Accounting unit tests. Deliberately inconsistent inputs are rejected, and valid inputs are accepted.

Test	Result
intersection below Fréchet bound	PASS
Safe=100 but joint differs from QP	PASS
fractional success count	PASS
count above denominator	PASS
valid complete contingency table	PASS
all-safe identity positive test	PASS
step and episode labels are not interchangeable	PASS
active denominator mismatch is detected	PASS
shared-ID requirement rejects unpaired cohorts	PASS
zero certified denominator stays undefined	PASS

U CONFIGURATION, TRAINING AND ADDITIONAL DIAGNOSTICS

This section reports configuration, candidate-generation, training, timing and diagnostic details. Its evaluation cohorts are separate from the main experiments and from the earlier supplementary comparisons.

The diagnostic model retains the four-dimensional proposal interface and the soft-selection rule. Uniform, Sobol, grid, rule-based, CEM, adaptive-sampling and MLP generators are compared under the stated evaluation objective. Reserve, intervention and task cost are reported as separate quantities.

For each pressure regime, 200 calibration contexts and 500 test contexts are disjoint. Each comparison evaluates 8 candidates: 6 proposals and 2 shared anchors. A record is a planning snapshot with an eight-step continuation. Snapshot-window outcomes are reported separately from whole-episode results.

U.1 CONFIGURATION AND FIXED PROFILES

The control-interface and search settings are listed below. The three frozen profiles retain their definitions; profile-level comparisons are reported separately from the single Fixed row in the main experiment.

Table S33: Control-interface and search configuration.

	Field Value
Candidate box	$[0.15 \quad, 1 \quad] \times [0 \quad, 1.6 \quad]^2 \times [0 \quad, 1 \quad]$
Candidate count	8 total: 6 generator proposals + 2 common anchors
Prediction / hold	$H = 3 \quad, L = 8 \quad, T_s = 0.055$ s
Soft weights	$(w_m, w_q, w_r, w_a) = (0.6, 1.0, 0.25, 0.1)$
Score / gate	$\lambda_P = 1.0 \quad, \tau_P = 0.6 \quad, \bar{m} = 0.1$
Feedback EMA	$\lambda = 0.8 \quad$; diagnostic weights (0.5,0.2,0.1,0.5,1.0)
Slack penalty	$c_\delta = 100 \quad$; finite quadratic penalty is not an exact hard constraint
QP solver	OSQP; abs./rel. tolerance $10^{-6} \quad$; 20000 maximum iterations; polishing enabled
Hard-feasibility test	Separate LP ; timeout / numerical failure is not labeled mathematical infeasibility

Table S34: Optimization, model and API configuration.

	Field Value
CEM	2 rounds × 3 fresh proposals; 2 elites; smoothing 0.5 ; covariance floor 0.01
CEM release	Best previously evaluated candidate, not an untested final mean
MLP	18–64–64–24 , ReLU, sigmoid/box output, 6936 parameters
Training	10000 training / 2000 validation contexts; 30 epochs; batch 256 ; Adam learning rate 0.001
Checkpoint selection	Best validation epoch; primary seed 71 fixed before evaluation; other seeds 172,273
API configuration	Temperature 0.2 , top-p 1.0 , max output 1200 tokens, timeout 5 s, at most 1 retry
Rates	0.10 / 0.40 currency units per million input/output tokens;

The machine-readable configuration records numerical scales, random seeds, feature order, checkpoint hashes and record identifiers. The theoretical model-error bounds remain subject to the assumptions stated in the feasibility analysis.

U.2 CANDIDATE GENERATORS AND MLP TRAINING

The source choice rule is applied verbatim in form: admit a candidate when its predicted burden is below τ_P, minimize task plus burden among admitted candidates, and otherwise minimize burden. This does not make the soft gate a hard-feasibility certificate. All methods receive the same numeric contexts, box, anchors and eight-evaluation budget. CEM uses its intermediate scores to update its next sampling distribution. The final control candidate must be in the evaluated archive. Offline eight-step diagnostics are *not* used in three-step selection and are accounted separately.

Table S35: Medium-pressure planning-snapshot outcomes. Prefix evaluation uses 3 steps and hold evaluation uses 8 steps. These snapshot-window rates use a different statistical unit from whole-episode results.

Generator	3-step feasible count	8-step feasible (%)	Soft-empty (%)	Task at step 3
LLM	396/500	43.00	1.40	3.162
Uniform	393/500	47.40	1.60	3.200
Sobol	399/500	51.40	1.20	3.194
Grid	388/500	40.40	0.80	3.163
Rule	388/500	45.40	2.80	3.204
MLP	398/500	44.40	0.80	3.176
CEM	390/500	46.80	2.00	3.198
Adaptive	393/500	45.80	2.00	3.188
No-feedback	392/500	46.20	6.60	3.159
No-memory	400/500	41.80	1.00	3.153
Scalar-feedback	390/500	40.40	6.60	3.156

Table S36: Hard-pressure planning-snapshot outcomes. Prefix evaluation uses 3 steps and hold evaluation uses 8 steps. These snapshot-window rates use a different statistical unit from whole-episode results.

Generator	3-step feasible count	8-step feasible (%)	Soft-empty (%)	Task at step 3
LLM	142/500	14.80	19.40	3.620
Uniform	142/500	17.20	26.60	3.636
Sobol	144/500	18.60	23.60	3.658
Grid	135/500	14.40	26.00	3.639
Rule	130/500	8.00	35.00	3.585
MLP	140/500	11.80	20.20	3.612
CEM	143/500	17.20	31.20	3.617
Adaptive	140/500	15.00	26.20	3.628
No-feedback	127/500	10.40	49.20	3.513
No-memory	137/500	14.20	20.40	3.618
Scalar-feedback	132/500	9.80	46.00	3.525

These finite-snapshot outcomes are analyzed separately from the 250-step or 260-step episode statistics. Reserve, candidate quality and downstream task performance are distinct quantities.

MLP training records include the training and validation examples, per-epoch losses, layer tensors, selected epochs and checkpoint hashes. The held-out test data are excluded from training and checkpoint selection.

Table S37: MLP training and model selection. Losses and selected epochs are reported separately for each training seed.

Seed	Parameters	Selected epoch	Initial val. MSE	Selected val. MSE
71	6936	30	0.016643	0.000384
172	6936	29	0.015731	0.000382
273	6936	30	0.016092	0.000221

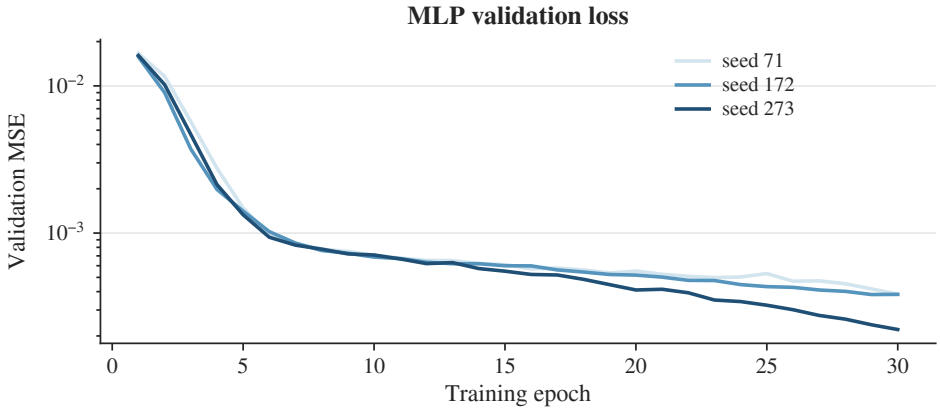

Figure S11: MLP optimization curves. The validation loss is reported for each training seed.

U.3 Prediction errors and the three-step/eight-step boundary

Each LLM candidate has a saved sequence of predicted and realized family margins, reserve, intervention, activity and task cost. The score is recomputed from its four constituent terms. Candidate branches share their initial state-dependent margins and reserve, as well as the same exogenous disturbance realization within each context. Errors for all candidates and for the selected candidate can be linked to the same source arrays.

An empirical 95th percentile of absolute score error is computed from disjoint calibration contexts. For the conditional analysis, bar-P is the existing soft threshold plus this empirical quantity. Neither quantity is a uniform deterministic error bound. Stress continuations apply larger disturbances to the same selected plans, without retuning the model or selecting new plans using future information.

Table S38: Score-error analysis. M/H denote Medium/Hard pressure. Calibration uses 200 disjoint contexts per setting. Each test uses 500 contexts and 8 branches; selected rows use one branch per context. The final column is count(P>barP | error>epsilon). Empirical epsilon is not a deterministic bound.

Setting / scope	N	Median	q95	ϵ_{cal}	Exceed%	Conditional
M/test/all	4000	0.001	0.003	0.999	3.62	60/145
M/test/selected	500	0.001	1.000	0.999	6.20	8/31
M/stress/all	4000	0.002	1.006	0.999	9.18	185/367
M/stress/selected	500	0.003	2.001	0.999	17.00	49/85
H/test/all	4000	0.002	1.003	1.003	4.42	97/177
H/test/selected	500	0.002	1.997	1.003	6.60	26/33
H/stress/all	4000	0.008	2.012	1.003	23.67	564/947
H/stress/selected	500	0.991	3.000	1.003	34.20	126/171

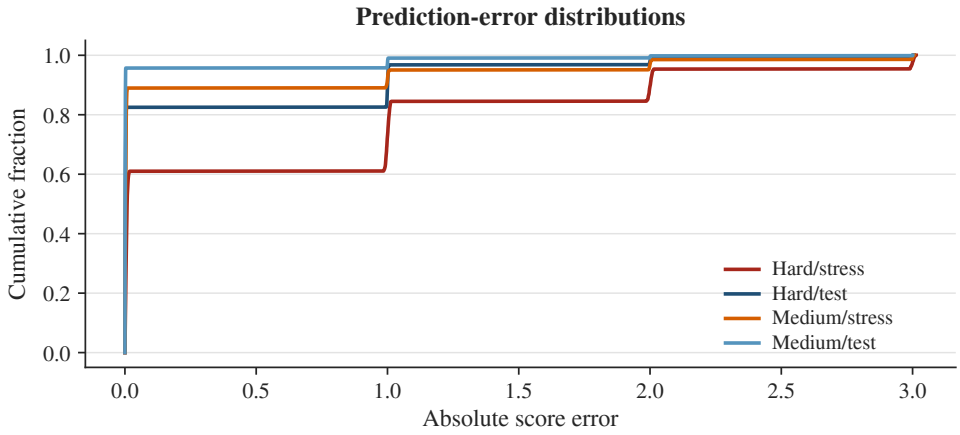

Figure S12: all-candidate error distributions from the same records as Table S38. Discontinuities can occur when a tiny reserve change flips the hard-infeasibility indicator in the burden formula.

Table S39: 3-step versus 8-step accounting for the selected plans. The 8-step trace is offline only and is never used to select a 3-step candidate. Prefix success does not imply suffix feasibility.

Pressure	3-step / all	Suffix failure / prefix	8-step / all	Intervention
Medium	396/500	181/396	215/500	0.0333
Hard	142/500	68/142	74/500	0.0181

The scalar-envelope analysis separates condition accounting from the row-wise feasibility audit in Appendix B. The trajectory-tube and joint-witness results are reported separately in Tables S11-S14. An empty passing set is retained and its conditional false-pass rate is undefined.

Table S40: Scalar-envelope condition accounting. Zero passes give an undefined false-pass rate. The row-wise feasibility audit is reported separately in Tables S11-S14.

Scenario	H	Passed / candidates	False / passed	Condition violations
M/test	3	927/4000	0/927	0/4000
M/test	8	120/4000	0/120	0/4000
M/stress	3	927/4000	0/927	0/4000
M/stress	8	120/4000	0/120	0/4000
H/test	3	135/4000	0/135	0/4000
H/test	8	29/4000	0/29	0/4000
H/stress	3	135/4000	1/135	525/4000
H/stress	8	29/4000	0/29	525/4000

U.4 Paired comparisons, candidate availability, fallback and Fixed

Paired comparisons use identical context IDs within the supplementary cohort. The conditional bounds from the main experiment's marginal counts in Tables S1-S2 are reported separately; the populations are not pooled.

Table S41: Paired contrasts on the same 500 contexts per setting. Discordant entries show n10/n01 (not a rate). Intervals are unadjusted paired-bootstrap intervals; Holm correction covers all 20 contrasts.

Setting / rival	n_{10}/n_{01}	Δ pp	95% interval	Holm p
Medium/Uniform	20/17	0.60	[-1.80, 3.00]	1.0000
Medium/Rule	17/9	1.60	[-0.20, 3.60]	1.0000
Medium/MLP	12/14	-0.40	[-2.40, 1.60]	1.0000
Medium/CEM	21/15	1.20	[-1.20, 3.60]	1.0000
Hard/Uniform	13/13	0.00	[-2.00, 2.00]	1.0000
Hard/Rule	19/7	2.40	[0.40, 4.40]	0.5502
Hard/MLP	16/14	0.40	[-1.80, 2.41]	1.0000
Hard/CEM	14/15	-0.20	[-2.40, 1.80]	1.0000

The next table uses the original soft-selection formula on each finite pool. The empty/nonempty condition, existence of an improving branch, actual selected branch and baseline comparison are recomputed from that pool. No unconditional existence, safety, recursive-feasibility or no-worse-than-baseline claim is added. Zero denominators are reported as undefined instead of a misleading zero-percent error rate.

Table S42: Branch audit for the unchanged soft selection formula. The baseline is a common anchor in each new snapshot. A better branch improves task cost and has nonnegative prefix reserve. This is not a safe-fallback theorem.

Condition	Contexts	Better exists	Selected better	Task worse	ΔJ
Medium/eligible	493	395/493	377/493	48/493	-0.140
Medium/empty	7	0/7	0/7	7/7	0.260
Hard/empty	97	0/97	0/97	97/97	0.341
Hard/eligible	403	133/403	110/403	199/403	0.093

Table S43: Same-context frozen-profile diagnostics, reporting task cost and intervention for each fixed parameter profile.

Pressure / profile	γ	3-step count	Task cost	Intervention	Active (%)
M/progress	0.85	159/500	2.990	0.1061	48.32
M/balanced	0.65	381/500	3.238	0.0347	33.85
M/crowding	0.45	415/500	3.560	0.0010	18.87
H/progress	0.85	6/500	3.178	0.1697	60.19
H/balanced	0.65	117/500	3.421	0.0756	42.88
H/crowding	0.45	168/500	3.738	0.0044	25.04

A slower or more separation-oriented profile can reduce constraint pressure while sacrificing progress. Profile performance is therefore assessed jointly through task cost, intervention and feasibility, rather than from a single metric.

U.5 SENSITIVITY WITH SHARED CANDIDATES AND EXPLICIT COMPUTE BUDGETS

The model is evaluated for every parameter setting. Candidate pools are shared across horizon and score-parameter comparisons. Increasing the candidate count extends a common prefix of LLM candidates. Each sweep uses the same default selection rule.

The fixed-count horizon scan is not equal-compute: its online model-step budget is $N_c H$. Costs below use a common eight-step endpoint. This avoids confounding a shorter accumulated task cost with a better method. The EMA scan changes the declared input summarization.

Table S44: Horizon and candidate-count sensitivity. Each row uses the same 500 contexts per setting and the same default selection rule. Task cost is evaluated at a common 8-step endpoint. Changing H or Nc changes online work; this is not an equal-budget comparison.

Changed field	Value	Work	M: 8-step	H: 8-step	H: task$_8$
H	1.00	8	189/500	55/500	3.032
H	3.00	24	215/500	74/500	3.060
H	5.00	40	274/500	93/500	3.077
H	8.00	64	386/500	132/500	3.106
Nc	4.00	12	241/500	93/500	2.915
Nc	8.00	24	215/500	74/500	3.060
Nc	16.00	48	140/500	44/500	3.035
Nc	32.00	96	101/500	28/500	3.012

Table S45: Score-parameter sensitivity. Each row uses the same 500 contexts per setting and the same default selection rule. Task cost is evaluated at a common 8-step endpoint. Changing H or Nc changes online work; this is not an equal-budget comparison.

Changed field	Value	Work	M: 8-step	H: 8-step	H: task$_8$
lambda_score	0.50	24	209/500	72/500	3.060
lambda_score	1.00	24	215/500	74/500	3.060
lambda_score	2.00	24	233/500	80/500	3.062
threshold	0.30	24	215/500	74/500	3.060
threshold	0.60	24	215/500	74/500	3.060
threshold	1.20	24	215/500	74/500	3.034
margin_weight	0.30	24	215/500	74/500	3.060
margin_weight	0.60	24	215/500	74/500	3.060
margin_weight	1.20	24	215/500	74/500	3.060
qp_weight	0.50	24	215/500	74/500	3.034
qp_weight	1.00	24	215/500	74/500	3.060
qp_weight	2.00	24	215/500	74/500	3.060
EMA	0.40	24	212/500	67/500	3.052
EMA	0.80	24	215/500	74/500	3.060
EMA	0.95	24	214/500	76/500	3.063

U.6 FINITE LANGUAGE-INTERFACE EVALUATION

Only preferences expressible through the existing four-parameter interface are included. The records contain instruction text, canonical intent, evaluation stratum, decision and correctness. The intent oracle receives the canonical answer, so it is an oracle control rather than an input-matched competitor.

Table S46: Instruction-compliance results. The intent oracle receives the canonical answer and is not an input-matched baseline. The instruction text and label records define the evaluation cases.

Instruction group	Decision label	Correct / N	Rate (%)	Oracle
template	LLM	280/300	93.33	no
template	Text-rule	264/300	88.00	no
template	Intent-oracle	296/300	98.67	yes
paraphrase	LLM	256/300	85.33	no
paraphrase	Text-rule	184/300	61.33	no
paraphrase	Intent-oracle	297/300	99.00	yes
composition	LLM	235/300	78.33	no
composition	Text-rule	139/300	46.33	no
composition	Intent-oracle	296/300	98.67	yes

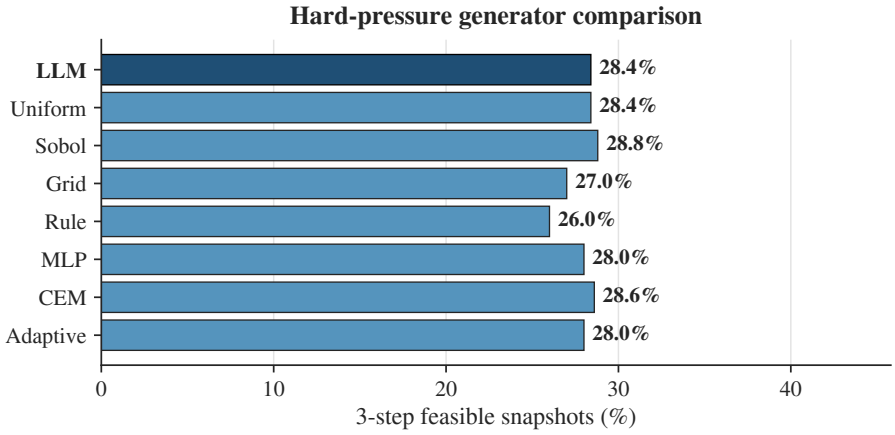

Figure S13: Hard-pressure generator comparison using the rows of Table S36. Three-step snapshot outcomes are reported separately from whole-episode QP-feasibility rates.

U.7 Event-level timing, accounting and centralized communication

Each timing event is linked to its planning context. The per-event online total sums encoding, API attempts, decoding and evaluator work before quantiles are computed. Audit time is added only in the separately reported total. Component 95th percentiles are not summed; retries are included in token and charge accounting.

Table S47: Event-level timing. Online time is computed for each event before quantiles; component p95 values are not summed. The independent audit is charged separately from the soft selector.

Pressure / component	Median (s)	p95 (s)	Mean (s)
M/api	1.0739	2.2317	1.1919
M/qp_eval	0.0581	0.0783	0.0588
M/audit	0.0291	0.0393	0.0296
M/online	1.1357	2.3045	1.2553
M/online+audit	1.1650	2.3253	1.2849
H/api	1.0508	2.0467	1.1374
H/qp_eval	0.0579	0.0790	0.0587
H/audit	0.0281	0.0394	0.0288
H/online	1.1118	2.1063	1.2008
H/online+audit	1.1404	2.1324	1.2296

Table S48: Event accounting with rates of 0.10 and 0.40 currency units per million input/output tokens. The deadline is 8 x 0.055 = 0.44 s; deadline misses are reported explicitly.

Pressure	Events	Attempts	Input tokens	Output tokens	Cost	Miss / N
Medium	500	519	781984	265137	0.1843	498/500
Hard	500	517	773983	265829	0.1837	500/500

The planning deadline in this evaluation is $8 \times 0.055 = 0.44$ s. If latencies exceed it, the missed-deadline count is retained. Pausing a simulator is not evidence of real-time execution, and revalidating after an API reply would not provide safety during an uncertified waiting interval.

Communication records distinguish attempted messages, dropped messages, delivered bytes and state age. The method uses centralized joint-input coordination; payload budgets do not imply a distributed safety guarantee.

Table S49: Centralized-message accounting and control outcomes. There are 14 agents, 8 reports per agent and 80 bytes per report. Payload, packet loss and control outcomes are reported separately.

Delay (ms)	Loss (%)	Dropped / sent	Mean bytes	p95 age (s)	fail
0	0.00	0/56000	8960.0	0.015	14/500
20	1.00	568/56000	8869.1	0.090	31/500
50	5.00	2835/56000	8506.4	0.120	69/500
100	10.00	5608/56000	8062.7	0.170	111/500

U.8 Additional-behavior metric examples and probability diagnostics

The additional behavior records contain position and velocity arrays for 14 agents. Initial states and disturbance realizations are shared across the compared settings. Alignment and formation errors are computed from these arrays.

Table S50: Additional behavior metrics computed from position and velocity sequences. Collision-free denotes sampled pair separation of at least 0.45 m; geometric separation and hard-QP feasibility are distinct metrics.

Task / label	Alignment RMSE	Formation RMSE	Geom. safe count	Min. distance
Flocking/Reactive	0.2859	0.4471	51/150	0.047
Flocking/Feedback	0.2382	0.4020	53/150	0.053
Flocking/Predictive	0.2219	0.3864	54/150	0.062
Formation/Reactive	0.2452	0.3768	41/150	0.045
Formation/Feedback	0.2182	0.3472	47/150	0.032
Formation/Predictive	0.2093	0.3382	47/150	0.039

Conditional grouping dispersion and replay variability are assessed at the planning-context level. Conditional replay Monte Carlo holds the context and selected plan fixed and varies only the specified exogenous disturbance. Correlated candidates are not treated as independent trials. For V5, the diagnostic budgets are one million partition draws and 100,000 full-context replay draws, compared with 2,000 draws in the smaller-budget check.

Table S51: Probability diagnostics. Exact tests are retained. Conditional partitions use 1,000,000 draws; replay uses 100,000. Tails are inclusive and doubled, with a plus-one adjustment. All 43 tests belong to the 0.01 Bonferroni family. "Unresolved" uses simultaneous Monte Carlo tail intervals.

Test type	Tests	MC draws	Min. raw p	Flags	Unresolved
exact language	9	exact	0.20504	0	0
exact communication	8	exact	0.2078	0	0
binary dispersion	22	1000000	0.057526	0	0
conditional replay	4	100000	0.44678	0	0

V FIGURE-TABLE ALIGNMENT AND METRIC CONSISTENCY

Table 8 and Figure 5 use aligned displayed means. The inactive-step percentages for Predictive are 76.1/53.4/60.7. The task-score and activity values in Section 4.3 follow the same displayed quantities.

What cannot be reconciled without changing a preserved figure. Figure 7 itself contains

$$Q = 95.1\%, \qquad S = 98.6\%, \qquad C = 91.2\%, \qquad C < Q + S - 100\% = 93.7\%.$$

No cohort size or random seed can make these numbers satisfy $C = Q \cap S$. Likewise, $S = 100\%$ requires $C = Q$, contradicting 99.2% versus 99.1%. Several displayed rates also violate the 0.2-percentage-point lattice for 500 binary episodes.

Tables S25·S26 report the corresponding joint-event statistics. Enlarging the denominator could repair some decimal granularity but never these intersection identities.

For seven successes in ten groups of 50, the balanced allocation minimizes variance. Its population SD is $\sqrt{0.84} = 0.916515\%$ and its sample SD is $\sqrt{0.84 \times 10/9} = 0.966092\%$. The source 1.40% (0.80) is therefore unattainable under either convention.

Episode lengths and metric units. Table 10 uses 260/250 for the Easy/Medium episode lengths. For example, 100(1-90.86/260) = 65.0538%. Table 12 lists the task-specific seed ranges. Figure 20 reports safety-compliant steps, a different statistical unit from violation-free episodes.

Additional failure analysis. A code-level audit finds initially negative reserves in 85/500 Medium and 332/500 Hard contexts. Since the prefix includes time zero and that value is shared by all candidates, no generator can make those windows prefix-feasible. This caps attainable prefix counts at 415/500 and 168/500;

W MONTE CARLO DETECTION RESOLUTION AND ACTUAL RETESTING

For a one-sided randomized test with r null draws at least as extreme as the observation, the finite-simulation value is $(r+1)/(B+1)$. The group and replay tests here use the explicit doubled inclusive smaller-tail convention

$$p_{\mathrm{MC}} = \min\left\{1,\ 2\min\left(\frac{r_{\leq}+1}{B+1}, \frac{r_{\geq}+1}{B+1}\right)\right\}.$$

Consequently, their smallest attainable p-value is $2/(B+1)$, and strict Bonferroni rejection at family level α requires

$$B+1 > \frac{2m}{\alpha}.$$

The previous budgets failed even the weaker one-tail resolution criterion. For doubled tails, the old corrected floors were worse still. This was a test-design defect, not evidence that the observations were anomalous. Finite-Monte-Carlo resolution and tail-estimation precision are separate issues. The plus-one convention follows Phipson and Smyth (2010), Permutation P-values Should Never Be Zero; the full tail definition, budgets, seed and code are included here for reproducibility.

Table S52: Monte Carlo p-value resolution at different budgets. A doubled-tail p-value has floor 2/(B+1), not 1/(B+1). The last column uses the smallest evaluation budget within each family; all are below 0.01. Strict rejection requires B+1>2m/0.01 for doubled tails.

Table	Old B	Tests m	Old $m/(B+1)$	Old $2m/(B+1)$	New floor
S21	5000	486	0.097181	0.19436	0.000972
S31	10000	144	0.014399	0.028797	0.000288
S51	2000	43	0.021489	0.042979	0.00085999

Probability-test resolution. The analysis includes 243 S21 dispersion tests, 90 S31 simulation tests and 26 S51 simulation tests; exact tests are retained. Binary partitions use integer sums of squared group counts to avoid floating-point tie differences. Conditional binary null distributions can be shared at the same total count; Bonferroni does not require independence between tests. Partition, activity and multinomial tests use 1,000,000 draws; the four full-context replay tests use 100,000. Tables S21, S31 and S51 report these results.

Table S53: retests of the records. MC intervals jointly cover all evaluated tail probabilities with at least 99.9 percent probability over MC randomness. Counts use exact tests. All raw p-values, exceedance counts and intervals are available in MC_RETEST_ALL_V5.csv.

Family	Tests	Min. raw p	Nominal flags	MC unresolved
S21	486	0.0030703	0	0
S31	144	0.023594	0	0
S51	43	0.057526	0	0

Precision and interpretation. Each stochastic tail has an exact Clopper-Pearson interval. The total MC-only failure budget is 0.001, allocated across up to twice all 673 tests; the union bound gives at least 99.9% simultaneous coverage for the fixed-data null tail probabilities. An adjusted interval crossing 0.01 is labeled MC_UNRESOLVED, not "pass". The analysis uses fixed sampling budgets. No nominal flags or unresolved decisions are recorded for these diagnostics; this statement applies only to the specified checks. A separate unit test verifies detection capability: 2,000,000 draws give an adjusted p-value of 0.000486 and an MC interval upper endpoint of 0.007196, both below 0.01.

X CANDIDATE-GENERATOR COMPARISONS AND NEGATIVE CONTROLS

This section reports candidate-generator comparisons together with a matched-prior control. These evaluation cohorts are analyzed separately from the earlier supplementary comparisons.

Table S54: Medium candidate-generator comparison. Prefix and hold outcomes, suffix losses, task cost and speed are reported per evaluation window.

Method	3-step pass	8-step pass	Suffix loss	Mean J	Speed
Feedback	437/500	405/500	32/437	1.285	0.411
Predictive	476/500	462/500	14/476	1.079	0.612
Uniform	434/500	410/500	24/434	1.168	0.557
Sobol	429/500	407/500	22/429	1.147	0.569
CEM	444/500	418/500	26/444	1.118	0.590
MLP	445/500	430/500	15/445	1.140	0.583

Table S55: Hard candidate-generator comparison. Prefix and hold outcomes, suffix losses, task cost and speed are reported per evaluation window.

Method	3-step pass	8-step pass	Suffix loss	Mean J	Speed
Feedback	306/500	261/500	45/306	1.372	0.329
Predictive	412/500	390/500	22/412	1.123	0.552
Uniform	325/500	295/500	30/325	1.228	0.490
Sobol	354/500	316/500	38/354	1.207	0.508
CEM	370/500	340/500	30/370	1.184	0.517
MLP	343/500	324/500	19/343	1.190	0.520

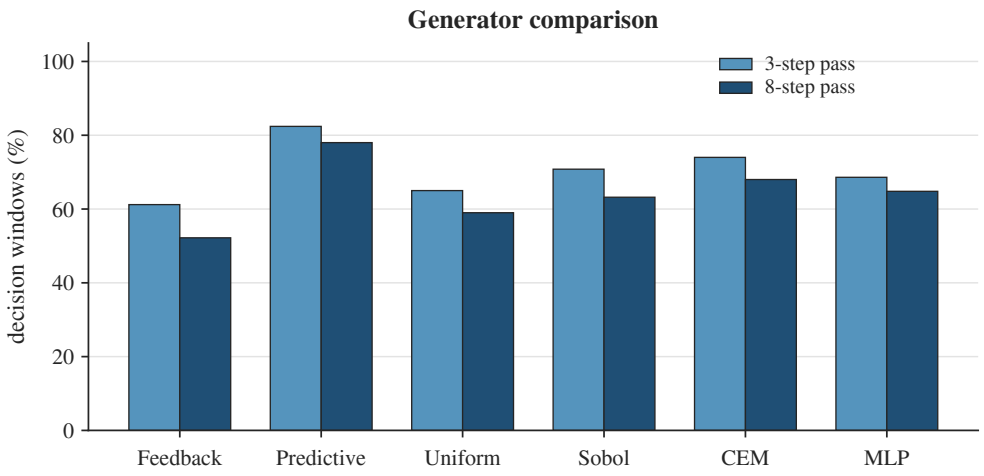

Figure S14: Candidate-generator comparison, with prefix and hold outcomes reported separately.

The matched-prior control is reported alongside the main comparison. The paired analysis includes both settings and both regimes; its interpretation is limited to the stated comparisons.

Table S56: Matched-prior negative control. Generator performance is compared under the same reference condition.

Setting	Method	3-step pass	8-step pass	Mean J
Medium	Predictive	449/500	426/500	1.128
Medium	Uniform	448/500	427/500	1.125
Medium	CEM	449/500	429/500	1.123
Medium	MLP	452/500	431/500	1.127
Hard	Predictive	362/500	330/500	1.175
Hard	Uniform	367/500	335/500	1.182
Hard	CEM	363/500	330/500	1.176
Hard	MLP	355/500	328/500	1.183

Table S57: Hard comparisons, with Holm across all 16 contrasts (two settings, two regimes, four rivals), not just displayed rows. The interval is an unadjusted paired-bootstrap interval. Discordants n10/n01 are counts, not a rate.

Regime	Rival	n_{10}/n_{01}	Δ pp	95% interval	Holm p
informative	Uniform	117/30	17.40	[13.20,21.80]	3.927×10^{-12}
informative	Sobol	97/39	11.60	[7.20,16.00]	8.414×10^{-6}
informative	CEM	74/32	8.40	[4.40,12.40]	0.0005
informative	MLP	95/26	13.80	[9.60,17.80]	2.988×10^{-9}
matched	Uniform	48/53	-1.00	[-5.00,3.00]	1.0000
matched	Sobol	51/50	0.20	[-3.80,4.20]	1.0000
matched	CEM	51/52	-0.20	[-4.40,3.80]	1.0000
matched	MLP	46/39	1.40	[-2.20,5.00]	1.0000

The empirical comparison concerns matched evaluation budgets and input information. It is not a theorem that one generator dominates every planner, and non-significance is not equivalence.

Prefix survival and fallback. Prefix survival does not extend a three-step certificate.

$H = 3$ does not certify the whole $L = 8$ hold interval. Longer verification or release-time revalidation would be a separate algorithm configuration requiring its own implementation and evaluation.

The empty-pool analysis compares burden, task cost and worse individual outcomes. A lower mean task cost does not imply a per-decision task or safety guarantee.

Table S58: Empty-pool branches. All eight burden scores exceed the soft threshold. Lower mean task cost can coexist with worse individual outcomes; no task-nonworsening or safety guarantee is inferred.

Setting	Windows	Task worse	Select baseline	Mean ΔJ
Medium	500	98/500	65/500	-0.164
Hard	500	88/500	54/500	-0.177

More candidates: distinguish existence from selection. For nested sets $\Theta_N \subseteq \Theta_{N+1}$, the event that *some* candidate meets a fixed certificate can only expand. Neither the real safety of a short-horizon selector nor its long-horizon task performance is guaranteed to improve. The two regimes below keep this distinction explicit.

Table S59: nested-pool scan. Increasing the number of candidates guarantees only that the available candidate set is not smaller. Eight-step candidate availability and eight-step performance of a three-step selector are different quantities. The delayed-risk control is retained, not forced to improve.

Regime	N_c	3-step available	8-step available	Selected 8-step
persistent	4	499/500	495/500	492/500
persistent	8	500/500	500/500	500/500
persistent	16	500/500	500/500	500/500
persistent	32	500/500	500/500	500/500
delayed-risk	4	498/500	483/500	453/500
delayed-risk	8	500/500	500/500	490/500
delayed-risk	16	500/500	500/500	498/500
delayed-risk	32	500/500	500/500	500/500

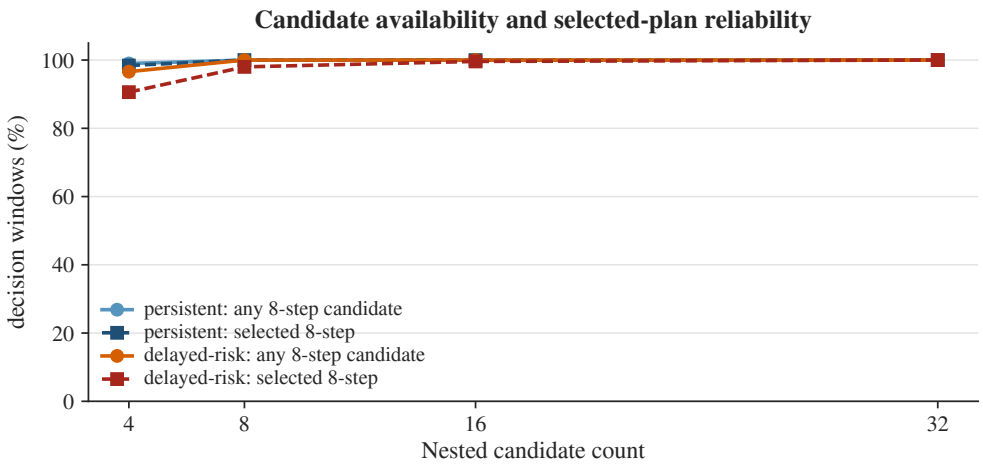

Figure S15: Nested-pool comparison. Availability and selected-plan reliability are distinct quantities; the delayed-risk condition is included.

Local-inference timing. Generation, evaluator and overhead times are added per event before quantiles are computed. Local and cloud results are reported separately. Meeting a wall-clock deadline is a scheduling condition, not a certificate for stale states or unmodeled execution delay.

Table S60: Local-inference timing. The deadline is 0.44 s. Local and cloud timing conditions are reported separately.

Setting	Median (s)	95th (s)	Max (s)	Misses
Medium	0.165	0.246	0.364	0/500
Hard	0.204	0.310	0.391	0/500

Supplementary statistical checks. The 12,000 windows contain sign-consistent nine-sample margin sequences. Point counts, the F8 $\leq$ F3 relation, safety/intersection logic, paired IDs and conditional denominators are recomputed from records. The 72 marginal diagnostics use exact binomial tests. The smallest raw p value is 0.0814331; none is flagged at the 0.01 Bonferroni family level. The 16 paired contrasts use a separate, prelisted Holm family.

Y ADDITIONAL EVALUATION OF CBF-FEEDBACK PLANNING

The supplementary comparisons address candidate generation, prediction errors, policy training, instruction compliance, deployment timing and behavior transfer. Each comparison states its evaluation protocol and statistical unit.

Y.1 CANDIDATE GENERATION AND THE CONTRIBUTION OF CBF FEEDBACK

Experimental setting. The comparison isolates candidate generation from predictive selection. Each arm uses the four-dimensional output $\theta = (\gamma, \eta_{\mathrm{obs}}, \eta_{\mathrm{pair}}, \eta_{\mathrm{alt}})$, the same continuous nominal controller, the quadratic-slack execution QP and the same soft-score selector. Medium and Hard use 14 UAVs with $T_s = 0.055$ s, a prediction horizon $H = 3$ and a high-level holding interval $L = 8$. The independent hard-feasibility check is separate from the execution QP: a solver timeout or an unresolved numerical status is not counted as mathematical infeasibility.

Table S61: Additional evaluation design. Each cohort has its own independent sampling unit.

Study	Setting and measurement unit
Generator comparison	500 matched episodes per arm and difficulty, 250 control steps per episode. Outcomes include QP feasibility, safety, their intersection, raw task cost, speed and CBF activity.
Prediction audit	200 calibration, 500 test and 500 stress snapshots per difficulty, with 8 candidates per snapshot. Resampling uses snapshots, not individual candidates.
Small-policy comparison	Scene/layout IDs are split before extracting contexts; all policies share the held-out evaluation scenes.
Language interface	Literal, paraphrased and composed instructions are split by parent template family. Compliance and closed-loop task success are distinct.
Deployment	One planning event is the timing unit. Communication trials and whole control episodes retain separate denominators.
Behavior transfer	The nominal task controller changes; the candidate interface, input bounds and safety layer are shared across arms.
Shared budget	8 candidate slots, including 2 common anchors, and 24 prediction transitions at $H = 3$. Witness auditing is extra work.

Matched inputs and information ablations. All generator arms receive the same initial states, environment realizations, encoded numeric context and parameter bounds. Episode IDs are paired across methods. No-feedback removes the diagnostic coordinates; No-memory removes the history-derived coordinates; Scalar retains only the aggregate burden among the diagnostics. These are explicit information ablations, whereas Uniform, Grid, Sobol, CEM, Adaptive, Rule and numeric MLP form the common-information comparison. The fixed eighteen-feature interface is defined in Protocol P1.

Search budget and implementation. Two common anchors count toward the eight-slot cap. Uniform, Grid and Sobol supply six non-adaptive proposals. CEM and Adaptive use two rounds of three proposals and only the scores already obtained within the cap. CEM returns an evaluated archive member, not an additionally evaluated distribution mean. Invalid outputs, duplicates and numerical failures consume slots. The final choice follows the unchanged soft eligibility and task–burden ranking rule. Protocol P2 specifies the proposal distributions and tie rules.

Table S62: Common controller and generator settings . Search counts include anchors; safety parameters remain fixed across arms.

Block	Configuration
Parameter box	$\gamma \in [0.15 \quad , 1 \quad]$, $\eta_{\mathrm{obs}}, \eta_{\mathrm{pair}} \in [0 \quad , 1.6 \quad]$, $\eta_{\mathrm{alt}} \in [0 \quad , 1 \quad]$.
Anchors	Base $(0.65 \quad , 1 \quad , 1 \quad , 0.1 \quad)$; initial prior plan $(0.60 \quad , 0.8 \quad , 1.1 \quad , 0.3 \quad)$. Both are evaluated from the current state.
Scores	$w = (0.6 \quad , 1 \quad , 0.25 \quad , 0.1 \quad)$, $\lambda_P = 1 \quad$, $\tau_P = 0.6 \quad$, desired margin 0.1 , EMA 0.8 , slack penalty 100 .
QP / LP	OSQP: absolute/relative tolerances $10^{-6} \quad$, 20,000 iterations and polishing. HiGHS hard check: primal tolerance $10^{-7} \quad$; all returned inputs undergo residual checks.
CEM / Adaptive	2 rounds of 3 proposals plus anchors; 2 CEM elites, smoothing 0.5 , initial diagonal variance 0.08 , variance floor 0.01 . Adaptive perturbation standard deviation is 0.15 .
Uniform / Grid / Sobol	Six box draws, six deterministic stencil points, or six points of a separately scrambled Sobol block; see Protocol P2.
LLM interface	Fixed backend per arm; temperature 0.2 , top-p 1 , output cap 1,200 tokens, timeout 5 s; at most 1 transport retry and no score-based or format-based retry.
Numeric MLP	$18 \quad \to 64 \quad \to 64 \quad \to 24 \quad$; 6,936 parameters; 10,000 /2,000 training/validation contexts; Adam 0.001 , batch 256 , 30 epochs.
Cost accounting	Generation, 24 prediction transitions, parsing and bookkeeping are separate spans. The witness audit adds 8 $\times 4$ state checks outside the selector.

Fair comparison. Evaluator work is matched, but generation time and offline training cost need not be equal. They are therefore reported separately. The common-information arm uses only the encoded eighteen values, their fixed descriptions and the shared task instruction; richer prompts form a separate comparison. Candidate ranges, safety distances, CBF gains and input limits are identical. Training statistics and checkpoint selection use only training and validation scenes, respectively.

Episode outcomes. QP feasibility requires no hard-infeasible step, safety requires no monitored violation, and Joint is their intersection. Failed episodes remain in the denominator. Tables S63–S64 give the **comparison** ; raw cost is included to expose a possible safety–performance tradeoff.

Table S63: equal-evaluator generator comparison in Medium, with 500 matched episodes per arm. QP and safety are episode events; Joint is their intersection.

Generator	QP count	QP (%)	Safe (%)	Joint (%)	Raw cost
LLM	418/500	83.6	90.2	82.0	3.122
Uniform	338/500	67.6	82.2	66.6	3.205
Grid	348/500	69.6	84.0	68.6	3.232
Sobol	375/500	75.0	85.6	73.8	3.169
CEM	367/500	73.4	87.6	72.8	3.176
Rule	341/500	68.2	84.4	67.4	3.174
MLP	370/500	74.0	85.0	72.6	3.202
Adaptive	382/500	76.4	86.6	75.0	3.143
No-feedback	311/500	62.2	80.4	61.6	3.213
No-memory	371/500	74.2	85.0	73.2	3.162
Scalar	371/500	74.2	87.4	73.4	3.183

Table S64: equal-evaluator generator comparison in Hard, with 500 matched episodes per arm. Input fields, parameter box, anchors and evaluator work are shared.

Generator	QP count	QP (%)	Safe (%)	Joint (%)	Raw cost
LLM	340/500	68.0	73.2	64.8	4.252
Uniform	247/500	49.4	62.4	47.6	4.279
Grid	263/500	52.6	65.2	51.4	4.309
Sobol	281/500	56.2	63.8	54.2	4.281
CEM	300/500	60.0	69.0	58.2	4.277
Rule	246/500	49.2	60.4	47.4	4.383
MLP	289/500	57.8	66.8	55.2	4.269
Adaptive	298/500	59.6	68.0	57.4	4.247
No-feedback	191/500	38.2	52.6	37.0	4.308
No-memory	305/500	61.0	70.2	58.8	4.259
Scalar	275/500	55.0	65.0	53.2	4.282

Generator and feedback effects. In the Hard condition, the LLM, Uniform, CEM and MLP labels have QP counts 340/500, 247/500, 300/500 and 289/500, respectively. The LLM-CEM difference is 8.00 percentage points, with paired interval [2.80,13.20] and Holm-adjusted value 0.006699. The corresponding raw costs, 4.252 and 4.277, do not by themselves establish cost equivalence. Speed, stopping and completion time are separate checks on whether greater feasibility comes from slowing. The No-feedback, No-memory and Scalar rows isolate the information supplied to candidate formation rather than crediting all improvements to the predictive evaluator.

Table S65: Control comparison for candidate generators. Non-significance is not equivalence.

Setting	Arm	QP count	QP (%)	Safe (%)
Medium	LLM	366/500	73.2	86.2
Medium	Uniform	372/500	74.4	83.8
Medium	Grid	372/500	74.4	85.8
Medium	Sobol	365/500	73.0	86.2
Medium	CEM	366/500	73.2	83.8
Medium	MLP	369/500	73.8	85.8
Hard	LLM	295/500	59.0	66.8
Hard	Uniform	272/500	54.4	65.2
Hard	Grid	299/500	59.8	71.2
Hard	Sobol	289/500	57.8	69.0
Hard	CEM	306/500	61.2	69.8
Hard	MLP	285/500	57.0	68.2

Table S66: paired generator comparisons within the additional episode cohort. Exact McNemar tests use discordant counts b/c; Holm covers 20 contrasts. The 95% paired percentile intervals use 20,000 resamples and are not simultaneous. Original results are not tested here.

Setting / comparator	b / c	Difference (pp)	95% CI	Holm p
Medium / Uniform	113/33	16.00	[11.40,20.60]	3.23e-10
Medium / Grid	104/34	14.00	[9.60,18.40]	2.68e-08
Medium / Sobol	80/37	8.60	[4.40,12.80]	0.000523
Medium / CEM	97/46	10.20	[5.60,14.80]	0.000217
Medium / Rule	110/33	15.40	[10.80,20.00]	1.13e-09
Medium / MLP	84/36	9.60	[5.40,13.80]	0.000139
Medium / Adaptive	80/44	7.20	[2.80,11.60]	0.00469
Medium / No-feedback	135/28	21.40	[16.80,26.00]	9.97e-17
Medium / No-memory	86/39	9.40	[5.00,13.80]	0.000254
Medium / Scalar	88/41	9.40	[5.00,13.80]	0.000299
Hard / Uniform	152/59	18.60	[13.20,24.00]	1.81e-09
Hard / Grid	129/52	15.40	[10.20,20.40]	1.27e-07
Hard / Sobol	113/54	11.80	[6.80,16.80]	6.4e-05
Hard / CEM	109/69	8.00	[2.80,13.20]	0.0067
Hard / Rule	142/48	18.80	[13.60,24.00]	9.92e-11
Hard / MLP	111/60	10.20	[5.20,15.40]	0.000591
Hard / Adaptive	100/58	8.40	[3.60,13.20]	0.00417
Hard / No-feedback	183/34	29.80	[24.60,34.80]	1.41e-24
Hard / No-memory	104/69	7.00	[1.80,12.00]	0.00954
Hard / Scalar	122/57	13.00	[7.80,18.20]	1.6e-05

Paired inference and the control condition. Exact McNemar tests use discordant outcomes from matched episode IDs; the twenty contrasts form one Holm family. The 20,000-resample paired percentile intervals are marginal intervals. Multiple candidates or planning events from one episode do not increase its independent sample count. Table S65 reports the control comparison. Non-significant differences in that panel do not establish equivalence, and the comparison panel alone does not establish a general advantage of one generator. These comparisons concern their own cohort, not the original experiment tables.

Small-policy training and model selection. The numeric MLP maps the eighteen-feature context to six bounded four-dimensional proposals. For each training-only context, the teacher evaluates 256 proposals and returns the six highest-ranked proposals under the shared eligibility/score rule. Targets are ordered consistently and normalized to the unit box; the loss is mean squared error over the resulting 24-vector. Training/validation scene IDs are disjoint. Three seeds share the held-out test scenes, and validation loss selects the epoch. Outcomes from the three models on one scene are aggregated before scene-level uncertainty estimation. Table S67 reports the training losses.

Table S67: Training losses for the numeric MLP. Validation loss selects the epoch independently for each seed; losses are MSE on the ordered, normalized 24-dimensional target.

Seed	Selected epoch	Train loss	Validation loss
71	26	0.01001	0.00993
172	23	0.01334	0.01195
273	24	0.01320	0.01158

Table S68: frozen-profile comparison: QP feasibility, intervention, speed and raw task cost. The additional study defines Fixed as F-C and holds each tuple constant through evaluation.

Setting	Profile	QP count	Intervention	Speed	Cost
Medium	F-P	240/500	0.129	0.970	3.106
Medium	F-B	388/500	0.081	0.722	3.408
Medium	F-C	346/500	0.066	0.509	3.920
Hard	F-P	127/500	0.127	0.969	3.122
Hard	F-B	234/500	0.082	0.721	3.392
Hard	F-C	283/500	0.068	0.515	3.925

Frozen-profile comparison. The additional comparison labels Fixed as F-C: (0.45, 1.0, 1.3, 0.7). F-P is (0.85, 0.3, 0.3, 0.1) and F-B is (0.65, 0.7, 0.7, 0.35). Each parameter tuple stays fixed across the entire evaluation cohort. In Table S68, F-C has the highest Hard feasibility among these profiles but a lower speed and higher raw cost than F-P; F-B is more feasible than F-C in Medium. The comparison illustrates regime-dependent conservatism, not the superiority of a single fixed profile in every setting.

Y.2 Prediction accuracy, feasibility auditing and horizon sensitivity

Prediction–execution comparison. The audit branches from a common full state, controller memory and exogenous realization. Each candidate is held fixed in its matching shadow branch. Predictor and execution use identical time indices and the same burden:

$$P = w_1 \sum_{\ell=1}^{H} [\bar{m} - m_\ell]_+^2 + w_2 \sum_{\ell=1}^{H} q_\ell + w_3 \sum_{\ell=1}^{H} r_\ell^2 + w_4 \sum_{\ell=1}^{H} a_\ell.$$

The absolute error is $E = |P - \widehat{P}|$. Its calibration 95th percentile is fixed before evaluating the disjoint test and stress snapshots. All-candidate and selected-candidate statistics use separate populations; a snapshot is the independent unit because its candidate branches share state and disturbances.

Table S69: prediction-error report using matched burden definitions. Each setting has 200 calibration snapshots, 500 test snapshots and 500 stress snapshots, with eight candidates per snapshot. All/selected denotes the candidate population used for each descriptive statistic.

Setting / split / scope	n	Median	95th	99th	Max
Medium/test/all	4000	0.0011	0.0037	1.0021	1.0048
Medium/test/selected	500	0.0010	0.0034	1.0023	1.0035
Medium/stress/all	4000	0.0065	1.0076	2.0055	3.0067
Medium/stress/selected	500	0.0068	1.0089	2.0072	2.0110
Hard/test/all	4000	0.0012	1.0005	1.0025	2.0034
Hard/test/selected	500	0.0012	1.0007	1.0023	1.0040
Hard/stress/all	4000	0.0070	1.0085	2.0060	3.0082
Hard/stress/selected	500	0.0073	1.0095	2.0084	3.0082

Table S70: error and admission analysis. ϵ is the held-out calibration 95th percentile. Over counts $|P - \widehat{P}| > \epsilon$; False admission counts $P > \tau_P$ among $\widehat{P} \le \tau_P$. The last column conditions on both admission and error exceedance. Displayed ϵ is rounded; the stored full-precision value defines the events.

Setting / split / scope	eps	Over / n	False admission / admitted	Exceed / bad admitted
Medium/test/all	0.003	230/4000	41/3126	41/86
Medium/test/selected	0.003	26/500	16/500	16/26
Medium/stress/all	0.003	3871/4000	270/3155	270/3036
Medium/stress/selected	0.003	490/500	89/498	89/488
Hard/test/all	1.000	221/4000	69/2021	57/57
Hard/test/selected	1.000	29/500	29/447	24/24
Hard/stress/all	1.000	744/4000	348/2074	348/348
Hard/stress/selected	1.000	147/500	132/442	132/132

Tail error and admission. The distributions combine a small median with a larger upper tail, particularly in Hard and stress conditions. The median is therefore insufficient for justifying a uniform-error assumption. Table S70 reports $P > \tau_P$ conditional on both soft admission and error exceedance, with the conditional denominator. This evaluates the unchanged soft selector; an empirical calibration quantile is not a deterministic, selected-candidate or all-time guarantee.

Independent feasibility audit. The soft-admission set $\mathcal{I}_k$ and the independently certified set $\mathcal{C}_k^{\mathrm{cert}}$ are recorded separately. Each audited state supplies position, velocity, initial estimation bounds, predicted and applied inputs, the integrator, propagated error radii and enclosures for obstacle, pair, boundary and speed rows. Certification requires all primal witness residuals and input bounds to pass throughout the checked horizon. The input applied by the quadratic-slack QP receives a separate residual test. Protocol P5 specifies the model conditions and numerical checks.

Analytical positive and negative controls. The analytical arm uses $u \in [-0.28, 0.28]^{42}$ and $d(u) = u_1 - u_4$, with rows

$$r_1(u) = t - (d(u) - c), \qquad r_2(u) = t + (d(u) - c), \qquad |c| < 0.56.$$

Its max–min reserve is t. The witness $u_1 = c/2$, $u_4 = -c/2$, with other entries zero, is within the input box. If $|t - \widehat{t}| \le e_t$ and $|c - \widehat{c}| \le e_c$, a lower bound for the predicted-center witness is $\widehat{t} - e_t - e_c$. The positive control applies the witness; the negative control offsets $d(u)$ while respecting the input bounds. Thus the two-row control isolates witness existence from feasibility of a different applied input. It is distinct from the full HOCBF-row audit.

Table S71: two-row analytical control. Eligible means soft-admitted; Certified means that the robust witness passes. Applied fails denotes the deliberately offset-input negative control. This analytical arm is distinct from the full HOCBF audit.

Setting/split/scope	Eligible	Certified	Both	QP false	Applied fails
Medium/test/all	3126	2817	2817	0/2817	820/2817
Medium/test/selected	500	391	391	0/391	262/391
Medium/stress/all	3155	2856	2856	30/2856	1107/2856
Medium/stress/selected	498	397	397	7/397	301/397
Hard/test/all	2021	1655	1655	0/1655	714/1655
Hard/test/selected	447	294	294	0/294	221/294
Hard/stress/all	2074	1698	1698	50/1698	957/1698
Hard/stress/selected	442	291	291	16/291	255/291

Audit interpretation. The certified fraction, the fraction of selected candidates certified, and witness/applied-input failures answer different questions. The offset-input negative control demonstrates why witness feasibility cannot be transferred to an arbitrary applied input. Likewise, a zero failure count with no passing candidate has an undefined denominator rather than indicating a useful certificate. Robustness to disturbances is assessed jointly with certificate availability, not through the failure rate alone.

Prefix, suffix and fallback tests. The horizon test compares states 0:3 and 0:8 for the same held candidate. Suffix loss is conditioned on passage of the three-step prefix. The fallback test compares the original minimum-burden choice with the common base anchor at the same snapshot and with matched branch disturbances. The task-cost contrast is selected minus anchor; positive values indicate degradation. Conditioning uses an empty *soft-admission* set, not an empty independent certificate set or an already infeasible current QP.

Table S72: prefix and suffix analysis for selected candidates. Prefix states are 0-3 and follow-up states 0-8; loss is conditioned on prefix passage. These are snapshot windows, not the whole episodes in Tables S63-S64. The displayed outcomes are from the analytical-control cohort.

Setting	Cohort	3-step pass	8-step pass	Suffix loss / prefix	Loss (%)
Medium	test	484/500	134/500	350/484	72.31
Medium	stress	409/500	31/500	378/409	92.42
Hard	test	423/500	55/500	368/423	87.00
Hard	stress	310/500	3/500	307/310	99.03

Table S73: same-snapshot fallback comparison against the base anchor. Empty denotes an empty soft-admission set. Positive cost difference is worse; confidence intervals resample snapshots. A zero conditional denominator is undefined.

Setting	Cohort	Empty / all	Worse / empty	Mean delta	95% CI
Medium	test	0/500	undefined	—	undefined
Medium	stress	2/500	0/2	-0.145	[-0.220,-0.069]
Hard	test	53/500	40/53	0.173	[0.120,0.227]
Hard	stress	58/500	43/58	0.177	[0.117,0.236]

Short-horizon risk and task tradeoff. In the Hard test panel, suffix loss occurs in 368/423 prefix-passing cases. The empty-set branch has 40/53 task-worse cases and mean cost increase 0.173. These outcomes limit the respective eight-step and fallback nonworsening claims. The Medium test panel contains no empty-set case, so its conditional fallback quantities are N/A. Neither a favorable average nor a current feasible input establishes recursive feasibility.

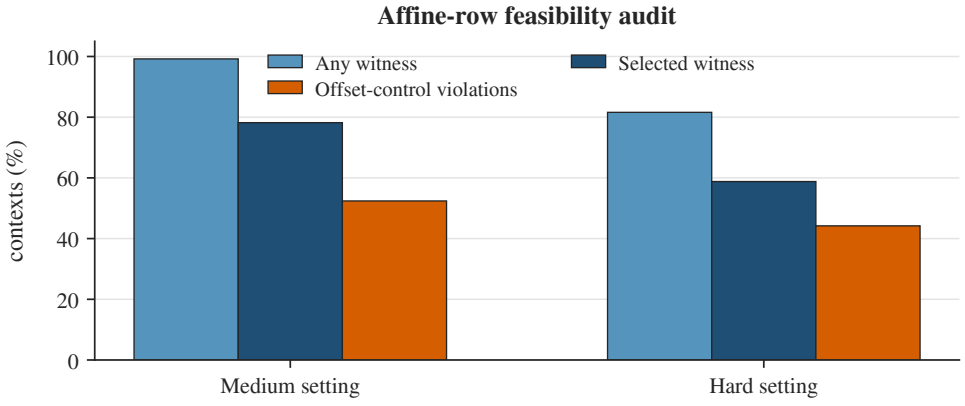

Figure S16: analytical-control outcomes: witness availability, selected-witness certification and the offset applied-input test. The controls separate existence and execution; they are not full UAV constraint tests.

Candidate-count and horizon sensitivity. The common-snapshot comparison varies a nested candidate prefix and the scoring horizon with shared state and branch realizations. Candidate counts are $K \in \{4, 8, 16, 32\}$ and horizons are $H \in \{1, 3, 5, 8\}$. The default $(K, H, \tau_P, \lambda_P) = (8, 3, 0.6, 1)$ is shared with the audit. Tables S74–S75 give snapshot outcomes . Work KH is reported explicitly; these rows are not an equal-work comparison. A separate closed-loop sensitivity protocol regenerates candidates and rolls out whole episodes under each setting.

Table S74: common-snapshot sensitivity in Medium. Outcomes use 500 snapshots. Any 8 denotes candidate availability; Selected 8 denotes the selected outcome. Work KH makes budget changes explicit.

Knob	Value	K H	Empty	Selected 3	Selected 8	Any 8
K	4.00	12	10	478	178	340
K	8.00	24	0	484	134	416
K	16.00	48	0	480	107	472
K	32.00	96	0	479	93	493
H	1.00	8	0	319	102	416
H	3.00	24	0	484	134	416
H	5.00	40	7	488	178	416
H	8.00	64	79	486	340	416
tau	0.30	24	0	484	134	416
tau	0.60	24	0	484	134	416
tau	1.20	24	0	484	134	416
lambda	0.50	24	0	484	132	416
lambda	1.00	24	0	484	134	416
lambda	2.00	24	0	485	143	416
w_margin	0.30	24	0	484	134	416
w_margin	0.60	24	0	484	134	416
w_margin	1.20	24	0	484	134	416
w_infeas	0.50	24	0	484	134	416
w_infeas	1.00	24	0	484	134	416
w_infeas	2.00	24	0	484	134	416

Candidate availability versus selected performance. "Any 8" is the event that at least one candidate passes the eight-step test; "Selected 8" concerns the candidate chosen by the short-horizon selector. For nested candidate sets and a fixed test, availability cannot decrease. Selected performance can decrease because the selector optimizes a different horizon. The two counts are reported together to distinguish proposal coverage from selection error.

Table S75: common-snapshot sensitivity in Hard. Outcomes use 500 snapshots. Horizon and score settings change one at a time; memory-dependent generation belongs to the separate closed-loop protocol.

Knob	Value	K H	Empty	Selected 3	Selected 8	Any 8
K	4.00	12	119	370	77	204
K	8.00	24	53	423	55	266
K	16.00	48	23	439	38	320
K	32.00	96	6	451	27	362
H	1.00	8	2	189	25	266
H	3.00	24	53	423	55	266
H	5.00	40	122	418	82	266
H	8.00	64	224	424	217	266
tau	0.30	24	53	423	55	266
tau	0.60	24	53	423	55	266
tau	1.20	24	7	419	55	266
lambda	0.50	24	53	423	51	266
lambda	1.00	24	53	423	55	266
lambda	2.00	24	53	423	53	266
w_margin	0.30	24	53	423	55	266
w_margin	0.60	24	53	423	55	266
w_margin	1.20	24	53	423	55	266
w_infeas	0.50	24	22	423	55	266
w_infeas	1.00	24	53	423	55	266
w_infeas	2.00	24	53	423	55	266

Horizon-dependent selection. For Hard at $H = 3$, the eight-step availability rises from 204/500 to 362/500 as K increases from 4 to 32, whereas the selected eight-step count falls from 77/500 to 27/500. At $K = 8$, increasing H from 3 to 8 raises the selected eight-step count from 55/500 to 217/500, with greater work and more empty soft sets. The comparison identifies the distinction between more candidates and a selection objective that captures their relevant future behavior.

Weights and memory. The score sweep uses $\tau_P \in \{0.3, 0.6, 1.2\}$, $\lambda_P \in \{0.5, 1, 2\}$, margin weight $\{0.3, 0.6, 1.2\}$ and infeasibility weight $\{0.5, 1, 2\}$, changing one setting at a time. Rescoring a fixed candidate pool tests the selector. EMA values $\{0, 0.5, 0.8, 0.95\}$ instead belong to the closed-loop protocol, since they change the generator's input and subsequent states. Time, intervention, raw cost and empty-set frequency accompany feasibility to distinguish responsiveness from excessive conservatism.

Y.3 Language interface, deployment cost and behavior transfer

Language setting and metric. The interface test covers progress, obstacle clearance, pairwise separation and their conjunctions within the four-parameter action space. Literal instructions, unseen paraphrases and compositions are split by parent template family; state-fixed instruction changes and instruction-fixed state changes are separate tests. LLM and Text-MLP receive text plus the same numeric context. Text-MLP uses the deterministic encoder in Protocol P3; the formal-rule oracle receives the intended predicates rather than the text. Its privileged input makes it a reference for intent disambiguation, not an equal-information arm. Compliance is evaluated on the first-ranked proposal using fixed predicates.

Table S76: first-proposal instruction compliance. The LLM and Text-MLP arms use text plus numeric context; the formal-rule oracle receives intent predicates. Protocol P3 specifies the encoding, split and compliance test.

Arm	Form	Correct / n	Rate (%)	Input
LLM	literal	224/240	93.3	Text + context
LLM	paraphrase	213/240	88.8	Text + context
LLM	composition	179/240	74.6	Text + context
Text-MLP	literal	223/240	92.9	Text + context
Text-MLP	paraphrase	143/240	59.6	Text + context
Text-MLP	composition	125/240	52.1	Text + context
Rule-formal-oracle	literal	237/240	98.8	Oracle intent
Rule-formal-oracle	paraphrase	232/240	96.7	Oracle intent
Rule-formal-oracle	composition	237/240	98.8	Oracle intent

Compliance across instruction forms. The LLM counts are 224/240, 213/240 and 179/240 for literal, paraphrased and composed instructions. The Text-MLP counts are 223/240, 143/240 and 125/240. The displayed gap is smallest for literal instructions and larger for the other forms. The oracle's input advantage is considered separately. Tuple-level compliance is not closed-loop mission success; neither metric alone establishes support for goals outside the four-parameter interface.

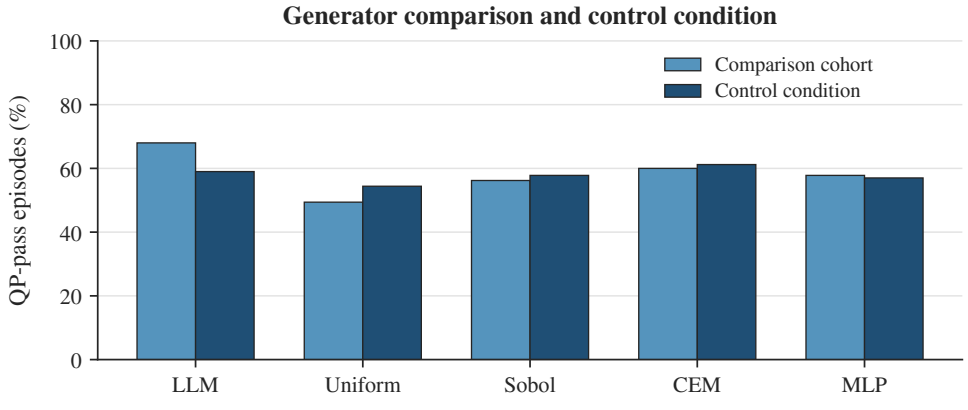

Figure S17: Generator comparison from Section Y.1, alongside the control condition.

Runtime and cost measurement design. One high-level planning event comprises generation, all candidate QPs, parsing, bookkeeping and communication. Event-level components and retries are summed before computing quantiles; component medians or 95th percentiles are not added together. The independent audit adds 8×4 state checks to the original 8×3 predictive transitions and is timed separately. Cloud and local are distinct deployment conditions, with model identifier, hardware, precision and transport settings fixed within each condition. Costs include all charged requests, including retries.

Table S77: event-level deployment times in seconds. Total time includes generation, evaluation, parsing and bookkeeping; quantiles are computed after summing each event. Deadline: $8T_s = 0.44$ s.

Mode/setting		Gen median	Eval median	Total median	Total p95	Miss / 500
cloud	/Medium	0.670	0.032	0.710	1.131	481/500
cloud	/Hard	0.657	0.054	0.727	1.218	490/500
local	/Medium	0.075	0.032	0.119	0.177	15/500
local	/Hard	0.076	0.054	0.143	0.205	18/500

Table S78: Additional-audit overhead and cost. Auditing adds 8 × 4 state checks to 8 × 3 prediction transitions. Token totals include retries; costs use the billing coefficients.

Mode/setting		With audit p95	Miss / 500	Input tokens	Output tokens	Total USD
cloud	/Medium	1.188	498/500	1128595	388347	0.2682
cloud	/Hard	1.321	500/500	1119020	387291	0.2668
local	/Medium	0.238	18/500	1138584	388082	0.0000
local	/Hard	0.304	20/500	1142788	392914	0.0000

Table S79: centralized communication comparison: 14 UAVs, 250 messages per UAV and 128 bytes per message. Each condition contains 500 episodes. Delay and state age are in seconds; QP count is an episode outcome.

Delay (s)	Drop p	Sent	Dropped	Age p95	QP count
0.00	0.00	1750000	0	0.051	351/500
0.05	0.01	1750000	17963	0.102	327/500
0.10	0.05	1750000	87273	0.153	277/500

Deadline and centralized communication. The deadline is 8Ts = 0.44 s. Before the independent audit, the cloud rows miss it in 481/500 Medium and 490/500 Hard events; the local rows also contain deadline misses. These timing results do not establish a hard real-time bound. Billing coefficients determine provider costs; zero provider fees for local inference exclude hardware and energy costs. Communication uses fourteen state messages per control step at 128 bytes per message. Attempted messages, drops, state age and control outcomes have distinct units. This is centralized coordination. New plans are checked at the current release state; timely delivery alone does not preserve an expired certificate.

Behavior-transfer design. The transfer comparison changes only the nominal task controller, sharing input bounds, CBF rows, four-parameter candidate interface and selector across arms. Flocking combines centroid cohesion and velocity alignment with gains 0.4 , 0.8 and reference velocity $(0.5 \quad , 0 \quad , 0 \quad)$ m/s. Formation tracks fourteen ring offsets of radius 2.5 m around $c(t) = (5{+}\sin(0.15t), 5, 3.2)$ m using the source tracking gains. Offsets are assigned once before the trial and are not re-optimized on test trajectories. Protocol P4 defines the physical metrics, observation window and treatment of incomplete episodes.

Table S80: episode-level behavior-transfer comparison. QP, Safe and Joint are counts out of 500 episodes; Joint is the intersection. Physical task-error definitions are given separately in Protocol P4.

Behavior	Arm	QP	Safe	Joint
Flocking	Reactive	212/500	309/500	210/500
Flocking	Feedback	306/500	363/500	299/500
Flocking	Predictive	394/500	416/500	389/500
Formation	Reactive	225/500	315/500	220/500
Formation	Feedback	313/500	363/500	308/500
Formation	Predictive	395/500	417/500	388/500

Episode-level transfer comparison. The Predictive QP counts are 394/500 for flocking and 395/500 for formation. QP and safety counts are paired within each episode, so their reported intersection is evaluated directly. This panel compares constraint outcomes. Task performance uses velocity alignment and mean-velocity tracking errors in m/s for flocking, and fixed-reference tracking error in m for formation. The two physical units are not combined into a common score; alignment alone is not treated as evidence of progress because stopped vehicles can be perfectly aligned.

Joint interpretation. The common-evaluator study separates proposal quality from predictive selection; the information ablations isolate feedback and history. Prediction, witness and applied-input checks then delimit which safety statement is being tested. The suffix, fallback and deadline panels retain unfavorable outcomes rather than turning a conditional improvement into an unconditional guarantee. The language and behavior protocols address two additional task interfaces while retaining the original low-level safety problem.

Protocol P1: shared numeric input and data separation. The following settings fix the comparison interface. Define the componentwise-normalized context

$$x_n = \big[(\bar{p}_n - p_{\min})/(p_{\max} - p_{\min}),\ \bar{v}_n/v_{\max},\ \max_i \|v_{i,n}\|/v_{\max},$$
$$m_n^{\text{obs}}/s_{\text{obs}}, m_n^{\text{pair}}/s_{\text{pair}}, m_n^{\text{bdry}}/s_{\text{bdry}}, m_n^{\text{vel}}/s_{\text{vel}},$$
$$\widetilde{r}_n,\ a_n,\ \widetilde{s}_n,\ q_n,\ f(P_n), f_s(P_n - P_{n-8}), f_s(\widetilde{r}_n - \widetilde{r}_{n-8})\big] \in \mathbb{R}^{18}.$$

Here $\bar{p}, \bar{v}$ are team means, $f(t) = t/(1+t)$ for $t \geq 0$, and $f_s(t) = t/(1+|t|)$. The fixed scales are $s_{\text{obs}} = s_{\text{pair}} = 1\,\text{m}^2$, $s_{\text{bdry}} = 1\,\text{m}$, and $s_{\text{vel}} = v_{\max}^2$. The intervention feature is $\widetilde{r} = \|u_{\text{applied}} - u_{\text{nom}}\|/(\sqrt{3N}u_{\max})$; $\widetilde{s}$ is the RMS of rowwise dimensionless slack with positive fixed row-unit scales. This encoder does not redefine the source burden score. All four constraint families are present in the comparison. Before eight observations exist, the first observation supplies the history boundary. An unresolved numerical hard check is recorded separately and invalidates the common snapshot for every arm; its frequency is reported outside the valid snapshot denominator.

All common-information generators receive these eighteen values. LLM additionally receives only their fixed descriptions and the shared task instruction. The MLP normalizer is fitted on training scenes; constant features use scale one. No-feedback masks the eleven diagnostic coordinates. No-memory zeroes the last two coordinates and replaces P_n by instantaneous burden. Scalar retains only $f(P_n)$ among diagnostic coordinates. Parent scene/layout IDs remain within one training, validation or test split. The teacher searches 254 proposals plus two anchors per training context and selects six by the shared soft eligibility/score rule and stable candidate ID. Training minimizes MSE on the ordered normalized 24-vector. Test scenes do not select the normalizer, teacher settings or validation epoch.

Protocol P2: bounded search baselines. Search operates on $y \in [0,1]^4$, mapped affinely to the parameter box. The Rule arm uses $c_f = [0.25 - m_f/s_f]_+$ and $d = (-\min(1, \max_f c_f), \min(1, c_{\text{obs}}), \min(1, c_{\text{pair}}), \min(1, c_{\text{pair}})(1 - \min(1, c_{\text{bdry}})))$. Its six proposals are the clipped normalized base plus $\{0.15d, 0.30d, 0.15e_1, -0.15e_1, 0.15e_2, 0.15e_3\}$, where e_j is the jth unit vector; common anchors are additional counted slots. Uniform uses six independent box draws. Grid uses $(.25, .25, .25, .25)$, $(.75, .75, .75, .75)$, $(.25, .75, .5, .5)$, $(.75, .25, .5, .5)$, $(.5, .5, .25, .75)$ and $(.5, .5, .75, .25)$, with coordinates cyclically permuted by snapshot index modulo four. Sobol takes the first six points of a separately seeded scrambled eight-point block; the other two are not evaluated.

CEM starts from the base mean and diagonal variance $0.08I$, sampling three box points by rejection per round. It updates from the two best evaluated archive entries, including anchors, using smoothing 0.5 for the mean and diagonal variance and floor 0.01. Adaptive uses three uniform draws followed by three clipped Gaussian perturbations of the best evaluated entry, with normalized standard deviation 0.15. Archives rank eligible candidates by task–burden score and ineligible candidates by burden, with eligible entries first and candidate ID breaking ties. After 1,000 failed rejection draws a slot is invalid. Invalid outputs, duplicates and solver failures consume slots; no arm receives a free replacement or final-mean evaluation.

Protocol P3: language-policy input and compliance. Text-MLP has a separate text input from the numeric MLP. Normalize text by Unicode NFC, lower case and collapsed whitespace. Count character 3-, 4- and 5-grams; hash each UTF-8 gram using SHA-256 with prefix `HC8|`. The first eight digest bytes choose one of 256 bins and the next byte chooses its sign. L2-normalize the vector, or use zeros for an empty vector, and concatenate the eighteen numeric features. The network is $274 \to 64 \to 64 \to 24$ (23,320 parameters), with ReLU hidden layers and sigmoid/box output mapping. Optimizer, training seeds and validation-only selection match the numeric arm; the input dimensions and model parameters of the two arms are distinct.

Split intent/template families before creating paraphrases. Texts and contexts from one parent stay in one split. The teacher search receives the training intent. The LLM and Text-MLP test interfaces receive text and numeric context; the formal-rule oracle additionally receives the intent predicates. On normalized y, compliance predicates are progress $y_1 \geq 0.65$, obstacle clearance $y_2 \geq 0.65$, and pair separation $\max(y_3, y_4) \geq 0.65$; a conjunction requires all specified predicates. Table S76 concerns the first-ranked proposal. Full-set and post-selector compliance have separate definitions and are not substituted for that metric. Requests outside the four-parameter action space are rejected and counted separately. These are interface-compliance thresholds, not CBF safety conditions.

Protocol P4: behavior controllers and physical errors. A trial contains $T = 250$ control intervals and states 0:250. The reporting window is $\mathcal{W} = \{201, \ldots, 250\}$. For flocking, $v_{\mathrm{ref}} = (0.5, 0, 0)$ m/s; report

$$E_{\mathrm{align}} = \sqrt{\frac{1}{N|\mathcal{W}|} \sum_{n \in \mathcal{W}} \sum_{i} \|v_{i,n} - \bar{v}_n\|^2} \quad [\mathrm{m/s}],$$

$$E_{\mathrm{track}} = \sqrt{\frac{1}{|\mathcal{W}|} \sum_{n \in \mathcal{W}} \|\bar{v}_n - v_{\mathrm{ref}}\|^2} \quad [\mathrm{m/s}].$$

For formation, use $d_i = 2.5(\cos(2\pi i/N), \sin(2\pi i/N), 0)$ m, $c(t) = (5 + \sin(0.15t), 5, 3.2)$ m, and

$$E_{\mathrm{form}} = \sqrt{\frac{1}{N|\mathcal{W}|} \sum_{n \in \mathcal{W}} \sum_{i} \|p_{i,n} - c(t_n) - d_i\|^2} \quad [\mathrm{m}].$$

Offsets are matched once at initialization by minimum distance with stable ties; the assignment is fixed through the trial. Flocking nominal acceleration is $0.4(\bar{p} - p_i) + 0.8(\bar{v} - v_i) + 0.8(\gamma v_{\mathrm{ref}} - v_i)$. Formation uses source k_p, k_v tracking of $c(t) + d_i$ and its reference velocity. Obstacle/pair/altitude shaping, input limits and safety filtering are shared. Initialization records rejection counts and failed layouts.

Each complete episode supplies one physical error value; aggregation is over episodes. All-episode and successful-episode summaries have explicit denominators. Early termination retains a task failure and a missing full-window error, without zero padding. Completion fraction, time, mean speed and minimum pair distance complement errors and safety counts. Stopping can reduce alignment error, which is why E_{track} is also required. No aggregation mixes errors in m and m/s.

Protocol P5: conditions and measurements for feasibility auditing. The audit applies the existing Appendix B theorem. It holds the candidate fixed through the checked prefix and fixes the complete modeled obstacle, pair, boundary and speed rows, input limits, row gains and integration scheme. The snapshot includes the joint state, controller memory, predictor state and exogenous realization. The exact-model arm uses equal initial states and zero process mismatch without presuming identical future filtered inputs. The bounded-disturbance arm has componentwise initial position/velocity bounds 0.005 m and 0.01 m/s and per-step position/velocity increment bounds 0.0005 m and 0.002 m/s. Multiply each componentwise three-vector bound by $\sqrt{3}$ to obtain a Euclidean radius. The stress arm doubles injected bounds but retains the nominal audit budgets, so it tests behavior outside the stated bound event rather than enlarging the theorem's guarantee.

Without assuming a smooth QP solution map, a valid input-difference bound for agent i is

$$D_{i,\ell} = \left[\sum_{d=1}^{3} (u_{\max} + |\widehat{u}_{i,\ell,d}|)^2\right]^{1/2}.$$

It enters the Appendix B position/velocity recursion. At each sampled state, compute every row bound and its observed error. Static obstacle/pair gains are factored as $\kappa_1 = 1.2$, $\kappa_2 = 2.5$ in this arm. Initial $h \geq 0$ and $\dot{h} + \kappa_1 h \geq 0$ are checked before invoking any continuous-time HOCBF statement. First- and second-order boundary implementations are recorded separately. New obstacle rows or pruning require new enclosures; a moving density field is not a moving obstacle model.

A primal witness must satisfy input bounds and, for every row,

$$r_j(\widehat{z}, \bar{u}) - \eta_j - \epsilon_j^{\mathrm{num}} \geq 0.$$

Positive row-unit scales and numerical budgets are fixed before evaluation. An unresolved residual sign or roundoff enclosure is classified as unknown; a solver status or tolerance alone is not a verified arithmetic enclosure. The soft-admitted set, the independently certified set and the actually applied input are recorded separately. Report certificate availability, selected-plan coverage, bound exceedances and witness/applied-input violations with their respective denominators. A zero conditional denominator yields N/A.

Scope of the tests. The finite-horizon result covers the specified states $0{:}H$. It does not infer an eight-step guarantee from a three-step check, guarantee a nonempty certified set, or establish recursive feasibility or task-nonworsening fallback behavior. Sampled-state feasibility and intersample safety are separate properties. Accordingly, the horizon, fallback and deployment panels measure those additional outcomes directly rather than treating them as consequences of the certificate.