BEYOND A SINGLE SAFETY HORIZON: MULTI-HORIZON POLICY COORDINATION FOR OFFLINE SAFE REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Safe offline reinforcement learning aims to maximize task performance under safety constraints using only static datasets. Existing approaches typically deploy a single policy, collapsing potentially different temporal safety preferences into one behavior. However, safety violations can unfold over substantially different time scales, such that changing the safety horizon alters which future hazards are visible when evaluating an action and can induce distinct safety-oriented behaviors. This motivates us to exploit temporal safety specialization rather than compressing multiple safety scales into a single policy. We propose **Multi-Horizon Safety Policy Coordination (MSPC)**, which learns and coordinates a family of safety-first policies specialized to different temporal horizons. At the low level, safety supervision over multiple future horizons is used to extract complementary, behavior-supported policies from the same offline dataset. At the high level, shared reward and cost critics evaluate their candidate actions, and a residual-budget coordinator selects among them based on the remaining safety allowance. We further use policy disagreement to avoid redundant coordination when candidate actions remain similar. Experiments on 18 tasks from two offline safe RL benchmark suites show that the evaluated fixed specialists exhibit non-nested safety outcomes, highlighting the need for adaptive coordination among complementary temporal safety behaviors. MSPC further achieves favorable reward–safety performance against representative offline safe RL baselines, demonstrating the effectiveness of temporal safety specialization and budget-aware action-level coordination.

1 INTRODUCTION

Safe offline reinforcement learning (RL) (Levine et al., 2020) aims to learn high-performing policies under safety constraints using only static datasets, avoiding risky online interaction in domains such as autonomous driving and energy-system control (Wu et al., 2024; Zhang et al., 2025b; Yu et al., 2025; Wang et al., 2025). Typically formulated as a constrained Markov decision process (CMDP) (Altman, 2021), it requires maximizing long-term return while satisfying a cumulative safety budget under limited offline coverage (Xu et al., 2022; Hong et al., 2024). A key but underexplored difficulty is that safety consequences can emerge at different temporal scales: some actions lead to violations within a few steps, whereas others appear locally benign but expose the agent to hazards only much later (Dalal et al., 2018; Yu et al., 2022).

Representative safe offline RL methods incorporate safety through constrained optimization (Le et al., 2019; Lee et al., 2022) or safety-aware policy extraction (Koirala et al., 2024; Zheng et al., 2024). Recent advances include goal-assisted stitching (Zifan et al., 2026), reachability-guided one-step policy extraction (Tayal et al., 2026), and joint long- and short-term cost learning (Li et al., 2026). We study a complementary question: *how does the temporal scope of safety supervision shape the policies learned from offline data?* Extending a supervision window changes which future violations in logged trajectories are considered when assessing a state–action pair and can alter the relative weighting of dataset actions during policy extraction. We refer to the resulting horizon-dependent behaviors as *temporal safety specializations*. Fig. 1 qualitatively illustrates how local and global safety assessments can differ.

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.

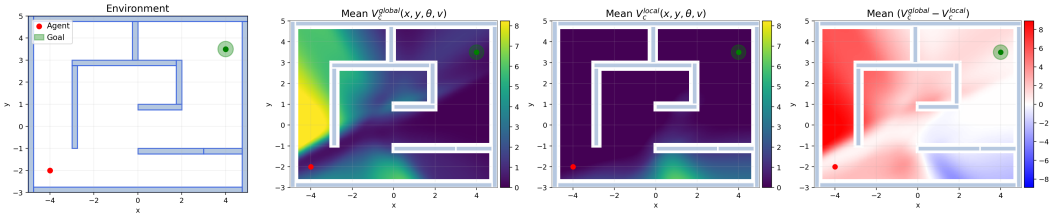


Figure 1: **Different temporal safety signals can induce different safety regions.** A local safety signal preserves a broader short-range feasible region, whereas a global signal incorporates more distant consequences and induces a distinct, typically more restrictive region. This illustrates that the notion of desirable safe behavior depends on the temporal scope over which future violations are considered.

Broader supervision, however, need not yield safer closed-loop behavior. Although finite-window violation labels are nested as the horizon increases, the extracted policies need not inherit this ordering: horizon-dependent action weighting can change action preferences and deployment-time state distributions. Our experiments reveal non-nested safe-task sets across the evaluated temporal specialists, motivating their coordination rather than commitment to a single horizon.

We propose **Multi-Horizon Safety Policy Coordination (MSPC)**, which trains safety-first specialists from the same offline dataset using finite-window and recursively propagated Global feasibility targets, without incorporating reward into policy extraction. At deployment, shared reward and cost critics evaluate their current action proposals, and a residual-budget coordinator selects among them. Action disagreement controls when this selection is refreshed. This separates the construction of temporal safety behaviors from their budget-aware use, without requiring specialist-specific long-term value estimates. Empirically, coordination achieves broader safe-task coverage than the union of the fixed specialists' safe-task sets, supporting the usefulness of temporal complementarity.

Our main contributions are as follows:

- We introduce *temporal safety specialization* to learn distinct safety-oriented policies across supervision horizons. We show empirically that broader supervision need not yield safer closed-loop behavior.
- We propose MSPC for budget-aware action-level coordination of temporal specialists, using shared reward and cost critics with disagreement-triggered refresh.
- Experiments on 18 tasks across two benchmark suites demonstrate competitive reward–safety performance and broader safe-task coverage than the union of fixed specialists' safe-task sets.

2 Related Work

Safe Offline Reinforcement Learning. Safe offline RL aims to learn constraint-satisfying policies from static datasets. Existing methods address offline safety constraints through value learning and distribution correction (Xu et al., 2022; Lee et al., 2022), or sequence and generative modeling (Liu et al., 2023b; Zhang et al., 2023; Zheng et al., 2024; Yao et al., 2024). ARMOR (Li et al., 2026) learns long- and short-term cost critics and uses safety-aware reweighting to train a budget-conditioned diffusion actor. CAPS (Chemingui et al., 2025) constructs and coordinates policies with different reward–cost preferences under a safety budget. MSPC differs in the source and use of policy diversity: horizon-dependent violation supervision shapes separate safety-only specialists from the same offline dataset and retains them for budget-aware action-level coordination.

Multi-Horizon Learning and Temporal Abstraction in Reinforcement Learning. Hierarchical RL studies temporally extended behaviors and their coordination (Sutton et al., 1999; Dietterich, 2000; Nachum et al., 2018). Multi-timescale methods study value estimation across horizons (Romoff et al., 2019; Fedus et al., 2019), while the effective horizon also influences regularization and pessimism in offline RL (Hu et al., 2022). Hierarchical structures have also been explored in safe RL through safety-aware subgoals, barrier-based controllers, and safety-critical planning (Ahmad et al., 2025;

Xiong et al., 2022; Li et al., 2021; Zhang et al., 2025a). Our use of multiple horizons differs from these temporal abstractions in both purpose and execution semantics. In MSPC, a safety horizon specifies how far into the future violation information is considered when constructing safety supervision, thereby inducing different temporal safety specializations from the same dataset.

3 PRELIMINARIES

3.1 CONSTRAINED MARKOV DECISION PROCESS

We consider a discounted constrained Markov decision process (CMDP) defined by the tuple $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, r, c, \gamma, B)$, where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $P(s' \mid s, a)$ is the transition kernel, $r(s, a)$ is the reward function, $c(s, a) \geq 0$ is the instantaneous safety cost, $\gamma \in (0, 1)$ is the discount factor, and B is the safety budget (Altman, 2021). A policy $\pi(a \mid s)$ induces the discounted reward and cost objectives

$$J_r(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right], \qquad J_c(\pi) = \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t c(s_t, a_t) \right], \tag{1}$$

and safe RL seeks to solve

$$\max_\pi J_r(\pi) \qquad \text{s.t.} \qquad J_c(\pi) \leq B. \tag{2}$$

In the offline setting, the policy must be learned entirely from a static dataset $\mathcal{D}$ collected by an unknown behavior policy. We assume that the dataset retains trajectory ordering, such that each trajectory can be written as $\tau = (s_0, a_0, r_0, c_0, \ldots, s_T)$; this temporal structure will later be used to construct horizon-dependent safety supervision.

The cumulative cost c serves two roles that should be distinguished. First, it defines the original CMDP constraint and is used for budget accounting during deployment. Second, its occurrence indicates whether a transition violates the safety constraint. To express the latter in a form suitable for feasibility learning, we define a signed safety signal (Zheng et al., 2024)

$$h(s, a) = \begin{cases} h_+, & c(s, a) > 0, \\ h_-, & c(s, a) = 0, \end{cases} \qquad h_+ > 0,\ h_- < 0, \tag{3}$$

and write $h_t = h(s_t, a_t)$. Positive values therefore denote constraint violations, while negative values denote non-violating transitions. Importantly, h does not replace the original CMDP cost: c and B retain the cumulative budget semantics of the constrained objective, whereas h is used only to construct feasibility-oriented learning targets. The specific values of h_+ and h_- used in our experiments are provided in the implementation details.

3.2 FEASIBILITY VALUE LEARNING

Feasibility-guided offline RL uses safety values to distinguish dataset actions according to their future constraint outcomes and to bias policy extraction toward safer behavior (Zheng et al., 2024). We denote the corresponding action- and state-level feasibility values by $Q_f(s, a)$ and $V_f(s)$, respectively, where a larger value represents a less desirable safety outcome under the signed convention above. We use the subscript f to distinguish these feasibility values from the cumulative cost critic Q_c used later for budget-aware candidate evaluation.

Following the value-learning principle of Implicit Q-Learning (IQL) (Kostrikov et al., 2021), the state value is fitted to feasibility values on dataset actions through expectile regression,

$$\mathcal{L}_{V_f}(\psi) = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[L_2^\tau \left(Q_{f,\hat{\theta}}(s, a) - V_{f,\psi}(s) \right) \right], \tag{4}$$

where

$$L_2^\tau(u) = |\tau - \mathbb{1}(u < 0)|\, u^2. \tag{5}$$

The feasibility action critic is trained by regression toward a target y_f,

$$\mathcal{L}_{Q_f}(\theta) = \mathbb{E}_{(s,a)\sim\mathcal{D}} \left[\left(Q_{f,\theta}(s, a) - y_f \right)^2 \right]. \tag{6}$$

The target y_f determines the temporal meaning of the feasibility estimate. MSPC uses finite-window and Global targets, defined in Sec. 4.1, and extracts the corresponding safety-first policies using the common objective in Sec. 4.2.

4 METHODOLOGY: MULTI-HORIZON SAFETY POLICY COORDINATION

There are two main stages for the proposed MSPC framework. We first construct a family of safety-first policies whose safety preferences are specialized to different temporal horizons. At deployment, these policies propose candidate actions that are coordinated according to shared reward–cost estimates and the remaining safety budget. The key distinction is that the safety horizon determines the temporal scope of offline safety supervision rather than the duration for which a policy is executed.

4.1 MULTI-HORIZON FEASIBILITY SUPERVISION

Our starting point is that the safety assessment of a logged state–action pair depends on how far its future consequences are considered. Using the signed safety signal h_t defined in Sec. 3.1, we construct horizon-dependent feasibility targets directly from the trajectory structure of the offline dataset. For a finite safety horizon H, we define

$$y_t^H = \max_{0 \le j < \min(H, T-t)} h_{t+j}, \tag{7}$$

where T denotes the terminal time of the trajectory containing (s_t, a_t). Since $h_+ > 0$ denotes a constraint violation and $h_- < 0$ denotes a non-violating transition, y_t^H is positive whenever a violation occurs within the following H transitions and remains negative otherwise. Therefore, increasing H changes which delayed violations are attributed to the current state–action pair rather than merely changing the numerical scale of the same cost target.

We consider a set of finite safety horizons $\mathcal{H} = \{H_1, \dots, H_K\}$. For each $H \in \mathcal{H}$, we learn a horizon-specific feasibility pair (Q_f^H, V_f^H) using the IQL-style procedure introduced in Sec. 3.2, with Q_f^H regressed toward the target in Eq. 7. Because the targets are constructed directly from observed trajectory suffixes, the finite-horizon branches do not require model rollouts or counterfactual execution of the learned policies.

Finite windows capture safety information only within their predefined temporal scope. To complement them with a long-range safety specialization, we additionally introduce a Global branch, denoted by G, using the same signed safety signal h_t with the recursive target

$$y_t^G = (1-\gamma)h_t + \gamma(1-d_t)\max\left(h_t, V_f^G(s_{t+1})\right). \tag{8}$$

The complete set of temporal safety indices K is denoted as

$$\mathcal{K} = \mathcal{H} \cup \{G\}. \tag{9}$$

The finite branches use observed trajectory windows, while Global recursively propagates long-range feasibility. Extending a finite window changes labels exactly for anchors whose next logged violation lies in the newly included interval (Proposition B.1). Very broad windows may also saturate binary violation labels (Corollary B.2). These targets therefore provide different temporal views of safety, rather than progressively more accurate estimates of a single value. Each view is retained to train a separate specialist.

4.2 MULTI-HORIZON SAFETY-FIRST POLICY SPECIALIZATION

Given the temporal feasibility estimators $\{(Q_f^k, V_f^k)\}_{k\in\mathcal{K}}$, we extract one safety-first policy for each temporal specialization. We use twin action critics and take the conservative estimate

$$\bar{Q}_f^k(s,a) = \max\left\{Q_{f,1}^k(s,a), Q_{f,2}^k(s,a)\right\}, \tag{10}$$

where larger feasibility values correspond to less desirable safety outcomes under the signed convention defined in Sec. 3.1. We then define the horizon-specific feasibility advantage as

$$A_f^k(s,a) = \bar{Q}_f^k(s,a) - V_f^k(s). \tag{11}$$

An action with $A_f^k(s,a) > 0$ is less favorable than the state-level feasibility reference under temporal specialization k, whereas a negative advantage indicates a relatively preferable safety outcome.

We extract each policy through feasibility-weighted behavior cloning,

$$w_k(s,a) = \min\left\{\exp\left(-\alpha A_f^k(s,a)\right), w_{\max}\right\}, \tag{12}$$

and optimize

$$\mathcal{L}_{\pi_k} = -\mathbb{E}_{(s,a)\sim\mathcal{D}}\left[w_k(s,a)\log\pi_k(a \mid s)\right]. \tag{13}$$

All specialists use the same feasibility-weighted cloning objective without a reward term; specialization is induced by their different temporal safety targets. Under the idealized extraction model of Proposition B.3, each specialist reweights the same behavior distribution using its horizon-specific action weights. Different relative action weights can therefore produce different policies from the same data. These policy differences can alter closed-loop state distributions, so nested labels alone do not establish a policy-level safety ordering. We retain the family $\Pi = \{\pi_k\}_{k\in\mathcal{K}}$ for action-level coordination rather than aggregating the targets into one actor. The safety horizon specifies supervision, not execution duration.

4.3 Budget-Aware Action-Level Coordination

The multi-horizon policy family provides complementary safety-oriented behaviors, but no individual specialization is assumed to be uniformly preferable. MSPC therefore coordinates the specialists at the level of their current action proposals. Given a state s_t, each policy $\pi_k \in \Pi$ proposes a candidate action

$$a_t^k = \pi_k(s_t), \qquad k \in \mathcal{K}. \tag{14}$$

Rather than learning a separate long-term value for each policy identity, we evaluate all candidates using a shared pair of action-conditioned reward and cost critics. This makes the coordination rule agnostic to the horizon index: the temporal semantics are encoded in the candidate actions produced by the specialized policies, while the coordinator only evaluates their predicted task value and budget consumption.

Shared reward–cost action critics. We learn shared reward and cost critics $Q_r(s,a)$ and $Q_c(s,a)$ from the offline dataset using IQL-style value learning (Kostrikov et al., 2021). Importantly, these critics take only the state–action pair as input and do not condition on the safety horizon, policy identity, or remaining budget. For $x \in \{r,c\}$, the corresponding state value is learned by expectile regression,

$$\mathcal{L}_{V_x} = \mathbb{E}_{(s,a)\sim\mathcal{D}}\left[L_2^{\tau_x}\left(\bar{Q}_x(s,a) - V_x(s)\right)\right], \tag{15}$$

where $\bar{Q}_x$ denotes the conservative twin-critic estimate. The action critics are updated using

$$y_x = x_t + \gamma(1-d_t)V_x(s_{t+1}), \qquad \mathcal{L}_{Q_x} = \sum_{i=1}^{2}\mathbb{E}_{(s_t,a_t,x_t,s_{t+1})\sim\mathcal{D}}\left[\left(Q_{x,i}(s_t,a_t) - y_x\right)^2\right]. \tag{16}$$

For candidate evaluation, we use the conservative scores

$$\widehat{Q}_r(s,a) = \min_{i\in\{1,2\}} Q_{r,i}(s,a), \qquad \widehat{Q}_c(s,a) = \max_{i\in\{1,2\}} Q_{c,i}(s,a), \tag{17}$$

so that reward is evaluated pessimistically while cost is evaluated conservatively. Because the same critics evaluate all candidate actions, MSPC avoids explicitly estimating the long-term counterfactual outcome of executing each learned specialist.

Residual-budget coordination. Let B denote the episodic safety budget and let $C_t = \sum_{i=0}^{t-1} c_i$ be the raw cost consumed before time t. The remaining budget is $b_t = \max(B - C_t, 0)$. Let $n_t = T - t$ denote the remaining episode length. To compare the residual episodic budget with the discounted cost-to-go estimated by Q_c, we convert b_t into the discounted allowance

$$\epsilon_t = b_t \frac{1-\gamma^{n_t}}{(1-\gamma)n_t}. \tag{18}$$

We first construct the predicted feasible candidate set

$$\mathcal{F}_t = \left\{k \in \mathcal{K} : \widehat{Q}_c(s_t, a_t^k) < \epsilon_t\right\}. \tag{19}$$

The coordinator then selects

$$k_t^* = \begin{cases} \arg\max_{k \in \mathcal{F}_t} \widehat{Q}_r(s_t, a_t^k), & \mathcal{F}_t \neq \emptyset, \\ \arg\min_{k \in \mathcal{K}} \widehat{Q}_c(s_t, a_t^k), & \mathcal{F}_t = \emptyset. \end{cases} \tag{20}$$

Thus, the cost critic defines which candidate actions are compatible with the remaining safety budget, while the reward critic discriminates among predicted feasible actions. This separation avoids collapsing reward and cost into a single fixed scalarization and allows the coordination rule to adapt directly to the residual budget.

Disagreement-triggered refresh. Re-evaluating all candidates with the shared critics at every time step is unnecessary when the temporal specialists propose nearly identical actions. We therefore use policy disagreement only to determine *when* the coordinator should be refreshed; disagreement itself does not determine which candidate is safe. We measure instantaneous disagreement by

$$D_t = \max_{i,j} \frac{\|a_t^i - a_t^j\|_2}{d_{\mathcal{A}}}, \tag{21}$$

where $d_{\mathcal{A}}$ is the diagonal length of the action box. We calibrate a threshold τ_D from the empirical disagreement distribution on offline states. When $D_t > \tau_D$, the residual-budget rule in Eq. 20 is recomputed. To capture persistent small deviations, when policy k_t remains active, we additionally accumulate

$$z_t = z_{t-1} + \delta_t, \qquad \delta_t = \max_j \frac{\|a_t^{k_t} - a_t^j\|_2}{d_{\mathcal{A}}}, \tag{22}$$

and trigger a refresh when z_t exceeds a calibrated cumulative threshold. The accumulator is reset after each coordination query. When no refresh is triggered, the currently selected specialist remains active but still generates a new action at the current state. Hence, the mechanism reduces redundant critic queries and rapid policy switching without imposing temporally extended execution of any horizon policy.

4.4 Practical Implementation

In all main experiments, we instantiate the finite safety-horizon set as $\mathcal{H} = \{8, 32, 128\}$ and augment it with the global feasibility branch, yielding $\mathcal{K} = \{8, 32, 128, G\}$. The same configuration is used across all tasks and benchmark suites without task-specific tuning. Each feasibility branch uses independent twin Q critics, a state-value network, and a Gaussian policy head, with signed safety values $h_+ = 25$ and $h_- = -1$. The shared reward and cost critics are trained with IQL-style value learning and conservative twin-critic aggregation. Disagreement thresholds are calibrated once from the empirical action-disagreement distribution on offline states and remain fixed during evaluation. Appendix D provides implementation details, pseudocode, and computational measurements.

5 Experiments

Experiment Setup. We evaluate MSPC on 18 offline safe RL tasks from two benchmark suites: **Bullet-Safety-Gym** (Gronauer, 2022) and **Safety-Gymnasium** (Ji et al., 2023). These benchmarks are included in the DSRL dataset collection (Liu et al., 2023a) and span diverse continuous-control settings including locomotion, navigation, goal reaching, and object interaction under safety constraints. We report normalized return and normalized cost following the benchmark-specific normalization protocol (Appendix C.4), and refer to them as "reward" and "cost" for brevity. Following the DSRL evaluation protocol, the episodic safety budget is set to $B = 10$, and a task is considered safe when its average normalized cost does not exceed 1.0.

Baselines. We compare MSPC with seven representative offline safe RL baselines covering distinct algorithmic paradigms: **(1) BCQ-Lag** (Fujimoto et al., 2019), batch-constrained Q-learning with Lagrangian safety penalties; **(2) CPQ** (Xu et al., 2022), offline value learning with constraint penalties; **(3) COptiDICE** (Lee et al., 2022), constrained policy optimization via stationary-distribution correction; **(4) CDT** (Liu et al., 2023b), sequence modeling conditioned on reward- and cost-to-go; **(5) TREBI** (Lin et al., 2023), diffusion planning under deployment-time safety budgets; **(6)**

Table 1: Compact summary on DSRL benchmarks. Reward and cost are suite-level task averages; $\pm$ denotes the average per-task standard deviation across seeds. A result is *safe* if its mean normalized cost is at most 1.0. **Bold**: safe; gray: unsafe; "Safe" counts safe tasks, and "Total Safe" sums these counts across both suites. Full per-task results are reported in Appendix E.

Method	Bullet-Safety-Gym			Safety-Gymnasium			Total Safe
	Reward ↑	Cost ↓	Safe	Reward ↑	Cost ↓	Safe	
BCQ-Lag	0.69 ± 0.09	4.92 ± 1.24	1/8	0.45 ± 0.04	10.25 ± 2.81	1/10	2/18
CPQ	$\mathbf{0.30 \pm 0.06}$	$\mathbf{0.97 \pm 1.20}$	5/8	0.14 ± 0.09	12.45 ± 2.33	1/10	6/18
COptiDICE	0.54 ± 0.02	4.38 ± 0.72	1/8	0.31 ± 0.05	4.84 ± 2.19	1/10	2/18
CDT	0.62 ± 0.01	1.05 ± 0.20	3/8	0.53 ± 0.02	4.06 ± 0.52	3/10	6/18
FISOR	$\mathbf{0.38 \pm 0.03}$	$\mathbf{0.25 \pm 0.24}$	8/8	$\mathbf{0.31 \pm 0.03}$	$\mathbf{0.74 \pm 0.24}$	6/10	14/18
TREBI	0.42 ± 0.05	2.39 ± 1.16	1/8	0.09 ± 0.02	2.10 ± 1.10	4/10	5/18
CAPS-IQL	$\mathbf{0.52 \pm 0.03}$	$\mathbf{0.43 \pm 0.32}$	7/8	0.34 ± 0.05	1.54 ± 1.05	6/10	13/18
Ours	$\mathbf{0.39 \pm 0.07}$	$\mathbf{0.31 \pm 0.25}$	7/8	$\mathbf{0.29 \pm 0.03}$	$\mathbf{0.50 \pm 0.23}$	8/10	15/18

FISOR (Zheng et al., 2024), Hamilton–Jacobi-style feasibility learning with diffusion-based policy extraction; **(7) CAPS-IQL** (Chemingui et al., 2025), IQL-based budget-aware switching among policies with different reward–cost trade-offs.

Main Results. Table 1 summarizes the benchmark results, with full per-task results deferred to Appendix E. Across the two benchmark suites, MSPC achieves the strongest overall constraint-satisfaction coverage, satisfying the safety budget on $15/18$ tasks while maintaining competitive returns. On Bullet-Safety-Gym, MSPC satisfies $7/8$ tasks and achieves a reward–cost trade-off comparable to the strongest safety-oriented baselines: FISOR attains slightly broader safety coverage, while CAPS-IQL achieves higher reward at the same safe-task count. On Safety-Gymnasium, MSPC achieves the highest safe-task coverage among all compared methods, satisfying $8/10$ tasks while maintaining a low suite-level cost and competitive reward relative to other safety-oriented approaches. In contrast, several higher-reward value- or sequence-based methods incur substantially larger costs and violate the constraint on more tasks. Notably, most compared methods directly incorporate reward maximization into policy learning, whereas the candidate policies in MSPC are extracted exclusively from safety-oriented supervision, with reward information introduced only during deployment-time coordination. The competitive reward–cost trade-off therefore suggests that combining complementary safety-specialized policies can retain strong task performance without requiring reward preferences to be explicitly embedded in each low-level policy.

6 Ablation Study and Analysis

6.1 Non-Nested Safety Across Temporal Specialists

We compare deterministic deployments of the four specialists with MSPC on the same 18 tasks. A task is safe when its mean normalized cost is at most 1. Fixed specialists are evaluated with two actor seeds and ten episodes per seed. MSPC uses the same actor family crossed with three shared-critic seeds, yielding six actor–critic configurations with ten episodes per task for each configuration.

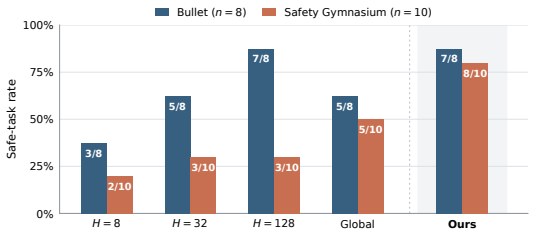
Non-nested closed-loop safety. H8, H32, H128, and Global satisfy the budget on 5, 8, 10, and 10 tasks, respectively (Fig. 2a). Even among finite horizons, safety need not improve with a longer window: H8 is safe on SwimmerVelocity, while H32 and H128 are not (Fig. 2b). H128 is safe on CarRun and DroneRun where Global is not; the reverse holds on HalfCheetahVelocity and SwimmerVelocity. These reversals show why nested finite-window supervision should not be confused with an ordering of closed-loop safety. Appendix F examines the associated violation timing and persistence in the offline data.

Action-level coordination beyond fixed-policy selection. The best fixed specialist is safe on 10 tasks, and the union of all fixed specialists' safe-task sets covers 12. MSPC satisfies the budget on 15 tasks, including five of the six tasks on which every fixed specialist is unsafe on average. Its coverage gain therefore cannot be explained solely by selecting an independently safe specialist for

Table 2: Temporal reward–cost operating points.

	Bullet-Safety-Gym		Safety-Gymnasium	
Policy	Reward ↑	Cost ↓	Reward ↑	Cost ↓
Fixed temporal specializations				
H8	$.548 \pm .035$	$2.514 \pm .423$	$.383 \pm .030$	$2.623 \pm .706$
H32	$.355 \pm .097$	$1.763 \pm .248$	$.382 \pm .027$	2.787 ± 1.211
H128	$.365 \pm .030$	$1.363 \pm .280$	$.377 \pm .022$	3.631 ± 1.716
Global	$.260 \pm .053$	$1.584 \pm .359$	$.327 \pm .022$	$1.038 \pm .355$
Routed temporal policies				
H8+G	$.440 \pm .037$	$.886 \pm .184$	$.293 \pm .037$	$.712 \pm .450$
Full4	$.389 \pm .072$	$.308 \pm .252$	$.291 \pm .028$	$.498 \pm .227$

each task. These results are consistent with useful action-level complementarity among temporal safety behaviors.



(a) Safe-task coverage within each benchmark suite.

Task	H8	H32	H128	Global
CarRun	**0.000**	**0.015**	**0.145**	1.455
DroneRun	6.555	6.590	**0.000**	4.975
HalfCheetahVel.	4.965	3.625	7.360	**0.985**
SwimmerVel.	**0.005**	7.185	13.085	**0.015**

(b) Representative normalized costs.

Figure 2: Fixed-policy safety is neither uniform nor generally nested across temporal specializations. **(a)** Fraction of tasks satisfying normalized cost ≤ 1; bar labels report safe-task counts. MSPC coordinates all four temporal specializations. **(b)** Representative tasks on which H128 and Global reverse their relative safety outcomes.

6.2 Temporal Complementarity Beyond Short–Long Endpoints

Table 2 shows that the fixed temporal specializations occupy substantially different reward–cost operating points. On Bullet-Safety-Gym, H8 attains the highest fixed-policy reward, whereas Global achieves substantially lower reward, with H32 and H128 occupying intermediate operating regions. Safety cost, however, is not monotonic in the horizon, particularly on Safety-Gymnasium. Thus, the safety horizon should not be interpreted as a scalar conservatism parameter; different temporal supervision induces qualitatively different reward–cost behaviors. The H8–Global pair captures two distinct endpoints but does not fully reproduce the benefit of the complete family. On Bullet-Safety-Gym, H8+Global achieves higher reward than Full4 (0.440 versus 0.389), but Full4 reduces cost from 0.886 to 0.308 and improves safe-task coverage from $6/8$ to $7/8$. On Safety-Gymnasium, the two controllers obtain nearly identical reward (0.293 versus 0.291), while Full4 reduces cost from 0.712 to 0.498 and improves coverage from $7/10$ to $8/10$. These results indicate that the intermediate temporal specializations contribute safety behaviors that are not fully captured by the short–long endpoints alone.

6.3 Disagreement Identifies When Coordination Is Informative

Evaluating the shared reward and cost critics at every environment step is often unnecessary. When temporal specialists propose nearly identical actions, critic-based coordination has limited opportunity to meaningfully distinguish among them. We therefore use actor disagreement only as a *query-worthiness signal*. To examine whether disagreement is informative for candidate evaluation, we compare each finite-horizon specialist with Global on logged BallRun states. States are grouped into deciles according to their Global-relative action disagreement, and we measure the corresponding

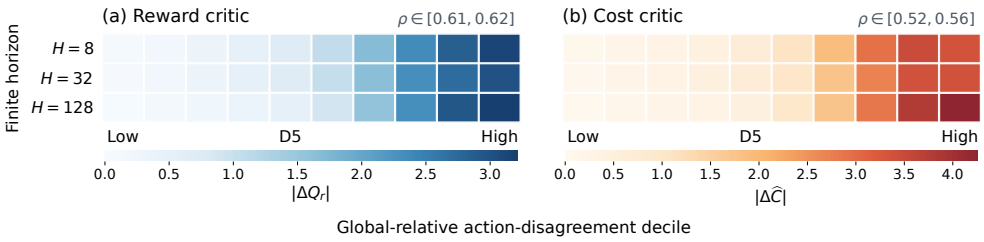


Figure 3: Critic-score separation as a function of Global-relative action disagreement on BallRun. Columns correspond to action-disagreement deciles, and rows compare each finite-horizon specialist with Global. Color denotes the mean absolute difference in reward score (left) or predicted cost (right). Larger disagreement is associated with stronger critic separation.

Table 3: Performance and coordination frequency of every-step and disagreement-gated coordination at budget $B = 10$. Coordination rate reports the mean percentage of environment steps on which the shared critics are re-evaluated.

Suite	Refresh	Reward ↑	Cost ↓	Coordination rate (%)
Bullet	Every-step	0.400 ± 0.073	0.354 ± 0.243	100.0
	Gated (ours)	0.389 ± 0.072	0.308 ± 0.252	81.8
Safety Gym.	Every-step	0.290 ± 0.026	0.430 ± 0.194	100.0
	Gated (ours)	0.291 ± 0.028	0.498 ± 0.227	70.5
All 18 tasks	Every-step	0.339 ± 0.047	0.396 ± 0.216	100.0
	Gated (ours)	0.334 ± 0.048	0.413 ± 0.238	75.5

absolute separation in reward and cost critic scores. Figure 3 shows that both score gaps increase substantially with action disagreement for all three finite horizons. Thus, low-disagreement states tend to produce similar critic evaluations across temporal specialists, whereas high disagreement marks states in which coordination provides greater discriminative information.

The disagreement gate also reduces coordination overhead in deployment. As shown in Table 3, disagreement gating avoids 24.5% of coordination queries across all 18 tasks while largely preserving overall performance, with aggregate normalized reward/cost of $0.334/0.413$, compared with $0.339/0.396$ for every-step re-selection. The gated specialist-switch rate is 44.3%, since a query may retain the active specialist. Together with Fig. 3, these results support disagreement-triggered refresh as a practical performance–efficiency trade-off, using disagreement to determine *when* to coordinate rather than to rank candidates.

7 Conclusion

We studied *temporal safety specialization* in offline safe RL: horizon-dependent violation supervision can induce distinct safety-oriented policies, while broader supervision need not yield safer closed-loop behavior. Building on this insight, we proposed **Multi-Horizon Safety Policy Coordination (MSPC)**, which learns temporal specialists from the same offline dataset using a common safety-only extraction objective. Shared reward and cost critics coordinate their current action proposals under a residual safety budget, separating safety-oriented policy learning from reward-aware deployment. Disagreement-triggered refresh further reduces the frequency of critic queries. Across 18 tasks from two benchmark suites, MSPC achieves competitive reward–safety performance and satisfies the evaluation budget on more tasks than are covered by the union of the fixed specialists' safe-task sets. These findings support the value of retaining and coordinating multiple temporal safety behaviors rather than committing to a single fixed specialization. Nevertheless, MSPC relies on a predefined horizon set and learned value estimates, and does not provide formal safety guarantees. Future work could explore adaptive horizon construction to better capture task-dependent safety timescales and uncertainty-aware coordination to mitigate the impact of cost-estimation errors.

AI USE STATEMENT

We used generative AI tools to assist with drafting, language editing, and manuscript presentation, including correcting grammar, refining phrasing, and improving clarity, conciseness, and organization. All AI-assisted revisions were reviewed and edited by the authors to ensure accuracy and consistency with the intended technical content. We take full responsibility for the final content of this work, including all text, claims, and results.

REFERENCES

HM Ahmad, Ehsan Sabouni, Alexander Wasilkoff, Param Budhraja, Zijian Guo, Songyuan Zhang, Chuchu Fan, Christos Cassandras, and Wenchao Li. Hierarchical multi-agent reinforcement learning with control barrier functions for safety-critical autonomous systems. *arXiv preprint arXiv:2507.14850*, 2025.

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

Eitan Altman. *Constrained Markov decision processes*. Routledge, 2021.

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

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-Jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pp. 2242–2253. IEEE, 2017.

Yassine Chemingui, Aryan Deshwal, Honghao Wei, Alan Fern, and Jana Doppa. Constraint-adaptive policy switching for offline safe reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 15722–15730, 2025.

Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3387–3395, 2019.

Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, Todd Hester, Cosmin Paduraru, and Yuval Tassa. Safe exploration in continuous action spaces. *arXiv preprint arXiv:1801.08757*, 2018.

Thomas G Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019.

Jaime F Fisac, Neil F Lugovoy, Vicenç Rubies-Royo, Shromona Ghosh, and Claire J Tomlin. Bridging hamilton-jacobi safety analysis and reinforcement learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8550–8556. IEEE, 2019.

Justin Fu, Aviral Kumar, Ofir Nachum, George Tucker, and Sergey Levine. D4rl: Datasets for deep data-driven reinforcement learning. *arXiv preprint arXiv:2004.07219*, 2020.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062. PMLR, 2019.

Sven Gronauer. Bullet-safety-gym: A framework for constrained reinforcement learning. Technical report, mediaTUM, 2022.

Kihyuk Hong, Yuhang Li, and Ambuj Tewari. A primal-dual-critic algorithm for offline constrained reinforcement learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 280–288. PMLR, 2024.

Hao Hu, Yiqin Yang, Qianchuan Zhao, and Chongjie Zhang. On the role of discount factor in offline reinforcement learning. In *International Conference on Machine Learning*, pp. 9072–9098. PMLR, 2022.

Jiaming Ji, Borong Zhang, Jiayi Zhou, Xuehai Pan, Weidong Huang, Ruiyang Sun, Yiran Geng, Yifan Zhong, Josef Dai, and Yaodong Yang. Safety gymnasium: A unified safe reinforcement learning benchmark. *Advances in Neural Information Processing Systems*, 36:18964–18993, 2023.

Prajwal Koirala, Zhanhong Jiang, Soumik Sarkar, and Cody Fleming. Latent safety-constrained policy approach for safe offline reinforcement learning. *arXiv preprint arXiv:2412.08794*, 2024.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*, 2021.

Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33:1179–1191, 2020.

Hoang Le, Cameron Voloshin, and Yisong Yue. Batch policy learning under constraints. In *International Conference on Machine Learning*, pp. 3703–3712. PMLR, 2019.

Jongmin Lee, Cosmin Paduraru, Daniel J Mankowitz, Nicolas Heess, Doina Precup, Kee-Eung Kim, and Arthur Guez. Coptidice: Offline constrained reinforcement learning via stationary distribution correction estimation. *arXiv preprint arXiv:2204.08957*, 2022.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Chengjing Li, Li Wang, and Xiaoyan Zhao. Safe robot control through multi-task offline reinforcement learning with multi-scale distribution debiasing. *Computers, Materials & Continua*, 88(1), 2026.

Jinning Li, Liting Sun, Jianyu Chen, Masayoshi Tomizuka, and Wei Zhan. A safe hierarchical planning framework for complex driving scenarios based on reinforcement learning. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2660–2666. IEEE, 2021.

Qian Lin, Bo Tang, Zifan Wu, Chao Yu, Shangqin Mao, Qianlong Xie, Xingxing Wang, and Dong Wang. Safe offline reinforcement learning with real-time budget constraints. In *International Conference on Machine Learning*, pp. 21127–21152. PMLR, 2023.

Yongshuai Liu, Jiaxin Ding, and Xin Liu. Ipo: Interior-point policy optimization under constraints. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 4940–4947, 2020.

Zuxin Liu, Zijian Guo, Haohong Lin, Yihang Yao, Jiacheng Zhu, Zhepeng Cen, Hanjiang Hu, Wenhao Yu, Tingnan Zhang, Jie Tan, et al. Datasets and benchmarks for offline safe reinforcement learning. *arXiv preprint arXiv:2306.09303*, 2023a.

Zuxin Liu, Zijian Guo, Yihang Yao, Zhepeng Cen, Wenhao Yu, Tingnan Zhang, and Ding Zhao. Constrained decision transformer for offline safe reinforcement learning. In *International Conference on Machine Learning*, pp. 21611–21630. PMLR, 2023b.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in Neural Information Processing Systems*, 31, 2018.

Alex Ray, Joshua Achiam, and Dario Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7(1):2, 2019.

Joshua Romoff, Peter Henderson, Ahmed Touati, Emma Brunskill, Joelle Pineau, and Yann Ollivier. Separating value functions across time-scales. In *International Conference on Machine Learning*, pp. 5468–5477. PMLR, 2019.

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

Adam Stooke, Joshua Achiam, and Pieter Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pp. 9133–9143. PMLR, 2020.

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

Mumuksh Tayal, Manan Tayal, and Ravi Prakash. Safe flow q-learning: Offline safe reinforcement learning with reachability-based flow policies. *Reinforcement Learning Journal*, 7, 2026.

Chen Tessler, Daniel J Mankowitz, and Shie Mannor. Reward constrained policy optimization. *arXiv preprint arXiv:1805.11074*, 2018.

Xiangwei Wang, Peng Wang, Renke Huang, Xiuli Zhu, Javier Arroyo, and Ning Li. Safe deep reinforcement learning for building energy management. *Applied Energy*, 377:124328, 2025.

Jingda Wu, Chao Huang, Hailong Huang, Chen Lv, Yuntong Wang, and Fei-Yue Wang. Recent advances in reinforcement learning-based autonomous driving behavior planning: A survey. *Transportation Research Part C: Emerging Technologies*, 164:104654, 2024.

Zikang Xiong, Ishika Agarwal, and Suresh Jagannathan. Hisarl: A hierarchical framework for safe reinforcement learning. In *SafeAI@ AAAI*, 2022.

Haoran Xu, Xianyuan Zhan, and Xiangyu Zhu. Constraints penalized q-learning for safe offline reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 8753–8760, 2022.

Yihang Yao, Zhepeng Cen, Wenhao Ding, Haohong Lin, Shiqi Liu, Tingnan Zhang, Wenhao Yu, and Ding Zhao. Oasis: Conditional distribution shaping for offline safe reinforcement learning. *Advances in Neural Information Processing Systems*, 37:78451–78478, 2024.

Haonan Yu, Wei Xu, and Haichao Zhang. Towards safe reinforcement learning with a safety editor policy. *Advances in Neural Information Processing Systems*, 35:2608–2621, 2022.

Peipei Yu, Hongcai Zhang, Yonghua Song, Zhenyi Wang, Huiyu Dong, and Liang Ji. Safe reinforcement learning for power system control: A review. *Renewable and Sustainable Energy Reviews*, 223:116022, 2025.

Qin Zhang, Linrui Zhang, Haoran Xu, Li Shen, Bowen Wang, Yongzhe Chang, Xueqian Wang, Bo Yuan, and Dacheng Tao. Saformer: A conditional sequence modeling approach to offline safe reinforcement learning. *arXiv preprint arXiv:2301.12203*, 2023.

Zhili Zhang, HM Sabbir Ahmad, Ehsan Sabouni, Yanchao Sun, Furong Huang, Wenchao Li, and Fei Miao. Safety-guaranteed robust multi-agent reinforcement learning with hierarchical control for connected and automated vehicles. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8522–8528. IEEE, 2025a.

Ziqian Zhang, Haojie Li, Tiantian Chen, NN Sze, Wenzhang Yang, Yihao Zhang, and Gang Ren. Decision-making of autonomous vehicles in interactions with jaywalkers: A risk-aware deep reinforcement learning approach. *Accident Analysis & Prevention*, 210:107843, 2025b.

Yinan Zheng, Jianxiong Li, Dongjie Yu, Yujie Yang, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Safe offline reinforcement learning with feasibility-guided diffusion model. *arXiv preprint arXiv:2401.10700*, 2024.

LIU Zifan, Xinran Li, Shibo Chen, and Jun Zhang. GAS: Enhancing reward-cost balance of generative model-assisted offline safe rl. In *The Fourteenth International Conference on Learning Representations*, 2026.

A Additional Discussions on Soft and Hard Safety Constraints

Safe reinforcement learning is commonly formulated either through expectation-based cumulative constraints or through stronger state-wise safety requirements. These formulations provide different notions of safety and should be distinguished when interpreting MSPC.

Soft Constraints. The dominant model-free formulation is the constrained Markov decision process (CMDP) (Altman, 2021), where the objective is

$$\max_{\pi} J_r(\pi) \qquad \text{s.t.} \qquad J_c(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty}\gamma^t c(s_t,a_t)\right] \le B. \tag{23}$$

Methods such as RCPO (Tessler et al., 2018), IPO (Liu et al., 2020), and Lagrangian safe RL (Stooke et al., 2020; Ray et al., 2019) enforce this constraint through reward–cost optimization, while related offline methods must additionally contend with limited data coverage and distribution shift (Fujimoto et al., 2019; Xu et al., 2022; Kumar et al., 2020; Levine et al., 2020). Such constraints control cumulative cost in expectation rather than guaranteeing that every individual trajectory or state transition is violation-free.

Hard Constraints. Hard-safety methods instead aim to maintain the system inside a safe set or enforce a state-wise condition at every decision step. A typical formulation requires

$$\Pr(s_{t+1} \in \mathcal{S}_{\text{safe}} \mid s_t \in \mathcal{S}_{\text{safe}}, a_t \sim \pi(\cdot \mid s_t)) = 1, \qquad \forall t. \tag{24}$$

Representative approaches include control barrier functions (Ames et al., 2019; Cheng et al., 2019), Hamilton–Jacobi reachability (Bansal et al., 2017; Fisac et al., 2019; Zheng et al., 2024), and shielding or safety-layer methods (Dalal et al., 2018; Alshiekh et al., 2018). These approaches can provide substantially stronger guarantees when their assumptions hold, but typically require explicit dynamics, barrier functions, reachable-set models, or sufficiently accurate safety certificates.

Positioning of MSPC. MSPC remains fundamentally a soft-constrained offline RL method: the original cumulative cost c and budget B define both the evaluation criterion and the deployment-time safety allowance. The feasibility signal used at the low level does not replace this CMDP constraint. Instead, it provides a safety-oriented supervision mechanism for constructing candidate policies with different temporal specializations. For a finite horizon H, MSPC labels a logged state–action pair according to whether a violation appears within the following H transitions, thereby biasing policy extraction toward behavior-supported actions associated with more favorable finite-horizon safety outcomes. The Global branch propagates the same signed violation semantics recursively to provide a complementary long-range specialization.

This feasibility-guided construction is related in spirit to hard-safety reasoning because it distinguishes desirable and undesirable safety outcomes at the action level, but it does not define an analytical safe set or enforce state-wise invariance. In particular, the learned feasibility critics are estimated from offline trajectories and are therefore subject to finite data coverage, function approximation, and behavior-policy mismatch. Consequently, the resulting temporal specialists should be interpreted as *safety-oriented candidate policies*, not as certified safe controllers.

At deployment, MSPC returns to the original soft budget semantics. Given candidate actions $\{a_t^k\}_{k\in\mathcal{K}}$, the shared cost critic identifies candidates whose predicted cost-to-go is compatible with the residual allowance,

$$\mathcal{F}_t = \left\{k \in \mathcal{K} : \widehat{Q}_c(s_t, a_t^k) < \epsilon_t\right\}, \tag{25}$$

and the reward critic selects among these candidates when $\mathcal{F}_t$ is nonempty. This mechanism is therefore a learned budget-aware screening rule rather than a hard safety filter: it depends on estimated cost values and does not guarantee that the executed action preserves membership in an invariant safe set.

Overall, MSPC combines two complementary safety semantics without conflating them. Temporal feasibility supervision is used to diversify the low-level candidate policies according to when future violations become visible, while the original CMDP cost and residual budget govern deployment-time coordination. This design improves empirical constraint satisfaction without claiming the formal guarantees associated with hard-safety methods.

B Theoretical Analysis of Temporal Safety Specialization and Coordination

This section provides structural analysis for three components of MSPC. We first characterize how increasing the safety horizon changes the offline feasibility supervision. We then show how horizon-

dependent feasibility weighting induces distinct policies from the same behavior distribution. Finally, we formalize why small action disagreement provides a useful signal for skipping redundant critic-based coordination. These results describe properties of the learning and coordination mechanisms and should not be interpreted as formal safety guarantees.

B.1 Structure of Multi-Horizon Safety Supervision

Consider an anchor transition (s_t, a_t) from the offline dataset and define the time to the next logged violation as

$$\tau_t = \inf \{j \geq 0 : c_{t+j} > 0\}, \tag{26}$$

where $\tau_t = \infty$ if no violation occurs before the end of the trajectory. For a finite safety horizon H, define the binary violation label

$$Y_H(s_t, a_t) = \mathbb{1}[\tau_t < H]. \tag{27}$$

This binary representation is equivalent to the signed target used by MSPC up to an affine transformation:

$$y_t^H = h_- + (h_+ - h_-)Y_H(s_t, a_t). \tag{28}$$

Proposition B.1 (Nested supervision and incremental temporal information)**.** *For any two finite horizons $H_1 < H_2$,*

$$Y_{H_1}(s_t, a_t) \leq Y_{H_2}(s_t, a_t) \tag{29}$$

for every logged anchor transition. Moreover,

$$Y_{H_2}(s_t, a_t) - Y_{H_1}(s_t, a_t) = \mathbb{1}[H_1 \leq \tau_t < H_2]. \tag{30}$$

Therefore, under the empirical anchor distribution $\rho_{\mathcal{D}}$,

$$\mathbb{E}_{\rho_{\mathcal{D}}}[Y_{H_2} - Y_{H_1}] = \Pr_{\rho_{\mathcal{D}}}(H_1 \leq \tau_t < H_2). \tag{31}$$

Proof. If $\tau_t < H_1$, then $\tau_t < H_2$, so every positive label under H_1 is also positive under H_2. Hence $Y_{H_1} \leq Y_{H_2}$. The two labels differ exactly when the first violation occurs after $H_1 - 1$ but before H_2, which gives Eq. 30. Taking expectation under $\rho_{\mathcal{D}}$ yields Eq. 31. □

Proposition B.1 makes explicit what is added when the temporal horizon is extended: moving from H_1 to H_2 introduces supervision only for anchors whose next violation lies in the newly exposed interval $[H_1, H_2)$. Thus, a larger horizon incorporates additional delayed hazards rather than simply rescaling the same safety signal.

This construction also exposes a potential saturation effect. Let

$$p_H = \Pr_{\rho_{\mathcal{D}}}(\tau_t < H) = \mathbb{E}_{\rho_{\mathcal{D}}}[Y_H]. \tag{32}$$

Corollary B.2 (Binary-label saturation)**.** *The positive-label rate p_H is non-decreasing in H, and*

$$\mathrm{Var}_{\rho_{\mathcal{D}}}(Y_H) = p_H(1 - p_H). \tag{33}$$

Consequently, once $p_H \geq 1/2$, further increases in H weakly decrease the variance of the binary label.

Proof. Monotonicity of p_H follows directly from Proposition B.1. Since Y_H is Bernoulli with success probability p_H, its variance is $p_H(1-p_H)$. The function $p(1-p)$ is non-increasing for $p \geq 1/2$. □

The corollary highlights a basic trade-off in finite-horizon supervision. Increasing H reveals additional delayed violations, but a sufficiently broad window can make most logged anchors receive the same positive label, reducing class balance and potentially weakening the discrimination available for action reweighting. Importantly, nested labels do not imply nested learned policies or monotonic closed-loop safety, because each horizon induces a different feasibility estimator and policy update, as analyzed next.

B.2 Temporal Specialization under Feasibility-Weighted Policy Extraction

We next characterize the policy induced by horizon-specific feasibility weighting. For a fixed state s, let $p_{\mathcal{D}}(a \mid s)$ denote the conditional behavior distribution represented by the offline dataset, and let $w_H(s,a) > 0$ denote the weight assigned to action a under safety horizon H. Consider the idealized weighted maximum-likelihood objective

$$\mathcal{L}_H(\pi \mid s) = -\mathbb{E}_{a\sim p_{\mathcal{D}}(\cdot|s)}\left[w_H(s,a)\log \pi(a \mid s)\right]. \tag{34}$$

Proposition B.3 (Horizon-dependent reweighted behavior policy)**.** *Assume that the policy class can represent any conditional distribution over the support of $p_{\mathcal{D}}(\cdot \mid s)$ and that*

$$0 < Z_H(s) = \mathbb{E}_{a\sim p_{\mathcal{D}}(\cdot|s)}\left[w_H(s,a)\right] < \infty. \tag{35}$$

Then the minimizer of Eq. 34 is

$$\pi^*_H(a \mid s) = \frac{p_{\mathcal{D}}(a \mid s) w_H(s,a)}{Z_H(s)}. \tag{36}$$

Proof. Define

$$q_H(a \mid s) = \frac{p_{\mathcal{D}}(a \mid s) w_H(s,a)}{Z_H(s)}. \tag{37}$$

Then Eq. 34 can be written as

$$\mathcal{L}_H(\pi \mid s) = -Z_H(s)\mathbb{E}_{a\sim q_H(\cdot|s)}\left[\log \pi(a \mid s)\right]. \tag{38}$$

Using the cross-entropy decomposition,

$$\mathcal{L}_H(\pi \mid s) = Z_H(s)\left[\mathcal{H}(q_H) + D_{\mathrm{KL}}\left(q_H \| \pi\right)\right], \tag{39}$$

up to terms independent of π. Since the KL divergence is non-negative and equals zero only when $\pi = q_H$ almost everywhere on the support of the behavior policy, Eq. 36 follows. □

For the unclipped exponential feasibility weighting used to motivate MSPC,

$$w_H(s,a) = \exp\left[-\alpha A^H_f(s,a)\right], \tag{40}$$

Proposition B.3 gives

$$\pi^*_H(a \mid s) \propto p_{\mathcal{D}}(a \mid s)\exp\left[-\alpha A^H_f(s,a)\right]. \tag{41}$$

For two horizons H_1 and H_2,

$$\log \frac{\pi^*_{H_2}(a \mid s)}{\pi^*_{H_1}(a \mid s)} = -\alpha\left[A^{H_2}_f(s,a) - A^{H_1}_f(s,a)\right] + \log \frac{Z_{H_1}(s)}{Z_{H_2}(s)}. \tag{42}$$

Therefore, whenever $A^{H_2}_f(s,a) - A^{H_1}_f(s,a)$ is not constant with respect to a on the behavior support, the two horizons induce different conditional action distributions.

This result formalizes the notion of *temporal safety specialization*: all specialists are extracted from the same behavior distribution and use the same policy-learning rule, while differences in temporal feasibility advantages alter how probability mass is redistributed across supported actions. Importantly, Proposition B.1 does not impose a monotonic ordering on these advantages. Even when longer horizons expose a superset of logged violations, the relative feasibility ordering among actions can change, and the independently learned policies subsequently induce different closed-loop state distributions. Hence, nested offline supervision does not imply nested policy-level safety.

B.3 Disagreement as a Query-Worthiness Signal

MSPC uses action disagreement only to determine whether critic-based coordination should be refreshed. We next show that small disagreement necessarily limits the separation that sufficiently smooth action critics can assign to the candidate actions.

Let $\{a^k\}_{k\in\mathcal{K}}$ denote the candidate actions proposed at state s, and define

$$D(s) = \max_{i,j\in\mathcal{K}} \frac{\|a^i - a^j\|_2}{d_{\mathcal{A}}}, \tag{43}$$

where $d_{\mathcal{A}}$ denotes the diameter of the bounded action space.

Proposition B.4 (Disagreement bounds critic-score separation)**.** *Let $\widehat{Q}_x(s,a)$ denote either the shared reward score or predicted cost score, with $x \in \{r,c\}$. Suppose $\widehat{Q}_x(s,\cdot)$ is L_x-Lipschitz in the action argument at state s, i.e.,*

$$\left|\widehat{Q}_x(s,a) - \widehat{Q}_x(s,a')\right| \le L_x \|a - a'\|_2 \tag{44}$$

for all candidate actions a, a'. Then

$$\max_{i,j\in\mathcal{K}} \left|\widehat{Q}_x(s,a^i) - \widehat{Q}_x(s,a^j)\right| \le L_x d_{\mathcal{A}} D(s). \tag{45}$$

Proof. By the Lipschitz assumption, for every pair (i,j),

$$\left|\widehat{Q}_x(s,a^i) - \widehat{Q}_x(s,a^j)\right| \le L_x \|a^i - a^j\|_2. \tag{46}$$

Taking the maximum over candidate pairs and substituting the definition of $D(s)$ gives

$$\max_{i,j} \left|\widehat{Q}_x(s,a^i) - \widehat{Q}_x(s,a^j)\right| \le L_x \max_{i,j} \|a^i - a^j\|_2 = L_x d_{\mathcal{A}} D(s). \tag{47}$$

□

A direct consequence is that if $D(s) \le \tau_D$, then all candidate reward-score differences are bounded by $L_r d_{\mathcal{A}} \tau_D$ and all candidate cost-score differences are bounded by $L_c d_{\mathcal{A}} \tau_D$. Thus, low disagreement provides a sufficient condition for limited critic-score separation and motivates skipping redundant coordination queries when the candidate actions remain close.

The converse does not necessarily hold: large action disagreement does not guarantee large critic-score separation, nor does disagreement itself indicate risk or determine which candidate should be executed. The empirical analysis in Sec. 6.3 complements Proposition B.4 by showing that, for the learned critics used in our experiments, larger disagreement is in practice associated with larger reward and cost score gaps. The theoretical and empirical results together support disagreement as a lightweight *query-worthiness signal*, rather than as an additional safety estimator.

Summary. Proposition B.1 shows that extending the safety horizon introduces precisely the violations that become visible in the newly added temporal interval, while Corollary B.2 highlights that very broad binary windows can become increasingly saturated. Proposition B.3 shows how these horizon-dependent feasibility signals induce distinct reweighted behavior policies from the same offline data. Finally, Proposition B.4 explains why small action disagreement limits the critic separation available for coordination. Together, these results provide a structural justification for MSPC's three central design choices: multiple temporal safety specialists, feasibility-weighted policy extraction, and disagreement-triggered coordination.

C Environments and Datasets

In this section, we describe the benchmark environments and offline datasets used in our experiments, along with their safety constraints, task characteristics, and evaluation metrics. We emphasize properties that are particularly relevant to offline safe reinforcement learning, including distribution shift, sparse safety feedback, and heterogeneity in reward–cost coupling.

C.1 Benchmark Details

We evaluate our method on the **Datasets for Safe Reinforcement Learning (DSRL)** benchmark suite (Liu et al., 2023a), which provides standardized offline datasets for constrained and safe RL. DSRL follows a D4RL-style (Fu et al., 2020) interface and augments each dataset with explicit cost signals, enabling consistent evaluation of reward–safety trade-offs under a unified API.

In this work, we focus on two widely adopted environment suites within DSRL: **Safety-Gymnasium** (Ji et al., 2023) and **Bullet-Safety-Gym** (Gronauer, 2022). Together, these benchmarks cover a diverse range of continuous-control tasks with different dynamics models, safety mechanisms, sensing fidelity, and levels of stochasticity, allowing us to evaluate the generality of our approach across heterogeneous safety regimes.

C.2 Task Descriptions

Fig. 4a provides example renderings of representative environments. Below, we briefly summarize the characteristics of the two benchmark suites and highlight their implications for safety learning.

Safety-Gymnasium. Safety-Gymnasium is a MuJoCo-based benchmark designed for safe continuous control. It includes navigation and locomotion tasks such as `Goal`, `Button`, `Push`, `Circle`, and `Velocity`, instantiated with agents including Point, Car, Ant, HalfCheetah, and Walker. Safety constraints arise from collisions with hazards, obstacle contacts, velocity limits, and boundary violations. Many tasks exhibit sparse or delayed cost signals, where unsafe events may occur only after long sequences of seemingly safe actions. In addition, complex obstacle layouts and contact dynamics introduce discontinuities in the cost landscape, making local generalization unreliable under distribution shift. These properties make long-horizon safety estimation particularly difficult in the offline setting, and motivate the use of localized, horizon-adaptive safety evaluation.

Bullet-Safety-Gym. Bullet-Safety-Gym is built on the PyBullet physics engine and includes tasks such as `Run` and `Circle` with agents including Ball, Car, Drone, and Ant. Compared to MuJoCo-based tasks, these environments often exhibit noisier dynamics, shorter effective horizons, and higher sensitivity to control perturbations. As a result, small modeling or policy errors can rapidly lead to constraint violations. Safety costs typically arise from collisions or boundary violations, and the stochasticity in contact dynamics can amplify uncertainty when extrapolating from offline data. These characteristics emphasize the importance of conservative value estimation and robust policy evaluation, especially when deploying policies outside the dataset distribution.

C.3 Dataset Visualization

To illustrate the structure of the offline datasets, we visualize the distribution of trajectories in the reward–cost space for representative tasks, as shown in Fig. 4b. These visualizations highlight substantial heterogeneity in reward–cost coupling across environments, as well as the prevalence of trade-offs between task performance and safety.

In `CarButton` from Safety-Gymnasium, the dataset exhibits a narrow reward range but a wide spread in cost, indicating weak correlation between task success and safety. Most trajectories incur non-trivial costs even when achieving modest rewards, suggesting that naive reward maximization can easily violate safety constraints. This regime stresses the need for accurate local feasibility assessment and conservative constraint handling.

In contrast, `DroneRun` from Bullet-Safety-Gym shows a smoother and more continuous reward–cost trade-off, where higher rewards are generally associated with increased cost. Although high-reward safe trajectories remain sparse, the presence of intermediate reward–cost regimes provides opportunities for constrained optimization and highlights the role of strategic horizon selection in balancing exploration of performance gains against safety risks.

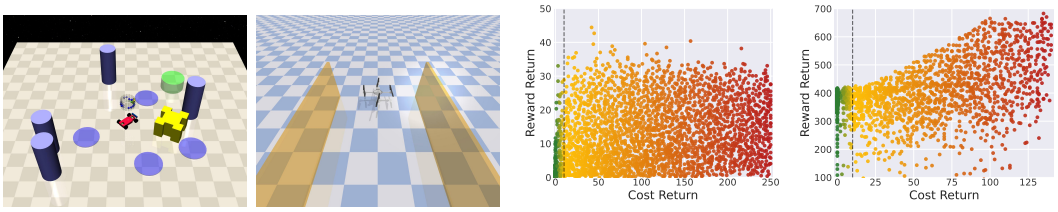
Overall, these visualizations reveal that offline safe RL datasets differ significantly in how safety violations manifest over time and across trajectories, ranging from abrupt, contact-driven failures to gradual accumulation of risk. Such diversity motivates methods that can adapt their planning horizon and safety evaluation strategy to local uncertainty, rather than relying on a single fixed-horizon or monolithic critic.

C.4 Evaluation Metrics

We evaluate all methods in the offline setting using fixed datasets with varying levels of coverage and safety quality. Following standard practice in offline reinforcement learning benchmarks and recent offline safe RL work (Fu et al., 2020), we report *normalized return* and *normalized cost* as evaluation metrics, which we refer to as "reward" and "cost" for brevity. The normalized reward is defined as

$$R_{\text{norm}} = \frac{R_\pi - r_{\min}(\mathcal{T})}{r_{\max}(\mathcal{T}) - r_{\min}(\mathcal{T})}, \tag{48}$$

where R_π denotes the cumulative reward of the learned policy π, and $r_{\max}(\mathcal{T})$, $r_{\min}(\mathcal{T})$ are the maximum and minimum returns observed in the dataset $\mathcal{T}$, respectively. The normalized cost is



(a) Environment rendering (b) Dataset visualization

Figure 4: **Environment and dataset visualization.** From left to right: environment renderings of CarButton and DroneRun, followed by their corresponding dataset return distributions (reward vs. cost).

computed as

$$C_{\text{norm}} = \frac{C_\pi}{\kappa + \epsilon}, \tag{49}$$

where C_π is the cumulative cost incurred by π, κ denotes the safety budget, and ϵ is a small constant for numerical stability. We use a uniform cost limit $\kappa = 10$ for all tasks to ensure consistent comparison of safety performance across methods. Throughout the paper, normalized cost at most 1.0 is treated as satisfying the evaluation budget.

D Implementation Details

D.1 Hyperparameters

For all baselines, we follow the official or released implementations and their recommended hyperparameters whenever available. For BCQ-Lag, CPQ, and CDT, we use the implementations provided by the OSRL framework to ensure a consistent evaluation protocol. For COptiDICE, TREBI, FISOR, and CAPS-IQL, we follow the corresponding released configurations when available. The evaluation horizon is aligned with the maximum episode length of each environment. For methods that expose an explicit safety-budget parameter, we set the episodic cost limit to $B = 10$; otherwise, we retain the default training configuration and evaluate all methods under the same normalized cost criterion, where normalized cost no greater than 1.0 is considered safe.

The common experimental settings and hyperparameters for MSPC are summarized in Table 4. We use the same MSPC configuration across all tasks without task-specific tuning. The finite safety-horizon set is $\mathcal{H} = \{8, 32, 128\}$ and is augmented with the Global feasibility branch, yielding $\mathcal{K} = \{8, 32, 128, G\}$. These finite horizons specify the temporal scope of safety supervision rather than policy execution duration; Appendix D.2 provides a dataset-informed analysis of this choice. We use discount factor $\gamma = 0.99$, and the shared IQL-style reward and cost critics use expectiles $\tau_r = \tau_c = 0.7$. Additional architecture, feasibility-learning, and disagreement-calibration parameters are reported below.

D.2 Dataset-Informed Horizon Set Construction

We instantiate the finite safety-horizon set as $\{8, 32, 128\}$ to provide compact coverage of short-, intermediate-, and long-range temporal safety responses. We use $H = 8$ as the short-range endpoint so that its supervision primarily reflects imminent violations and local corrective behavior. At the other endpoint, $H = 128$ is slightly longer than the nominal discounted horizon $1/(1-\gamma) = 100$ under $\gamma = 0.99$; a consequence occurring 128 steps later is weighted by $\gamma^{128} = 0.276$, corresponding to a 72.4% attenuation. We therefore use $H = 128$ as a practical long-range finite specialization, while retaining the Global branch as a separate recursively propagated long-range safety specialization.

To select an intermediate horizon without repeatedly training policies for different horizon sets, we use the temporal violation structure of the offline datasets. For each candidate $H \in \{8, 16, 32, 64, 128\}$ and anchor transition (s_t, a_t), we construct the binary feasibility label

$$Y_H(s_t, a_t) = \mathbb{1}\left[\sum_{j=0}^{\min(H, T-t)} c_{t+j} > 0\right], \tag{50}$$

Table 4: Key implementation settings of MSPC. The same configuration is used across all Bullet-Safety-Gym and Safety-Gymnasium tasks without task-specific tuning.

Component	Setting	Value
Temporal specialists	Finite safety horizons $\mathcal{H}$	$\{8, 32, 128\}$
	Global branch	Included
	Signed safety values (h_-, h_+)	$(-1, 25)$
	Feasibility expectile τ_f	0.7
	Feasibility weight coefficient α	2.0
Policy extraction	Policy class	Gaussian
	Hidden layers	$[256, 256]$
	Maximum weight $w_{\max}$	200
	Learning rate	3×10^{-4}
Shared critics	Reward / cost expectiles (τ_r, τ_c)	$(0.7, 0.7)$
	Hidden layers	$[512, 512]$
	Discount factor γ	0.99
	Batch size	512
	Reward / cost scales	$0.1/1.0$
Coordination	Evaluation budget B	10
	Instantaneous refresh threshold	p_{25}
	Cumulative refresh threshold	$4p_{25}$
	Threshold calibration	Offline dataset states
Optimization	Specialist batch size	1024
	Optimizer	Adam

Algorithm 1 Offline Training and Calibration of MSPC

Require: Offline dataset $\mathcal{D}$ with trajectory ordering, finite horizons $\mathcal{H}$
1: Set $\mathcal{K} = \mathcal{H} \cup \{G\}$ and construct signed safety signals h_t
2: Construct finite-horizon targets $\{y_t^H\}_{H \in \mathcal{H}}$ from logged trajectory suffixes ▷ Eq. 7
3: Initialize feasibility estimators $\{Q_f^k, V_f^k\}_{k \in \mathcal{K}}$, specialists $\{\pi_k\}_{k \in \mathcal{K}}$, and shared critics (Q_r, Q_c)
4: **while** training has not converged **do**
5: Sample mini-batch $\mathcal{B} \sim \mathcal{D}$
6: **for** $H \in \mathcal{H}$ **do**
7: Update (Q_f^H, V_f^H, π_H) using y_t^H and feasibility-weighted policy extraction ▷ Eq. 13
8: **end for**
9: Update (Q_f^G, V_f^G, π_G) using the recursive Global target ▷ Eq. 8
10: Update shared reward and cost critics using IQL-style value learning ▷ Eq. 16
11: **end while**
12: Calibrate instantaneous and cumulative disagreement thresholds on logged states
13: **Output:** specialist family $\Pi = \{\pi_k\}_{k \in \mathcal{K}}$, shared critics (Q_r, Q_c), and disagreement thresholds

where the window is truncated at the episode boundary. This label is consistent with the finite-horizon supervision used by MSPC: it indicates whether any constraint violation occurs within the following H transitions. Given the fixed endpoints $\mathcal{S} = \{8, 128\}$, we quantify the redundancy of a candidate H with respect to an existing horizon H' as

$$R(H, H') = |\mathrm{corr}(Y_H, Y_{H'})|, \tag{51}$$

and define its novelty relative to the endpoint set as

$$N(H \mid \mathcal{S}) = 1 - \max_{H' \in \mathcal{S}} R(H, H'). \tag{52}$$

A larger novelty therefore indicates that the candidate contributes temporal safety information that is less redundant with both endpoint horizons.

As shown in Table 5, H32 has the lowest maximum redundancy with the two endpoints and consequently the highest novelty, 0.293, compared with 0.145 for H16 and 0.167 for H64. It is selected in

Algorithm 2 Deployment-Time Coordination of MSPC

Require: Specialist family Π, shared critics (Q_r, Q_c), budget B, thresholds (τ_D, τ_Z)
Initialize cumulative cost $C_0 = 0$, accumulated disagreement $z_0 = 0$, active specialist $k_{-1} = \varnothing$
for $t = 0, \ldots, T-1$ **do**
  Generate candidate actions $a_t^k = \pi_k(s_t)$ for all $k \in \mathcal{K}$
  Compute instantaneous disagreement D_t and active-policy deviation δ_t ▷ Eq. 21, 22
  $\widetilde{z}_t = z_t + \delta_t$
  if $t = 0$ **or** $D_t > \tau_D$ **or** $\widetilde{z}_t > \tau_Z$ **then**
    Compute remaining budget b_t and allowance ϵ_t ▷ Eq.18
    Evaluate candidate actions with $\widehat{Q}_r$ and $\widehat{Q}_c$
    Select k_t using the residual-budget rule ▷ Eq. 20
    $z_{t+1} \leftarrow 0$
  else
    $k_t \leftarrow k_{t-1}$ and $z_{t+1} \leftarrow \widetilde{z}_t$
  end if
  Execute $a_t^{k_t}$, observe (c_t, s_{t+1}), and set $C_{t+1} \leftarrow C_t + c_t$
end for

Table 5: Dataset-informed selection of the intermediate safety horizon. States are balanced according to Y_{128} within each task and all 18 Bullet-Safety-Gym and Safety-Gymnasium tasks receive equal weight. Confidence intervals and selection counts are obtained from 500 trajectory-level bootstrap resamples. Lower endpoint redundancy and higher novelty are preferred.

Candidate	Max redundancy	Novelty (95% CI)	Selection
H16	0.855	0.145 [0.145, 0.146]	0/500
H32	**0.707**	**0.293** [**0.292**, **0.294**]	**500/500**
H64	0.833	0.167 [0.165, 0.168]	0/500

all 500 trajectory-level bootstrap resamples. H32 is also the geometric midpoint between H8 and H128, yielding the compact log-spaced family $8 \times 4 = 32$, $32 \times 4 = 128$. At the individual-task level, H32 is preferred on 12 of the 18 tasks, while H64 and H16 are preferred on five and one tasks, respectively. We therefore interpret this analysis as a dataset-informed criterion for constructing a compact and non-redundant horizon set shared across tasks, rather than as evidence that H32 is performance-optimal for every environment.

D.3 Computational Cost

Table 6 reports wall-clock training time, peak allocated GPU memory, and sequential batch-one inference latency measured on an NVIDIA RTX 6000 Ada using `OfflineAntCircle-v0`. All measurements use explicit CUDA synchronization with 20 warm-up and 100 measured training steps, and 200 warm-up and 2000 measured inference actions. Although MSPC evaluates all four temporal specialists at every step to compute action disagreement, its end-to-end inference latency remains below one millisecond and is substantially lower than that of the policy-switching baseline CAPS.

Table 6: Empirical computational cost measured on the same hardware and environment. Peak memory denotes the maximum allocated GPU memory under each method's native data placement. For MSPC, training time is the update-count-weighted average over feasibility-critic, specialist-policy, and shared reward–cost critic training. Inference latency includes four specialist action evaluations and disagreement computation at every step, together with shared-critic evaluation on coordination-query steps.

Method	Train / update (s)	Peak memory (GB)	Latency / action (ms)
FISOR	0.022	1.703	2.664
CDT	0.092	1.954	1.270
CAPS	0.047	0.049	1.968
Ours	0.0007	1.000	0.548

Table 7: **Performance Comparison on DSRL Benchmarks.** All methods use a training and evaluation cost budget of 10. ↑ indicates higher is better and ↓ indicates lower is better. A policy is marked *safe* when its normalized cost is at most 1.0. **Bold**: safe policy; gray: unsafe policy; **bold blue**: best safe policy; **bold**: second-best safe policy.

Task	BCQ-Lag		CPQ		COptiDICE		CDT		FISOR		TREBI		CAPS-IQL		Ours	
	Reward ↑	Cost ↓	Reward ↑	Cost ↓	Reward ↑	Cost ↓	Reward ↑	Cost ↓	Reward ↑	Cost ↓	Reward ↑	Cost ↓	Reward ↑	Cost ↓	Reward ↑	Cost ↓
Bullet-Safety-Gym																
AntRun	0.74	7.21	**0.01**	**0.00**	0.61	1.54	0.72	1.39	**0.45**	**0.30**	0.57	1.20	**0.57**	**0.68**	**0.49**	**0.65**
BallRun	0.64	4.90	0.16	2.13	0.60	6.00	**0.32**	**0.76**	**0.13**	**0.70**	0.22	1.83	0.18	1.01	**0.11**	**0.00**
CarRun	**0.88**	**0.01**	0.93	3.11	**0.90**	**0.00**	**0.99**	**0.64**	**0.69**	**0.01**	**0.97**	**0.99**	**0.97**	**0.22**	**0.89**	**0.00**
DroneRun	0.70	11.09	0.20	1.09	0.65	7.47	**0.58**	**0.22**	**0.32**	**0.95**	0.55	3.20	**0.49**	**0.44**	0.29	1.60
AntCircle	0.53	4.66	**0.00**	**0.00**	0.16	7.83	0.36	1.51	**0.24**	**0.00**	0.12	4.08	**0.30**	**0.00**	**0.17**	**0.00**
BallCircle	0.68	3.52	**0.65**	**0.64**	0.71	4.61	0.71	1.44	**0.34**	**0.00**	0.36	1.39	**0.59**	**0.26**	**0.51**	**0.03**
CarCircle	0.58	3.28	**0.71**	**0.00**	0.46	5.93	0.71	1.24	**0.36**	**0.00**	0.26	3.93	**0.61**	**0.36**	**0.33**	**0.00**
DroneCircle	0.74	4.69	**-0.23**	**0.77**	0.25	1.65	0.57	1.23	**0.46**	**0.03**	0.33	2.52	**0.48**	**0.44**	**0.32**	**0.18**
Bullet-SG Avg	0.69	4.92	**0.30**	**0.97**	0.54	4.38	0.62	1.05	**0.38**	**0.25**	0.42	2.39	**0.52**	**0.43**	**0.39**	**0.31**
Safety-Gymnasium																
CarButton1	0.06	5.75	0.48	32.85	-0.17	2.76	0.18	8.86	**-0.02**	**0.32**	-0.01	1.68	**-0.09**	**0.41**	**-0.03**	**0.43**
CarButton2	0.03	4.40	0.09	47.00	-0.05	3.08	0.19	11.35	**-0.01**	**0.59**	-0.03	1.71	**-0.09**	**0.76**	-0.10	1.02
CarPush1	0.19	1.77	0.06	1.32	0.20	2.13	0.26	1.90	0.28	1.28	**0.07**	**0.18**	0.13	1.003	**0.17**	**0.18**
CarPush2	0.14	4.63	0.12	18.02	0.14	3.90	0.19	4.99	0.11	1.15	-0.04	1.46	0.05	2.20	**0.07**	**0.88**
CarGoal1	0.50	2.16	0.70	5.00	0.37	1.51	0.60	4.07	0.44	1.55	**0.18**	**0.59**	**0.26**	**0.62**	**0.23**	**0.22**
CarGoal2	0.31	4.05	0.29	8.15	0.24	2.14	0.42	6.81	**0.03**	**0.73**	0.06	2.25	0.11	2.09	**0.09**	**0.42**
AntVel	1.01	7.37	**-1.01**	**0.00**	0.99	11.79	**0.98**	**0.68**	**0.89**	**0.00**	**0.35**	**0.17**	**0.90**	**0.25**	**0.50**	**0.00**
HalfCheetahVel	1.05	61.14	0.47	7.81	**0.60**	**0.00**	0.99	1.15	**0.89**	**0.00**	**0.07**	**0.00**	**0.82**	**0.32**	0.83	1.61
SwimmerVel	0.44	11.14	0.13	3.17	0.57	17.75	**0.65**	**0.74**	**-0.02**	**0.14**	0.17	11.36	0.51	7.37	**0.41**	**0.06**
Walker2dVel	**0.79**	**0.10**	0.09	1.16	0.14	3.29	**0.80**	**0.07**	0.50	1.61	0.12	1.63	**0.75**	**0.38**	**0.73**	**0.17**
Safety-Gym Avg	0.45	10.25	0.14	12.45	0.31	4.84	0.53	4.06	**0.31**	**0.74**	0.09	2.10	0.34	1.54	**0.29**	**0.50**

Training-time comparisons should be interpreted as method-native update costs rather than normalized FLOPs, since different algorithms perform different amounts of computation within a single update. For example, one FISOR update jointly performs multiple diffusion-policy and reward/safety value updates, whereas MSPC trains its feasibility critics, temporal specialists, and shared reward–cost critics in separate stages. For call-level profiling, we disable the production scan fusion used by MSPC and execute one JIT-compiled update per call. Under the native 100-update compiled scan, the complete 500k/1M/100k training schedule requires approximately 13.0 minutes of steady-state optimization time on `OfflineAntCircle-v0`, excluding one-time data loading and compilation.

The disagreement gate further reduces deployment-time critic evaluation. A step without coordination requires approximately 0.264 ms, while a coordination-query step requires approximately 0.640 ms. With the measured coordination-query rate of 75.5% across the 18 benchmark tasks, the resulting end-to-end latency is 0.548 ms per action. Thus, disagreement-based refresh reduces shared-critic evaluation while preserving low overall inference overhead.

E Full Benchmark Results

Table 7 reports the per-task normalized reward and cost for all 18 tasks, complementing the suite-level summary in Table 1. A task is counted as safe when its mean normalized cost is at most 1.0. Across both benchmark suites, MSPC achieves the highest overall safe-task coverage among the evaluated methods, satisfying the budget on 15/18 tasks, compared with 14/18 for FISOR and 13/18 for CAPS-IQL, while maintaining competitive reward–safety performance.

MSPC's advantage is particularly evident on Safety-Gymnasium, where it satisfies the budget on 8/10 tasks, compared with 6/10 for both FISOR and CAPS-IQL. Its suite-level mean normalized cost is 0.50, compared with 0.74 and 1.54, respectively, alongside a mean normalized reward of 0.29. On Bullet-Safety-Gym, MSPC satisfies the budget on 7/8 tasks with a mean normalized reward of 0.39 and cost of 0.31, providing a competitive reward–safety operating point. Together, these results highlight MSPC's broad empirical constraint-satisfaction coverage across heterogeneous tasks.

Notably, these results are obtained with temporal specialists trained exclusively through safety-oriented policy extraction. Finite-window and recursively propagated Global feasibility targets shape distinct candidate behaviors, while shared reward and cost critics coordinate their current action proposals under the remaining budget. The benchmark results demonstrate that broad safety coverage

Table 8: Temporal violation profiles of representative tasks. Risk-band percentages are conditioned on $\tau_t < 128$ and sum to 100% up to rounding. $\Pr(\tau < 128)$ is computed over all logged states, and L_c denotes the mean length of consecutive positive-cost runs. The final column reports the safest fixed specialization in Fig. 2.

		Logged violation profile					
Task	T	$\Pr(\tau < 128)$	$[0, 8)$	$[8, 32)$	$[32, 128)$	Mean L_c	Safest fixed
CarRun	200	60.2%	34.4%	36.2%	29.5%	2.9	H8
DroneRun	200	69.0%	41.8%	15.7%	42.5%	51.8	H128
HalfCheetahVelocity	1000	85.6%	48.3%	28.0%	23.7%	1.6	Global
SwimmerVelocity	1000	75.2%	28.6%	39.7%	31.7%	5.0	H8 (Global)

and competitive task performance can be achieved without directly embedding reward preferences into each low-level specialist.

F Additional Analysis

Why Do Tasks Favor Different Safety Horizons? Although the finite-horizon feasibility labels are nested by construction, $Y_8(s_t) \le Y_{32}(s_t) \le Y_{128}(s_t)$ for every logged state, the learned policies need not inherit the same ordering: different horizons reweight different subsets of offline actions and subsequently induce different closed-loop state distributions. To characterize this temporal structure, we define $\tau_t = \min\{j \ge 0 : c_{t+j} > 0\}$ as the time to the next logged violation and partition states with $\tau_t < 128$ into short $[0, 8)$, intermediate $[8, 32)$, and delayed $[32, 128)$ risk bands. We additionally report the mean length L_c of consecutive positive-cost runs. As summarized in Table 8, CarRun exhibits short violation episodes ($L_c = 2.9$), consistent with finite-horizon policies being sufficient, whereas DroneRun combines a large delayed-risk component (42.5% in $[32, 128)$) with highly persistent violations ($L_c = 51.8$), favoring H128. HalfCheetahVelocity instead exhibits brief but recurrent violations, with 85.6% of logged states already observing a violation within 128 steps; the resulting broad binary H128 supervision is comparatively less discriminative, while the recursively propagated Global specialization is safer. SwimmerVelocity exhibits smoother short-range regulation, for which both H8 and Global remain safe. These examples illustrate that the same numerical horizon can represent different effective safety time scales across systems, depending on control response time, delayed or persistent hazards, episode scale, offline coverage, and the discriminative information retained by the corresponding feasibility labels.

Should Reward Be Optimized within Temporal Specialists? MSPC deliberately extracts its temporal specialists using only horizon-dependent safety supervision, while reward enters only through deployment-time coordination. We examine whether explicitly embedding reward preference into each low-level specialist improves the resulting reward–cost trade-off. The default Safety-only weight is

$$w_k^{\mathrm{S}}(s, a) = \exp\big[\mathrm{clip}\big(-\alpha_c A_f^k(s, a), \log w_{\max}\big)\big], \tag{53}$$

whereas the matched Reward+Safety variant additionally incorporates a standardized reward advantage $\widetilde{A}_r$,

$$w_k^{\mathrm{R+S}}(s, a) = \exp\Big[\mathrm{clip}\Big(-\alpha_c A_f^k(s, a) + \alpha_r \widetilde{A}_r(s, a), \log w_{\max}\Big)\Big]. \tag{54}$$

All remaining actor training and coordination settings are kept matched, with disagreement thresholds recalibrated using the same offline procedure. We report four tasks for which the Safety-only controller satisfies the budget, allowing us to test whether introducing low-level reward preference degrades an already feasible operating point.

Table 9 shows that explicitly introducing reward preference into the temporal specialists consistently shifts the operating point toward higher reward but substantially higher cost. Averaged across the four tasks, normalized reward increases by 0.138, whereas normalized cost rises from 0.593 to 2.460, causing the Reward+Safety variant to exceed the safety threshold on every task. Thus, the additional reward gain does not constitute an improvement under the original safety constraint.

The Reward+Safety variant also produces more concentrated actor weighting on average, with normalized effective sample size decreasing from 0.169 to 0.117. This provides one possible

Table 9: Effect of adding reward preference to low-level temporal specialists. Reward and cost are mean ± std over six matched actor–critic runs, and Δ denotes Reward+Safety minus Safety-only.

	Reward / Cost		Paired change	
Task	Safety-only	Reward+Safety	ΔR	ΔC
AntRun	0.493 ± 0.046 / 0.650 ± 0.177	0.706 ± 0.054 / 3.922 ± 2.511	$+0.214$	$+3.272$
CarButton1	-0.026 ± 0.029 / 0.428 ± 0.343	0.108 ± 0.014 / 1.230 ± 0.530	$+0.133$	$+0.802$
CarGoal2	0.095 ± 0.034 / 0.415 ± 0.287	0.265 ± 0.051 / 3.003 ± 1.138	$+0.170$	$+2.588$
CarPush2	0.068 ± 0.037 / 0.880 ± 0.350	0.103 ± 0.033 / 1.683 ± 0.431	$+0.035$	$+0.803$
Panel average	0.157 ± 0.036 / 0.593 ± 0.289	0.295 ± 0.038 / 2.460 ± 1.152	$+0.138$	$+1.866$

explanation for the observed degradation, although the pattern is not universal across tasks. More broadly, introducing reward advantage directly changes the ordering of logged actions induced by temporal feasibility supervision. These results support MSPC's division of labor: low-level policies preserve complementary safety-oriented behaviors, while reward is introduced only when coordinating their candidate actions.

G Toy Diagnostic Details

This appendix provides additional details for the controlled toy diagnostic shown in Fig. 1. The purpose of this diagnostic is not to reproduce the full offline benchmark setting, but rather to visualize, in a deliberately simple maze, the distinct spatial roles of long-horizon and truncated local cost critics.

Environment and objective. The toy diagnostic uses a simple 2D maze navigation environment with static walls, two side branches, and a single goal region. The upper and lower branches are locally traversable but eventually lead to dead ends, whereas the central passage is the only route that remains globally favorable. Let $p_t \in \mathbb{R}^2$ denote the agent position at time t, let g be the goal location, and define the distance-to-goal as $d_t = \|p_t - g\|_2$. We use a dense progress reward

$$r_t = 1.5\,(d_t - d_{t+1}) + 50\,\mathbf{1}\{s_{t+1} \in \mathcal{G}\}, \tag{55}$$

where the first term continuously rewards motion toward the goal and the second term provides an additional terminal bonus upon reaching the goal region $\mathcal{G}$. Safety is collision-based:

$$c_t = \mathbf{1}\{\text{the agent collides with an obstacle at step } t\}. \tag{56}$$

Thus, each collision incurs an instantaneous unit cost. The task is formulated as a constrained MDP with discounted reward and cost objectives

$$J_r(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{T-1} \gamma^t r_t\right], \qquad J_c(\pi) = \mathbb{E}_\pi\left[\sum_{t=0}^{T-1} \gamma^t c_t\right], \tag{57}$$

where we use $\gamma = 0.99$. The policy is required to satisfy the discounted cost budget

$$J_c(\pi) \le b, \tag{58}$$

with $b = 10$ in this diagnostic. This reward–cost design intentionally combines smooth task progress with sparse local hazards, making the distinction between long-horizon reachability and short-horizon feasibility easy to visualize.

Training with PPO-Lag. For this qualitative study, we train the policy using PPO-Lag, a Lagrangian variant of Proximal Policy Optimization (PPO) widely used in safe RL benchmarks (Schulman et al., 2017; Ray et al., 2019; Stooke et al., 2020). PPO-Lag addresses the constrained objective

$$\max_\pi \; J_r(\pi) \qquad \text{subject to} \qquad J_c(\pi) \le b, \tag{59}$$

by introducing a nonnegative multiplier λ and optimizing a clipped surrogate with the Lagrangian advantage

$$\mathcal{L}_{\text{PPO-Lag}}(\theta) = \mathbb{E}_t\Big[\min\Big(\rho_t(\theta)\,\hat{A}_t^\lambda, \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)\,\hat{A}_t^\lambda\Big)\Big], \quad \hat{A}_t^\lambda = \hat{A}_t^r - \lambda \hat{A}_t^c, \tag{60}$$

where $\rho_t(\theta) = \pi_\theta(a_t \mid s_t)/\pi_{\theta_{\text{old}}}(a_t \mid s_t)$ is the standard PPO importance ratio, and $\hat{A}_t^r$ and $\hat{A}_t^c$ denote the reward and cost advantages, respectively. After each policy update, the multiplier is adjusted by dual ascent so that violations of the cost budget increase the effective penalty on future updates. In our setting, PPO-Lag serves only as a simple and stable cost-aware training backbone for the toy maze; it is not part of the offline training pipeline used in the main experiments.

Critic visualization. Within this controlled environment, we visualize a global long-horizon cost critic and a truncated local cost critic learned with the same underlying policy family. The global critic estimates the discounted long-horizon cost return, while the local critic uses a truncated discounted cost return with horizon $H = 15$. This controlled setup is chosen solely to isolate the tension between short-range traversability and long-horizon reachability.

To obtain the 2D spatial heatmaps, we evaluate the critics on a grid over the position variables (x, y) and average predictions over the remaining state variables, namely orientation and velocity. This projection suppresses nuisance variation and makes the spatial distinction between the critics visually interpretable. The global critic assigns elevated cost to regions that eventually funnel into dead-end branches, whereas the local critic preserves a broader set of immediately traversable states. We emphasize that this figure should be interpreted as a qualitative, mechanism-oriented illustration. It is intended to clarify why we decouple local feasibility shaping from long-horizon scheduling, but it is not a stand-alone proof that the same spatial pattern transfers directly to the high-dimensional benchmark tasks.