CoGeL: Constraint-to-Geometry Learning from Demonstrations and Failures for Safe Long-Horizon Robot Navigation

Jumman Hossain, Nirmalya Roy
Department of Information Systems
University of Maryland, Baltimore County (UMBC), United States
{jumman.hossain, nroy}@umbc.edu

Abstract: Robots are often deployed in environments where safety must be learned from experience rather than specified in advance. Demonstrations, operator interventions, near misses, and failed rollouts can reveal which behaviors are risky, but they provide only local evidence. The key challenge is not only to identify risky behavior, but to make that knowledge shape the distance used for long-horizon planning. Existing goal-conditioned methods can compose short-horizon behaviors into longer plans, but they typically assume that the relevant cost structure is already known. We introduce **CoGeL**, a constraint-to-geometry learning method that learns a soft safety signal from robot data and incorporates it directly into the goal-reaching distance used for planning. This makes the learned distance safety-aware rather than treating safety as a separate classifier, hard mask, or post-hoc filter. We show that errors in the learned safety signal induce bounded changes in the navigation geometry, that route consistency supports composition through intermediate waypoints, and that, under a calibration condition, the learned safety scores bound the expected number of safety violations. Experiments across wheeled and legged navigation tasks show that CoGeL reduces unsafe events while maintaining strong long-horizon goal-reaching performance. **Project Page:** https://pralgomathic.github.io/cogel/

Keywords: Robot Learning, Long-Horizon Navigation, Constraint-to-Geometry Learning, Goal-Conditioned Reinforcement Learning, Quasimetric Learning.

1 Introduction

Robots rarely receive safety as a clean, analytic cost map. Instead, they discover hazards empirically: a patch of terrain causes slip during a rollout, a foothold collapses, or an operator intervenes to prevent a collision [1, 2, 3]. While these events provide critical safety signals, they are fundamentally local, marking only the specific state and action where a failure occurred or was narrowly avoided. The central challenge is translating these sparse, localized hazard signals into a cohesive, globally safe strategy for long-horizon navigation. Goal-conditioned reinforcement learning has become a standard framework for learning policies and value functions that generalize across goals [4, 5, 6, 7, 8, 9]. Recent work further interprets goal-reaching values as distances in a learned space, including temporal-distance models, contrastive goal-conditioned RL, metric residual networks, and quasimetric RL [6, 7, 8, 10, 11]. This geometric view is especially relevant for robot navigation because traversal costs are often directional. Climbing a slope is not the exact reverse of descending it, entering low-friction terrain carries different risks than leaving it, and navigating a narrow passage depends heavily on the approach angle. A directed distance, or quasimetric, can therefore represent the effort and risk of moving between states when traversal costs depend on direction [8, 11, 12, 13, 14]. However, this geometric view has a critical vulnerability when applied to safety. Goal-reaching

10th Conference on Robot Learning (CoRL 2026), Austin TX, USA.

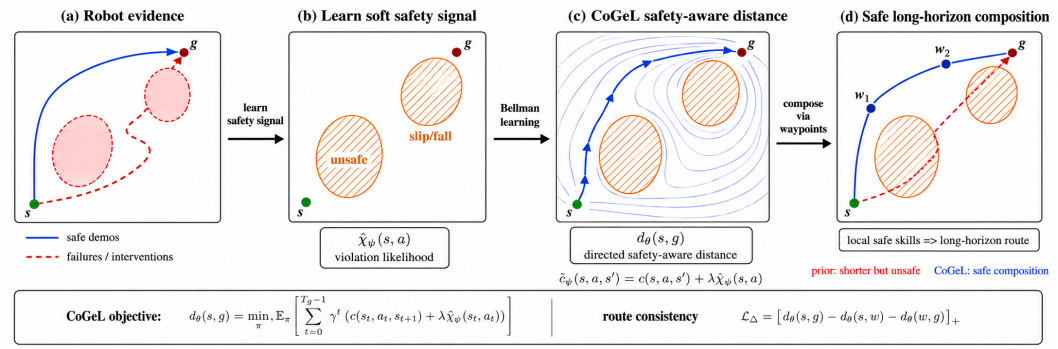


Figure 1: **CoGeL overview.** CoGeL learns a soft safety signal $\hat{\chi}_\psi(s,a)$ from robot evidence, including safe demonstrations, failures, and interventions. This signal is incorporated into the Bellman cost to learn a directed safety-aware distance $d_\theta(s,g)$. The learned distance supports waypoint-based composition, allowing local safe skills to form long-horizon routes that avoid unsafe regions.

distances are typically trained on environments where the underlying cost structure is already known. When safety must instead be inferred from noisy, localized evidence, current integration strategies fracture. Converting a local risk score into a hard threshold creates a brittle action space, where slight estimation errors can permanently block viable corridors or inadvertently open dangerous shortcuts. Conversely, applying the safety score as a post-hoc filter over a planned route leaves the underlying distance metric completely blind to the environment's true hazards, allowing small edge errors to compound catastrophically across intermediate waypoints. While prior work excels at recovering local constraint models from data [15, 16] or optimizing policies under known costs [17], it remains unresolved how localized safety evidence should fundamentally reshape the global geometry used for planning.

We introduce CoGeL, a constraint-to-geometry framework for safe long-horizon navigation (Fig. 1). CoGeL learns a soft safety signal from demonstrations, interventions, and failed rollouts, then incorporates it into the bounded Bellman cost used to learn a directed goal-reaching distance. This makes risky actions increase the learned distance to the goal, producing a safety-aware geometry that can represent directional hazards and compose through waypoints. Because the safety score enters as a bounded penalty rather than a hard decision boundary, the Bellman operator remains a contraction. We show that local safety-estimation errors induce bounded distortion in the learned navigation geometry, that this geometry controls waypoint-composition error, and that calibrated safety scores bound the expected number of true trajectory violations.

Our main contributions are:

- We formulate safe long-horizon navigation with latent constraints as a constraint-to-geometry problem, connecting local safety evidence with directed distance learning.
- We introduce CoGeL, which incorporates a learned soft safety signal directly into the Bellman cost and learns a safety-aware directed distance with route-consistency regularization.
- We provide theoretical guarantees linking safety-estimation error to geometry error, route-consistency error to waypoint composition, and, under a calibration condition, learned safety scores to expected safety violations.
- We show that symmetric metrics have an irreducible approximation gap on directional hazards, motivating CoGeL's directed structure. We validate the method across wheeled and legged navigation settings with hidden hazards, directional risks, and long-horizon routes.

2 Related Work

Goal-conditioned and quasimetric reinforcement learning. Goal-conditioned reinforcement learning learns policies and value functions that can reach many goals from shared experi-

ence [18, 4, 5, 9]. A key challenge is making local goal reaching useful for longer missions. Temporal-difference models, contrastive goal-conditioned RL, metric residual networks, and quasimetric RL address this by learning goal-reaching values that behave like distances [6, 7, 10, 8]. The quasimetric view is useful for navigation because traversal costs are often direction dependent, as in slopes, narrow passages, soft terrain, and directional hazards [8]. Recent work on horizon generalization and multistep quasimetric estimation shows that long-horizon success depends on whether local predictions compose through intermediate goals [19, 12]. These methods provide useful geometric structure, but generally assume that the cost used to learn the distance is already specified. CoGeL instead learns the safety-relevant part of the cost from robot evidence and uses it to shape the directed distance.

Learning safety constraints from data. When safety is not hand specified, it must be inferred from behavior. Inverse constraint learning and related methods recover constraints from demonstrations, unsafe behavior, or demonstrations with unknown rewards [15, 20, 21, 22, 16]. These approaches produce constraint models, safe sets, or local cost estimates, but not a goal-reaching distance that can be composed across waypoints. Hard constraint models can also be sensitive to threshold errors, while planner-side filters leave the learned distance unchanged. CoGeL uses the learned safety signal inside the Bellman cost, so the planning geometry changes smoothly with the safety estimate.

Safe, constrained, and risk-sensitive reinforcement learning. Safe and constrained RL optimizes policies under cost or constraint signals using constrained MDPs, constrained policy optimization, Lagrangian methods, or benchmarked safety costs [23, 17, 24, 25]. Risk-sensitive RL, including CVaR-based methods, provides tools for controlling rare but costly failures [26, 27]. These methods are complementary to CoGeL: they optimize once a safety signal is available, while CoGeL studies how safety evidence from demonstrations, failures, and interventions should shape the directed distance used for long-horizon planning. We use an expected-cost objective to preserve contraction-based analysis, while tail-risk objectives are a natural extension.

Learning-based robot navigation. Learning-based navigation increasingly combines simulation, real-world data, and sim-to-real transfer. Large-scale simulation has enabled efficient training for legged locomotion and interactive robot learning [28, 29, 30], while learned navigation systems show how robots can adapt from real experience and self-supervised data [1, 31]. These systems face hazards that are often not fully labeled before deployment, with evidence arriving through failures, interventions, and near misses. CoGeL addresses this setting by learning a safety-aware directed distance for both local goal reaching and waypoint-based long-horizon navigation.

3 Problem Formulation

We consider a goal-conditioned Markov decision process $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, c, \gamma, \mathcal{G})$, where $s \in \mathcal{S}$ is the robot state, $a \in \mathcal{A}$ is an action, $P(s' \mid s, a)$ is the transition kernel, $c(s, a, s') \geq 0$ is a task cost, $\gamma \in [0, 1)$ is a discount factor, and $g \in \mathcal{G}$ is a goal. A goal-conditioned policy $\pi(a \mid s, g)$ should reach g from s while avoiding unsafe behavior. Let $B(g) \subseteq \mathcal{S}$ denote the goal region, and define the capped hitting time

$$\tau_g := \min\{t \geq 0 : s_t \in B(g)\} \wedge T_{\max}.$$

If the robot fails to reach $B(g)$ by $T_{\max}$, it receives a terminal failure cost c_{fail}. The robot faces an unknown binary safety constraint

$$\chi^\star(s, a) \in \{0, 1\},$$

where $\chi^\star(s, a) = 1$ indicates a true violation, such as collision, slip, fall, excessive tilt, traversal of forbidden terrain, or an emergency stop. The learner does not observe $\chi^\star$ as a complete map. Instead, it receives local robot evidence

$$\mathcal{D} = \mathcal{D}_{\text{safe}} \cup \mathcal{D}_{\text{risk}},$$

where $\mathcal{D}_{\text{safe}}$ contains demonstrations or rollouts judged safe, and $\mathcal{D}_{\text{risk}}$ contains failed rollouts, interventions, near misses, or segments flagged by a safety monitor. Each labeled state-action pair is written as (s, a, y), where $y = 1$ means the pair is associated with unsafe behavior and $y = 0$ means

it is treated as safe evidence. The goal is to learn a policy and a planning distance that generalize from short training goals to longer test missions while keeping both task cost and true violations low. This setting is difficult for two reasons. First, safety is latent: the robot must infer a usable safety signal from incomplete local evidence. Second, deployment is long horizon: a policy that is locally reasonable can still fail if its value function does not support composition through intermediate goals. CoGeL addresses this by learning a soft safety signal from $\mathcal{D}$ and using that signal to shape the goal-reaching distance used for planning. The desired object is not only a classifier over unsafe actions, but a directed distance $d_\theta(s, g)$ that satisfies an approximate composition property,

$$d_\theta(s, g) \lesssim d_\theta(s, w) + d_\theta(w, g),$$

for useful intermediate waypoints w. This property allows local safe skills to form longer routes.

4 Method: Constraint-to-Geometry Learning

CoGeL learns a planning geometry from local safety evidence collected through demonstrations, exploratory rollouts, failures, and interventions. It learns a soft safety signal $\hat{\chi}_\psi(s, a)$ and incorporates it into the one-step cost used to train a directed distance $d_\theta(s, g)$ with Bellman regression and route-consistency regularization. The learned distance supports both local control and waypoint-based long-horizon planning.

4.1 Robot Evidence and Data Collection

CoGeL assumes that safety is not provided as a complete cost map. Instead, the robot collects local evidence while executing demonstrations, exploratory rollouts, and goal-conditioned trials. Each trajectory is stored as

$$\tau = (s_0, a_0, s_1, a_1, \ldots, s_T),$$

together with event information from human supervision or onboard safety monitors. Safe evidence comes from expert demonstrations, successful rollouts, or trajectory segments that reach their local goals without triggering a safety event. Risk evidence comes from failed rollouts, operator interventions, near misses, emergency stops, excessive slip, collision contacts, large tilt, unstable footholds, or traversal of a region later marked unsafe. For each visited state-action pair, we assign a binary evidence label $y_t \in \{0, 1\}$, where $y_t = 1$ means the pair is associated with unsafe behavior and $y_t = 0$ means it is treated as safe evidence. When a failure or intervention occurs at time t_f, we label a short temporal window before the event as risky,

$$y_t = 1 \qquad \text{for} \qquad t \in [t_f - K, t_f],$$

where K is a small pre-event labeling window. This accounts for the fact that the action causing slip, collision, or loss of stability may occur slightly before the monitor fires. Segments far from any failure or intervention are labeled safe if they satisfy the task monitor and remain within nominal state limits. The resulting dataset is

$$\mathcal{D} = \mathcal{D}_{\text{safe}} \cup \mathcal{D}_{\text{risk}},$$

where

$$\mathcal{D}_{\text{safe}} = \{(s_t, a_t, 0)\}, \qquad \mathcal{D}_{\text{risk}} = \{(s_t, a_t, 1)\}.$$

In simulation, the event labels are generated from hidden environment conditions such as low-friction patches, directional hazards, collisions, and unstable footholds. These hidden hazards are never given to the planner as a map. They only appear through the local evidence collected in $\mathcal{D}$. In hardware, the same interface can be populated from operator interventions, emergency-stop logs, contact events, slip detectors, tilt thresholds, and recovery behaviors. CoGeL uses this dataset in two ways. **First**, it trains the soft safety signal $\hat{\chi}_\psi(s, a)$, which estimates the likelihood that a state-action pair leads to a violation. **Second**, the same trajectories populate the replay buffer used for goal-conditioned Bellman learning, with goals relabeled from future states when appropriate. As new failures or interventions are observed, their local neighborhoods are added to $\mathcal{D}_{\text{risk}}$, allowing the safety signal and the learned distance to improve over time.

4.2 Learning a Soft Safety Signal

CoGeL first fits a probabilistic safety estimator, $\hat{\chi}_\psi(s,a) = P_\psi(y = 1 \mid s,a) \in [0,1]$, where ψ parameterizes a neural classifier and $y = 1$ denotes unsafe evidence. Given labeled pairs $(s,a,y) \sim \mathcal{D}$, we train the estimator with binary cross entropy,

$$\mathcal{L}_{\text{safe}}(\psi) = -\mathbb{E}_{(s,a,y)\sim\mathcal{D}}\left[y \log \hat{\chi}_\psi(s,a) + (1-y)\log\left(1-\hat{\chi}_\psi(s,a)\right)\right].$$

This choice is deliberately simple. The downstream method only requires a scalar score $\hat{\chi}_\psi(s,a) \in [0,1]$, so the estimator can be replaced by a calibrated classifier, a conservative safety model, or an uncertainty-aware estimator without changing the rest of CoGeL.

4.3 Safety-Aware Cost

The learned safety signal enters planning through the augmented cost $\tilde{c}_\psi(s,a,s') = c(s,a,s') + \lambda\hat{\chi}_\psi(s,a), \qquad \lambda \geq 0$. The weight λ controls the tradeoff between task efficiency and safety. When $\lambda = 0$, CoGeL reduces to standard goal-conditioned distance learning. As λ increases, actions that the safety estimator marks as risky become more expensive. Unlike a hard mask, this changes the cost smoothly with the estimated violation likelihood.

4.4 Directed Goal-Reaching Distance

Given $\tilde{c}_\psi$, CoGeL defines the safety-aware directed distance as the optimal augmented cost-to-go,

$$d^\star_\psi(s,g) = \min_\pi \mathbb{E}_\pi\left[\sum_{t=0}^{\tau_g-1} \gamma^t \tilde{c}_\psi(s_t,a_t,s_{t+1}) + c_{\text{fail}}\mathbf{1}\{\tau_g = T_{\max},\ s_{\tau_g} \notin B(g)\} \;\middle|\; s_0 = s\right].$$

The distance is generally asymmetric. Traversing uphill need not have the same safety cost as traversing downhill, entering low-friction terrain may be riskier than leaving it, and passing through a narrow gap can depend on the approach direction. For this reason, CoGeL learns a directed distance rather than a symmetric metric. The corresponding Bellman operator is

$$(T_\psi d)(s,g) = \min_{a\in\mathcal{A}} \mathbb{E}_{s'\sim P(\cdot|s,a)}\left[\tilde{c}_\psi(s,a,s') + \gamma d(s',g)\right], \qquad s \notin B(g),$$

with $(T_\psi d)(s,g) = 0$ for $s \in B(g)$. In practice, we approximate the fixed point with a neural distance $d_\theta(s,g)$.

4.5 Learning Objective

CoGeL trains d_θ with Bellman regression and a route-consistency penalty. For sampled transitions (s,a,s') and relabeled goals g, the Bellman target is $y(s,a,s',g) = \tilde{c}_\psi(s,a,s') + \gamma d_{\bar{\theta}}(s',g)$, where $d_{\bar{\theta}}$ is a slowly updated target network. The Bellman loss is

$$\mathcal{L}_{\text{Bell}}(\theta) = \mathbb{E}_{(s,a,s',g)}\left[\left(d_\theta(s,g) - y(s,a,s',g)\right)^2\right].$$

The optimal distance $d^\star_\psi$ above is the fixed point of the action-minimizing Bellman operator. Let $d^{\pi_\phi}_\psi$ denote the augmented cost-to-go under the current actor. In implementation, d_θ is learned from observed replay transitions and is therefore a finite-data, replay-supported approximation. Current-policy and exploration rollouts refresh the replay distribution during training, but Bellman regression always uses the observed successor s'. Consequently, with limited action coverage, d_θ can be pessimistic and need not equal $d^\star_\psi$.

To encourage composition through waypoints, CoGeL also penalizes violations of the quasimetric inequality. For sampled triplets (s,w,g), define

$$\mathcal{L}_\triangle(\theta) = \mathbb{E}_{(s,w,g)}\left[\left[d_\theta(s,g) - d_\theta(s,w) - d_\theta(w,g) - m\right]_+\right],$$

Algorithm 1 Constraint-to-Geometry Learning (CoGeL)

Require: Evidence $\mathcal{D}$, replay buffer $\mathcal{B}$, safety weight λ, consistency weight β, target rate ρ
Train safety estimator $\hat{\chi}_\psi$ on $\mathcal{D}$ using $\mathcal{L}_{\text{safe}}$
Initialize d_θ, $d_{\bar{\theta}} \leftarrow d_\theta$, actor π_ϕ, and one-step model $\hat{f}_\omega$
for each training iteration **do**
    Collect policy/exploration rollouts; update $\mathcal{B}$ and $\mathcal{D}$, and optionally update ψ
    Sample observed (s, a, s') and relabeled goal g from $\mathcal{B}$
    $\tilde{c}_\psi \leftarrow c(s, a, s') + \lambda \hat{\chi}_\psi(s, a), \quad y \leftarrow \tilde{c}_\psi + \gamma d_{\bar{\theta}}(s', g)$
    Sample (s, w, g) and update θ using $\mathcal{L}_{\text{Bell}} + \beta \mathcal{L}_\triangle$
    Update ω on replay using $\|\hat{f}_\omega(s, a) - s'\|_2^2$
    For continuous control, sample $a_\phi \sim \pi_\phi(\cdot \mid s, g)$ and set $\hat{s}' \leftarrow \hat{f}_\omega(s, a_\phi)$
    Update ϕ using $\tilde{c}_\psi(s, a_\phi, \hat{s}') + \gamma d_\theta(\hat{s}', g)$
    $d_{\bar{\theta}} \leftarrow \rho d_\theta + (1 - \rho) d_{\bar{\theta}}$
end for
return d_θ, π_ϕ, and waypoint planner

where $[x]_+ = \max(x, 0)$, and $m \geq 0$ is a slack margin. The full training objective is

$$\mathcal{L}(\theta, \psi) = \mathcal{L}_{\text{Bell}}(\theta) + \beta \mathcal{L}_\triangle(\theta) + \eta \mathcal{L}_{\text{safe}}(\psi).$$

In our implementation, $\hat{\chi}_\psi$ is pretrained from available safe and unsafe evidence and then updated when new failures, interventions, or safety-monitor events are collected. This lets the safety signal improve as the robot encounters new hazards.

4.6 Policy Extraction and Long-Horizon Planning

The learned distance induces an ideal local action-selection rule

$$a^\star(s, g) \in \arg\min_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(\cdot \mid s, a)} \left[\tilde{c}_\psi(s, a, s') + \gamma d_\theta(s', g) \right].$$

For continuous control, we approximate this rule with an actor π_ϕ. We fit a one-step dynamics model $\hat{f}_\omega(s, a)$ on observed replay transitions. For an actor action $a_\phi \sim \pi_\phi(\cdot \mid s, g)$, the model predicts $\hat{s}' = \hat{f}_\omega(s, a_\phi)$, and the actor is trained by minimizing

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{s, g} \left[\tilde{c}_\psi(s, a_\phi, \hat{s}') + \gamma d_\theta(\hat{s}', g) \right].$$

The one-step model is used only to differentiate the continuous-action actor objective. Bellman regression for d_θ always uses the observed replay successor s' rather than a model-predicted successor. At test time, CoGeL operates either directly or through waypoint composition. For short or moderate goals, the local actor $\pi_\phi(\cdot \mid s, g)$ is conditioned directly on the final goal. For longer missions, a lightweight planner selects a waypoint sequence

$$s = w_0, w_1, \ldots, w_N = g$$

using the learned directed distance $d_\theta(w_i, w_j)$ as the edge cost. The local actor then executes each successive waypoint $\pi_\phi(\cdot \mid s_t, w_{i+1})$ until the corresponding waypoint region is reached, after which the next waypoint becomes the local goal. This exposes the compositional structure encouraged by $\mathcal{L}_\triangle$ and allows locally learned goal-reaching behavior to support longer routes.

Because d_θ is learned from observed replay transitions, it is a finite-data, replay-supported approximation to the ideal optimal distance $d_\psi^\star$. Current-policy and exploration rollouts refresh the replay distribution during training, but limited action coverage can still make the learned distance pessimistic. Importantly, these rollouts improve coverage without replacing the observed successor s' in the Bellman target.

5 Experiments

We evaluate whether CoGeL turns local safety evidence into a planning geometry that remains useful over long horizons. The experiments are organized around four questions: **Q1** Does the



Figure 2: **Simulation Scenarios.** Left: navigation with a hidden terrain hazard that is not provided to the policy or planner as a safety map. Right: a directional slope hazard whose risk depends on the traversal direction. CoGeL uses local safety evidence to reshape the learned navigation distance and select safer routes around both types of hazards.

learned safety signal reduce unsafe behavior without destroying goal-reaching efficiency? **Q2** Does the learned distance generalize when test goals are much farther than training goals? **Q3** Do the theory-predicted links between safety-estimation error, distance error, and route consistency appear empirically? **Q4** Is a directed distance important on directional hazards?

Environments. We evaluate CoGeL in both Unity and Isaac-based navigation environments with wheeled and legged robot platforms. The wheeled suite uses Jackal/Husky-style robots in cluttered terrain with hidden low-friction regions that induce stochastic slip, while the legged suite uses a quadruped (Boston Dynamics Spot) on rough terrain with unstable footholds, tilt limits, and fall events. The hazards are not provided to the agent as an explicit safety or cost map; they are revealed only through robot evidence such as successful demonstrations, failed rollouts, safety-monitor triggers, and interventions. Figure 2 shows two example simulation settings, including a hidden terrain hazard and a directional slope hazard. Training goals are sampled within a short local horizon, while test goals span from $1\times$ to $8\times$ the training horizon.

Baselines. We compare CoGeL against representative alternatives: **GCBC**, a goal-conditioned behavior cloning policy based on goal-conditioned supervised learning [32]; **Lag-PPO**, a PPO-Lagrangian constrained policy trained with the same learned safety score [33, 25, 24]; **CRL**, a contrastive goal-conditioned RL baseline [7]; **QRL**, a quasimetric goal-reaching baseline without learned safety shaping [8]; and **Safety+Plan**, which learns the same soft safety signal as CoGeL but uses it only as an external graph-planning penalty rather than changing the learned distance. For fairness, every baseline that requires a safety cost receives the same learned score $\hat{\chi}_\psi$.

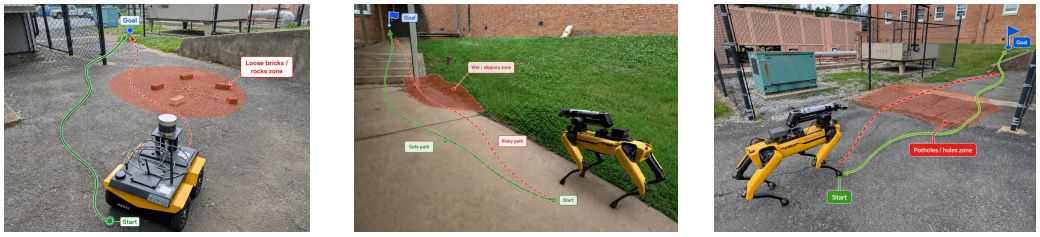
Metrics. We report success rate (**SR**), constraint violations per episode (**CV**), success weighted by path length (**SPL**), and long-horizon success at $4\times$ and $8\times$ the training horizon (**LH-4**$\times$, **LH-8**$\times$). Higher is better for SR, SPL, LH-4$\times$, and LH-8$\times$; lower is better for CV. All numbers are averaged over five random seeds.

Table 1: **Main navigation results.** CoGeL reduces unsafe events while preserving goal-reaching performance and gives the strongest long-horizon success.

	Wheeled navigation					Legged navigation				
Method	SR ↑	CV ↓	SPL ↑	LH-4× ↑	LH-8× ↑	SR ↑	CV ↓	SPL ↑	LH-4× ↑	LH-8× ↑
GCBC	0.79	4.35	0.61	0.34	0.16	0.72	4.70	0.54	0.28	0.12
Lag-PPO	0.80	1.85	0.58	0.45	0.24	0.76	2.15	0.51	0.39	0.20
CRL	0.85	3.05	0.67	0.52	0.31	0.80	3.35	0.60	0.46	0.27
QRL	0.89	2.75	0.73	0.65	0.43	0.84	3.10	0.66	0.58	0.36
Safety+Plan	0.83	1.55	0.64	0.61	0.39	0.79	1.80	0.57	0.53	0.33
CoGeL	**0.91**	**0.72**	**0.76**	**0.84**	**0.76**	**0.87**	**0.95**	**0.70**	**0.78**	**0.68**

Q1: Safety and efficiency. Table 1 evaluates the central tradeoff. A goal-reaching distance without safety shaping can reach nearby goals efficiently, but it often takes short routes through hidden slip or fall regions. CoGeL instead changes the distance used for planning, so risky state-action pairs

become expensive before long-horizon composition occurs. This reduces violations while keeping path efficiency competitive. Figure 3 illustrates the intended CoGeL behavior in these real-world navigation scenes: the safety-blind route takes a shorter path through local hazards, while the CoGeL route avoids loose debris, wet/slippery regions, or pavement holes with a slightly longer but safer path. Figure 4 (right) shows the corresponding safety–efficiency tradeoff as λ varies.



(a) Loose bricks/rocks zone (b) Wet/slippery zone (c) Potholes/holes zone

Figure 3: **Real-world Navigation Scenarios.** The scenes show robot executions in environments containing local hazards that are not provided as hand-specified safety maps. The green trajectory shows the CoGeL route around the risky region, while the red dashed trajectory indicates a shorter unsafe alternative.

Q2: Long-horizon generalization. Figure 4 (left) evaluates success as the test horizon grows beyond the training horizon. CoGeL maintains higher success at $4\times$ and $8\times$ horizons because the learned distance is regularized to compose through intermediate waypoints. In contrast, methods that either lack a compositional distance or apply safety only after planning degrade more quickly as the route length increases.

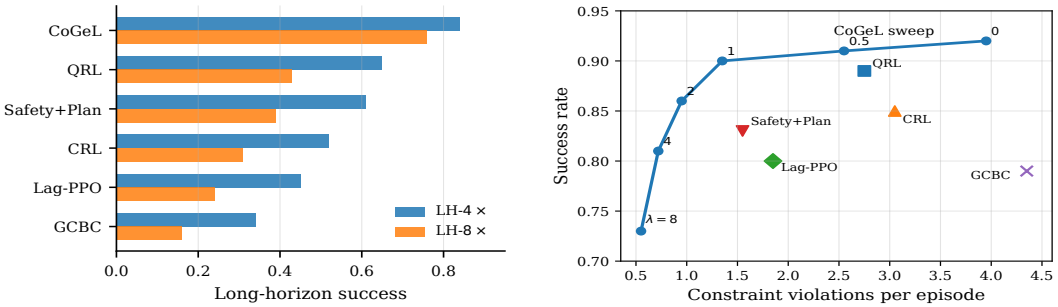


Figure 4: **Long-horizon generalization and safety-efficiency tradeoff.** Left: success at $4\times$ and $8\times$ the training horizon. Right: safety-efficiency tradeoff as the safety weight λ is varied.

Q3: Geometry error and route consistency. Theorem 1 shows that if $\|\hat{\chi}_\psi - \chi^\star\|_\infty \le \epsilon_\chi$, then $\|d^\star_{\hat{\chi}} - d^\star_{\chi^\star}\|_\infty \le \lambda\epsilon_\chi/(1-\gamma)$, so error in the learned safety score induces a controlled change in the learned distance. We test this relationship by injecting label noise into the safety dataset and measuring the resulting distance error against an oracle safety map used only for evaluation. We also measure route-consistency violation as the number of composed waypoint segments increases and compare calibrated predicted violations with observed true violations. Figure 5 shows that distance error grows smoothly with safety-estimation error, route error increases with the number of composed waypoint segments, and the calibrated violation bound remains above the observed violation count. These trends are consistent with Theorem 1, Proposition 1, and Proposition 2, respectively.

Q4: Directed geometry. The symmetric-distance ablation reduces directional-hazard success from 0.84 to 0.54 (Table 7), supporting the use of a directed geometry when traversal risk depends on direction.

Real-robot Evaluation. We additionally evaluate CoGeL in repeated physical navigation trials with a Boston Dynamics Spot and a Clearpath Jackal. Both CoGeL and Safety+Plan use matched start–

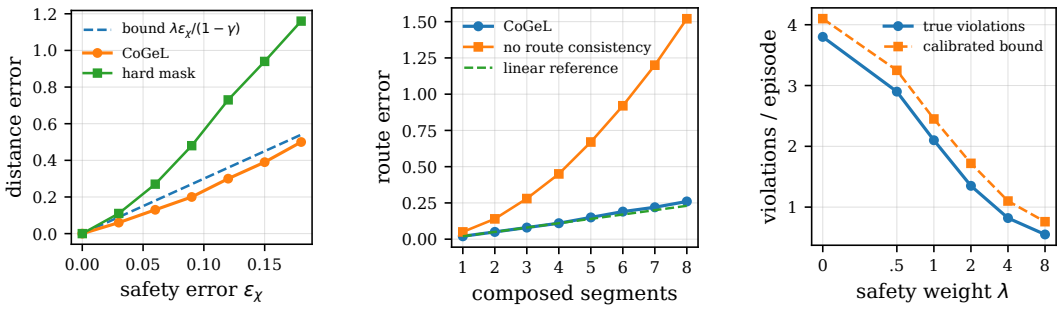


Figure 5: **Empirical checks of the theory.** Left: safety-estimation error yields bounded geometry error. Middle: route error grows with composed waypoint segments. Right: calibrated risk scores bound expected safety violations.

goal conditions and hazard settings. Table 2 shows that CoGeL improves success while reducing the number of observed safety events on both platforms.

Robot	Method	Trials	SR ↑	Safety events/trial ↓
Spot	Safety+Plan	20	0.75	0.45
	CoGeL	20	**0.85**	**0.20**
Jackal	Safety+Plan	20	0.80	0.40
	CoGeL	20	**0.90**	**0.15**

Table 2: **Real-robot Navigation Results.** Success rate (SR) and average safety events per trial over repeated physical navigation trials on Spot and Jackal. CoGeL achieves higher success and fewer safety events than Safety+Plan on both platforms.

6 Limitations and Failure Cases

CoGeL depends on the quality and coverage of the robot evidence used to learn the safety signal. Sparse, biased, or poorly timed labels can cause $\hat{\chi}_\psi(s, a)$ to miss hazards or over-penalize safe behavior; Appendix A.1 describes the labeling process, and Appendix E gives representative failure cases. CoGeL also does not guarantee zero violations. Because safety enters as a soft cost, risky actions may still be selected when λ is too small or when the safety estimator is poorly calibrated, especially under distribution shift; see Appendix A.2 and Appendix D.3. Finally, long-horizon performance depends on the waypoint graph. If useful waypoints or narrow safe corridors are missed, the global route can fail even when the local distance is accurate; Appendix C.7 provides detail.

7 Conclusion

We introduced CoGeL, a constraint-to-geometry learning framework that turns local robot safety evidence into a directed planning distance. CoGeL learns a soft safety signal from demonstrations, failures, and interventions, incorporates it into the Bellman cost, and trains a safety-aware distance that supports both local control and waypoint-based long-horizon planning. Our analysis shows how errors in the learned safety signal affect the resulting geometry, how route consistency supports composition through intermediate waypoints, and how calibrated scores relate to expected violations. Experiments across hidden hazards, directional risks, and long-horizon navigation tasks show that learned safety evidence can directly shape planning rather than remain an external filter. Future work will study richer transition- and history-dependent safety models, uncertainty-aware geometry under distribution shift, and tighter integration with online adaptation and risk-sensitive objectives. We also see an important direction in extending CoGeL to more complex embodied systems, including manipulation and humanoid robots, where safety depends not only on where the robot moves, but also on contact, interaction, and changing physical constraints.

Acknowledgments

This work has been partially supported by NSF CNS EAGER Grant #2233879, NSF IIS Grant #2509680, NSF CAREER Award #1750936, NSF I-Corps Grant #2502886, U.S. Army Grants #W911NF2120076 & #W911NF2410367, and ONR Grant #N00014-23-1-2119.

References

[1] G. Kahn, P. Abbeel, and S. Levine. BADGR: An autonomous self-supervised learning-based navigation system. *IEEE Robotics and Automation Letters*, 6(2):1312–1319, 2021.

[2] D. D. Fan, A.-A. Agha-Mohammadi, and E. A. Theodorou. Learning risk-aware costmaps for traversability in challenging environments. *IEEE robotics and automation letters*, 7(1):279, 2021.

[3] H. Karnan, E. Yang, D. Farkash, G. Warnell, J. Biswas, and P. Stone. STERLING: Self-supervised terrain representation learning from unconstrained robot experience. In *Conference on Robot Learning*, pages 2393–2413. PMLR, 2023.

[4] T. Schaul, D. Horgan, K. Gregor, and D. Silver. Universal value function approximators. In *International Conference on Machine Learning (ICML)*, pages 1312–1320, 2015.

[5] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, P. Abbeel, and W. Zaremba. Hindsight experience replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[6] V. Pong, S. Gu, M. Dalal, and S. Levine. Temporal difference models: Model-free deep RL for model-based control. In *International Conference on Learning Representations (ICLR)*, 2018.

[7] B. Eysenbach, T. Zhang, S. Levine, and R. R. Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. volume 35, pages 35603–35620, 2022.

[8] T. Wang, A. Torralba, P. Isola, and A. Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *International Conference on Machine Learning*, pages 36411–36430. PMLR, 2023.

[9] S. Park, D. Ghosh, B. Eysenbach, and S. Levine. Hiql: Offline goal-conditioned rl with latent states as actions. *Advances in Neural Information Processing Systems*, 36:34866–34891, 2023.

[10] B. Liu, Y. Feng, Q. Liu, and P. Stone. Metric residual networks for sample efficient goal-conditioned reinforcement learning. In *AAAI Conference on Artificial Intelligence*, volume 37, pages 8793–8800, 2023.

[11] J. Hossain and N. Roy. QPRL: Learning optimal policies with quasi-potential functions for asymmetric traversal. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 23828–23845, Vancouver, Canada, July 2025. PMLR.

[12] B. Zheng, V. Myers, B. Eysenbach, and S. Levine. Scaling goal-conditioned reinforcement learning with multistep quasimetric distances. In *The Fourteenth International Conference on Learning Representations*, 2026.

[13] J. Hossain, A.-Z. Faridee, D. E. Asher, J. Freeman, T. Trout, T. Gregory, and N. Roy. QuasiNav: Asymmetric cost-aware navigation planning with constrained quasimetric reinforcement learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2340–2346, Atlanta, GA, USA, May 2025. IEEE.

[14] J. Hossain and N. Roy. Learning anisotropic value geometry with finsler reinforcement learning. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 43950–43983, Seoul, South Korea, July 2026. PMLR.

[15] G. Chou, D. Berenson, and N. Ozay. Learning constraints from demonstrations. In *Workshop on the Algorithmic Foundations of Robotics (WAFR)*, pages 228–245, 2018.

[16] D. Lindner, X. Chen, S. Tschiatschek, K. Hofmann, and A. Krause. Learning safety constraints from demonstrations with unknown rewards. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 2386–2394, 2024.

[17] J. Achiam, D. Held, A. Tamar, and P. Abbeel. Constrained policy optimization. In *International Conference on Machine Learning (ICML)*, pages 22–31, 2017.

[18] L. P. Kaelbling. Learning to achieve goals. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1993.

[19] V. Myers, C. Ji, and B. Eysenbach. Horizon generalization in reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

[20] D. R. Scobee and S. S. Sastry. Maximum likelihood constraint inference for inverse reinforcement learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BJliakStvH.

[21] S. Malik, U. Anwar, A. Aghasi, and A. Ahmed. Inverse constrained reinforcement learning. In *International Conference on Machine Learning (ICML)*, pages 7390–7399, 2021.

[22] K. Kim, G. Swamy, Z. Liu, D. Zhao, S. Choudhury, and S. Z. Wu. Learning shared safety constraints from multi-task demonstrations. *Advances in Neural Information Processing Systems*, 36:5808–5826, 2023.

[23] E. Altman. *Constrained Markov Decision Processes*. Chapman and Hall/CRC, 1999.

[24] A. Stooke, J. Achiam, and P. Abbeel. Responsive safety in reinforcement learning by pid lagrangian methods. In *International Conference on Machine Learning*, pages 9133–9143. PMLR, 2020.

[25] A. Ray, J. Achiam, and D. Amodei. Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 2019.

[26] A. Tamar, Y. Glassner, and S. Mannor. Optimizing the CVaR via sampling. In *AAAI Conference on Artificial Intelligence*, 2015.

[27] Y. Chow, M. Ghavamzadeh, L. Janson, and M. Pavone. Risk-constrained reinforcement learning with percentile risk criteria. *Journal of Machine Learning Research*, 18(167):1–51, 2018. URL http://jmlr.org/papers/v18/15-636.html.

[28] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pages 91–100. PMLR, 08–11 Nov 2022.

[29] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State. Isaac Gym: High performance GPU-based physics simulation for robot learning. *arXiv preprint arXiv:2108.10470*, 2021.

[30] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6):3740–3747, 2023. doi:10.1109/LRA.2023.3270034.

[31] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. ANYmal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadh0143, 2024.

[32] D. Ghosh, A. Gupta, A. Reddy, J. Fu, C. M. Devin, B. Eysenbach, and S. Levine. Learning to reach goals via iterated supervised learning. In *International Conference on Learning Representations*, 2021.

[33] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Appendix

CoGeL: Constraint-to-Geometry Learning from Demonstrations and Failures for Safe Long-Horizon Robot Navigation

A Additional Method Details

A.1 Robot Evidence and Label Construction

CoGeL treats safety supervision as local robot evidence rather than a complete hazard map. Each collected trajectory is stored as

$$\tau = (s_0, a_0, s_1, a_1, \ldots, s_T),$$

together with event information from demonstrations, goal conditioned rollouts, operator interventions, recovery behavior, or onboard safety monitors. Safe evidence comes from expert demonstrations, successful rollouts, or trajectory segments that reach local goals without triggering a safety event. Risk evidence comes from failed rollouts, near misses, emergency stops, excessive slip, collision contacts, unstable footholds, large body tilt, or human interventions.

For each visited state and action pair, CoGeL assigns an evidence label $y_t \in \{0, 1\}$. If a safety event occurs at time t_f, we label a short pre-event labeling window before the event as risky,

$$y_t = 1 \qquad \text{for} \qquad t \in [t_f - K, t_f].$$

This window accounts for delayed event detection because the action that causes slip, collision, or loss of stability may occur before the monitor fires. Segments sufficiently far from a failure or intervention are labeled safe when they remain within nominal state limits and make progress toward the local goal. The resulting dataset is

$$\mathcal{D} = \mathcal{D}_{\text{safe}} \cup \mathcal{D}_{\text{risk}}, \qquad \mathcal{D}_{\text{safe}} = \{(s_t, a_t, 0)\}, \qquad \mathcal{D}_{\text{risk}} = \{(s_t, a_t, 1)\}.$$

In simulation, event labels are produced from hidden environment conditions such as low friction regions, loose debris, holes, contacts, falls, or directional hazards. These hidden hazard variables are used only to generate labels and evaluation metrics. They are not given to the policy or planner as a cost map. In hardware, the same evidence interface uses operator interventions, emergency stop logs, contact events, slip detectors, tilt thresholds, and recovery triggers.

A.2 Safety Estimator and Calibration

The soft safety signal is trained as a probabilistic classifier

$$\hat{\chi}_\psi(s, a) = P_\psi(y = 1 \mid s, a) \in [0, 1].$$

We use binary cross entropy

$$\mathcal{L}_{\text{safe}}(\psi) = -\mathbb{E}_{(s,a,y)\sim\mathcal{D}} \left[y \log \hat{\chi}_\psi(s, a) + (1 - y) \log(1 - \hat{\chi}_\psi(s, a)) \right].$$

The safety estimator has two hidden layers with 256 units in each layer, ReLU activations, and a sigmoid output. Risky examples are less common than safe examples, so we use class balanced mini batches or a weighted binary cross entropy loss when needed.

For calibration, the labeled evidence is split into training, calibration, and held out evaluation data. The training split fits $\hat{\chi}_\psi$. The calibration split fits a scalar temperature on the classifier logits. The held out split and on policy rollouts are used only to report calibration quality. Let $z_\psi(s, a)$ denote the classifier logit. We fit $\tau > 0$ and use

$$\hat{\chi}_{\psi,\tau}(s, a) = \sigma(z_\psi(s, a)/\tau).$$

We report expected calibration error and Brier score in Table 3 and Figure 6.

Table 3: **Calibration quality of the learned safety score.** Expected calibration error and Brier score are measured after temperature scaling. Lower values are better. Values are averaged over five seeds.

Evaluation split	ECE ↓	Brier ↓
Held out evidence	0.038 ± 0.006	0.082 ± 0.009
On policy rollouts	0.051 ± 0.008	0.096 ± 0.011
Shifted hazard layouts	0.073 ± 0.010	0.118 ± 0.014

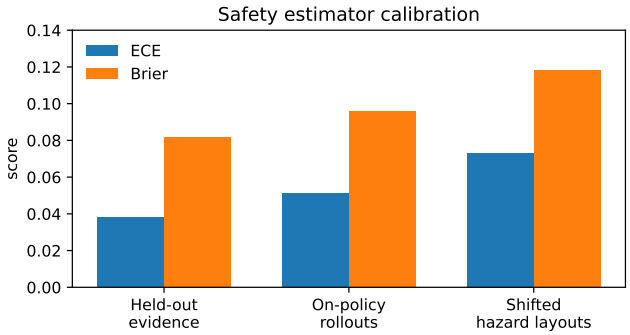


Figure 6: **Safety score calibration.** Calibration is strongest on held out evidence from the training distribution and degrades on shifted hazard layouts. The shifted result shows the importance of reliable calibration when the hazard layout changes.

A.3 Goal Relabeling and Triplet Sampling

The replay buffer stores transitions (s_t, a_t, s_{t+1}) from demonstrations and rollouts. For Bellman training, goals are sampled from a mixture of future states, random states in the same environment, and task goals,

$$g \sim p_{\text{goal}} = \omega_{\text{future}} p_{\text{future}} + \omega_{\text{rand}} p_{\text{rand}} + \omega_{\text{task}} p_{\text{task}}.$$

Future state relabeling improves local goal reaching supervision, while random and task goals expose the distance model to longer range pairs. For route consistency, we sample triplets (s, w, g) where w is a future state between s and g, a graph waypoint, or a nearby state under the current distance estimate. This encourages

$$d_\theta(s, g) \lesssim d_\theta(s, w) + d_\theta(w, g).$$

A.4 Extended Training Algorithm

Algorithm 2 expands Algorithm 1 with the data collection, safety labeling, and continuous control updates used in the experiments.

Bellman regression always uses the observed successor s' from the replay buffer. The one step model $\hat{f}_\omega$ is used only to differentiate the continuous action actor objective. Thus, d_θ remains a finite data, replay supported approximation to the ideal action minimized distance.

B Additional Theory and Proofs

We provide the full proofs for the main theoretical claims. The analysis follows the CoGeL pipeline. The learned safety signal perturbs the one-step Bellman cost. This cost determines the learned directed distance. Route consistency controls waypoint composition, and calibrated safety scores bound expected violations.

Algorithm 2 CoGeL with Robot Evidence Collection

Require: Initial safe evidence $\mathcal{D}_{\text{safe}}$, initial risky evidence $\mathcal{D}_{\text{risk}}$, replay buffer $\mathcal{B}$, safety weight λ, consistency weight β, pre-event labeling window K, target update rate ρ

$\mathcal{D} \leftarrow \mathcal{D}_{\text{safe}} \cup \mathcal{D}_{\text{risk}}$
Initialize $\hat{\chi}_\psi$, d_θ, $d_{\bar{\theta}} \leftarrow d_\theta$, π_ϕ, and $\hat{f}_\omega$
Train $\hat{\chi}_\psi$ on the available evidence $\mathcal{D}$
for each training round **do**
    Collect goal conditioned rollouts using π_ϕ or an exploration policy
    Store observed transitions (s_t, a_t, s_{t+1}) in $\mathcal{B}$
    if a failure, intervention, near miss, or safety monitor event occurs at t_f **then**
        Add $(s_t, a_t, 1)$ to $\mathcal{D}_{\text{risk}}$ for $t \in [t_f - K, t_f]$
    end if
    Add event free segments sufficiently far from safety events to $\mathcal{D}_{\text{safe}}$
    Update $\hat{\chi}_\psi$ on $\mathcal{D}_{\text{safe}} \cup \mathcal{D}_{\text{risk}}$
    for each gradient step **do**
        Sample observed (s, a, s') and relabeled goal g from $\mathcal{B}$
        $\tilde{c}_\psi(s, a, s') \leftarrow c(s, a, s') + \lambda \hat{\chi}_\psi(s, a)$
        $y \leftarrow \tilde{c}_\psi(s, a, s') + \gamma d_{\bar{\theta}}(s', g)$ using the observed successor s'
        Sample (s, w, g) and update θ using $\mathcal{L}_{\text{Bell}} + \beta \mathcal{L}_\triangle$
        Update ω using $\|\hat{f}_\omega(s, a) - s'\|_2^2$
        Sample $a_\phi \sim \pi_\phi(\cdot \mid s, g)$ and set $\hat{s}' = \hat{f}_\omega(s, a_\phi)$
        Update ϕ using $\tilde{c}_\psi(s, a_\phi, \hat{s}') + \gamma d_\theta(\hat{s}', g)$
        $d_{\bar{\theta}} \leftarrow \rho d_\theta + (1 - \rho) d_{\bar{\theta}}$
    end for
end for
return d_θ, π_ϕ, and the waypoint planner

B.1 Bellman Contraction

Lemma 1 (Bellman contraction)**.** *Assume* $0 \le c(s, a, s') \le c_0$*,* $\hat{\chi}_\psi(s, a) \in [0, 1]$*, and* $\gamma \in [0, 1)$*. Then* $0 \le \tilde{c}_\psi(s, a, s') \le c_{\max} = c_0 + \lambda$*, and the operator*

$$(T_\psi d)(s, g) = \min_{a \in \mathcal{A}} \mathbb{E}_{s' \sim P(\cdot|s,a)} \left[\tilde{c}_\psi(s, a, s') + \gamma d(s', g)\right]$$

is a γ*-contraction in the sup norm:*

$$\|T_\psi d_1 - T_\psi d_2\|_\infty \le \gamma \|d_1 - d_2\|_\infty.$$

Therefore T_ψ *has a unique fixed point* $d_\psi^\star$*, value iteration converges geometrically to it, and*

$$\|d_\psi^\star\|_\infty \le \frac{c_{\max}}{1 - \gamma}.$$

Proof. For any s, g, the only term that depends on the current distance estimate is $\gamma d(s', g)$. For a fixed action a,

$$\left|\mathbb{E}\left[\tilde{c}_\psi(s, a, s') + \gamma d_1(s', g)\right] - \mathbb{E}\left[\tilde{c}_\psi(s, a, s') + \gamma d_2(s', g)\right]\right| \le \gamma \|d_1 - d_2\|_\infty.$$

Taking a minimum over actions is non-expansive, so the same inequality holds after applying T_ψ. The fixed-point result follows from Banach's fixed point theorem. The value bound follows by summing the bounded discounted cost:

$$\sum_{t=0}^{\infty} \gamma^t c_{\max} = \frac{c_{\max}}{1 - \gamma}.$$

□

B.2 Safety Estimation Error Controls Geometry Error

Theorem 1 (Safety-estimation error controls geometry error)**.** *Let $d^\star_{\hat{\chi}}$ be the fixed point induced by $\tilde{c}_{\hat{\chi}} = c + \lambda\hat{\chi}_\psi$, and let $d^\star_{\chi^\star}$ be the fixed point induced by the true safety cost $\tilde{c}_{\chi^\star} = c + \lambda\chi^\star$. If*

$$\|\hat{\chi}_\psi - \chi^\star\|_\infty \le \epsilon_\chi,$$

then

$$\|d^\star_{\hat{\chi}} - d^\star_{\chi^\star}\|_\infty \le \frac{\lambda}{1-\gamma}\epsilon_\chi.$$

Proof. Let $T_{\hat{\chi}}$ and $T_{\chi^\star}$ be the Bellman operators induced by $\tilde{c}_{\hat{\chi}}$ and $\tilde{c}_{\chi^\star}$. The one-step cost difference is bounded by

$$\|\tilde{c}_{\hat{\chi}} - \tilde{c}_{\chi^\star}\|_\infty = \lambda\|\hat{\chi}_\psi - \chi^\star\|_\infty \le \lambda\epsilon_\chi.$$

Using the fixed-point equations,

$$d^\star_{\hat{\chi}} = T_{\hat{\chi}} d^\star_{\hat{\chi}}, \qquad d^\star_{\chi^\star} = T_{\chi^\star} d^\star_{\chi^\star},$$

we obtain

$$\begin{aligned}\|d^\star_{\hat{\chi}} - d^\star_{\chi^\star}\|_\infty &= \|T_{\hat{\chi}} d^\star_{\hat{\chi}} - T_{\chi^\star} d^\star_{\chi^\star}\|_\infty \\ &\le \|T_{\hat{\chi}} d^\star_{\hat{\chi}} - T_{\hat{\chi}} d^\star_{\chi^\star}\|_\infty + \|T_{\hat{\chi}} d^\star_{\chi^\star} - T_{\chi^\star} d^\star_{\chi^\star}\|_\infty \\ &\le \gamma\|d^\star_{\hat{\chi}} - d^\star_{\chi^\star}\|_\infty + \lambda\epsilon_\chi.\end{aligned}$$

Rearranging gives

$$\|d^\star_{\hat{\chi}} - d^\star_{\chi^\star}\|_\infty \le \frac{\lambda}{1-\gamma}\epsilon_\chi.$$

□

B.3 Waypoint Composition

Proposition 1 (Waypoint composition)**.** *Suppose the learned distance has maximum route-consistency violation*

$$\epsilon_\triangle = \sup_{s,w,g} \big[d_\theta(s,g) - d_\theta(s,w) - d_\theta(w,g)\big]_+.$$

For any waypoint sequence $w_0 = s, w_1, \ldots, w_N = g$,

$$d_\theta(s,g) \le \sum_{i=0}^{N-1} d_\theta(w_i, w_{i+1}) + (N-1)\epsilon_\triangle.$$

Proof. By definition of $\epsilon_\triangle$, for any s, w, g,

$$d_\theta(s,g) \le d_\theta(s,w) + d_\theta(w,g) + \epsilon_\triangle.$$

Apply this inequality first with $w = w_1$:

$$d_\theta(w_0, w_N) \le d_\theta(w_0, w_1) + d_\theta(w_1, w_N) + \epsilon_\triangle.$$

Then apply the same inequality to $d_\theta(w_1, w_N)$, and continue recursively. After $N-1$ applications,

$$d_\theta(s,g) \le \sum_{i=0}^{N-1} d_\theta(w_i, w_{i+1}) + (N-1)\epsilon_\triangle.$$

□

B.4 Calibrated Scores Bound Expected Violations

Proposition 2 (Calibrated scores bound expected violations)**.** *Assume that along the states visited by a policy π,*

$$\mathbb{E}[\chi^\star(s,a) \mid \hat{\chi}_\psi(s,a)] \le \hat{\chi}_\psi(s,a) + \epsilon_{\text{cal}}.$$

Then over trajectories of length at most $T_{\max}$,

$$\mathbb{E}_\pi\left[\sum_{t=0}^{\tau_g-1} \chi^\star(s_t,a_t)\right] \le \frac{1}{\lambda}\mathbb{E}_\pi\left[\sum_{t=0}^{\tau_g-1} \tilde{c}_\psi(s_t,a_t,s_{t+1})\right] + T_{\max}\epsilon_{\text{cal}}.$$

Proof. By calibration,

$$\mathbb{E}[\chi^\star(s_t,a_t)] \le \mathbb{E}[\hat{\chi}_\psi(s_t,a_t)] + \epsilon_{\text{cal}}$$

for each visited state-action pair. Summing over a trajectory of length at most $T_{\max}$ gives

$$\mathbb{E}_\pi\left[\sum_{t=0}^{\tau_g-1} \chi^\star(s_t,a_t)\right] \le \mathbb{E}_\pi\left[\sum_{t=0}^{\tau_g-1} \hat{\chi}_\psi(s_t,a_t)\right] + T_{\max}\epsilon_{\text{cal}}.$$

Since $c(s,a,s') \ge 0$,

$$\tilde{c}_\psi(s,a,s') = c(s,a,s') + \lambda\hat{\chi}_\psi(s,a) \ge \lambda\hat{\chi}_\psi(s,a).$$

Therefore,

$$\sum_{t=0}^{\tau_g-1} \hat{\chi}_\psi(s_t,a_t) \le \frac{1}{\lambda}\sum_{t=0}^{\tau_g-1} \tilde{c}_\psi(s_t,a_t,s_{t+1}).$$

Combining the two inequalities proves the result. □

B.5 Directed Geometry for Directional Hazards

Many robot hazards are not symmetric. Entering a slippery patch, climbing onto loose debris, or stepping into an unstable foothold can be more costly than leaving the same region. In such cases, the constraint-aware cost-to-go is directional:

$$d^\star(u,v) \ne d^\star(v,u).$$

A symmetric metric must assign the same value to both directions, so it cannot represent this asymmetry even with more data. Figure 7 illustrates this setting: entering the hazard has high cost, while leaving the same hazard has lower cost.

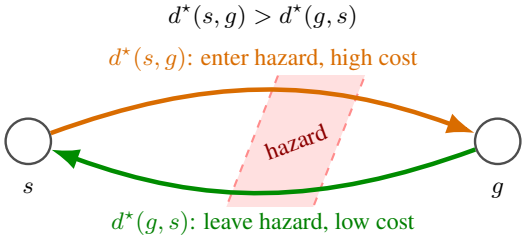


Figure 7: **Directional safety.** Entering a hazard can cost more than leaving it, yielding $d^\star(s,g) \ne d^\star(g,s)$.

The following proposition formalizes this limitation. If the directional cost gap is $\Delta > 0$, then any symmetric model has error at least $\Delta/2$ in one of the two directions. CoGeL avoids this modeling floor by learning a directed quasimetric distance rather than a symmetric metric. The empirical counterpart is the symmetric-distance ablation in Table 7, where removing directionality causes the largest degradation on directional-hazard environments.

Proposition 3 (Symmetric metrics have irreducible error on directional hazards)**.** *Suppose two states u, v have true directed costs satisfying*

$$\Delta = |d^\star(u, v) - d^\star(v, u)| > 0.$$

For any symmetric metric m with $m(u, v) = m(v, u)$,

$$\max\{|m(u, v) - d^\star(u, v)|,\ |m(v, u) - d^\star(v, u)|\} \geq \frac{\Delta}{2}.$$

Proof. Let $x = m(u, v) = m(v, u)$, $a = d^\star(u, v)$, and $b = d^\star(v, u)$. Since $|a - b| = \Delta$, no single scalar x can be within less than $\Delta/2$ of both a and b. Otherwise, the triangle inequality on the real line gives

$$\Delta = |a - b| \leq |a - x| + |x - b| < \frac{\Delta}{2} + \frac{\Delta}{2} = \Delta,$$

which is impossible. Thus at least one direction has error at least $\Delta/2$. □

C Experimental and Implementation Details

C.1 Simulation Environments and Robot Platforms

We use both Unity and NVIDIA Isaac based navigation environments with a common goal conditioned interface. The wheeled suite uses Jackal and Husky style differential drive robots in cluttered terrain with hidden low friction regions, loose debris, obstacle contacts, and pothole like discontinuities. The legged suite uses a quadruped on rough terrain with elevation changes, unstable footholds, wet or slippery surfaces, tilt limits, and fall events. We also use directional hazard settings where the expected violation cost depends on the direction of traversal. Hazard locations are never provided to the policy or planner as a safety map. They appear only through local evidence collected during demonstrations, rollouts, interventions, and safety monitor events.

For the wheeled setting, a slip event above the configured displacement threshold, an obstacle contact, a stuck event, or an intervention is recorded as a violation. For the legged setting, falls, excessive body tilt, unstable footholds, recovery triggers, and interventions are recorded as violations. Directional hazards include slopes and one way friction regions where entering the region can be more costly than leaving it.

Table 4: **Environment summary.** Hazard maps are hidden from the policy and planner. The agent receives robot state, local perception, and local safety evidence.

Suite	Robot	Hazards	Safety events
Wheeled navigation	Jackal or Husky style UGV	slip patches, debris, potholes	collision, slip, stuck, intervention
Legged navigation	quadruped	wet surfaces, unstable footholds, holes	fall, large tilt, slip, recovery trigger
Directional hazards	UGV or quadruped	slopes, one way friction, approach angle risk	direction dependent violation
Long horizon routes	UGV or quadruped	mixed hidden hazards	accumulated violations, timeout

C.2 Observation and Action Spaces

For wheeled navigation, the state includes robot pose, body frame velocity, local goal direction, local terrain features, and short range obstacle or debris features. The action is a velocity command $a = (v, \omega)$, where v is forward velocity and ω is yaw rate. For legged navigation, the state includes base pose, base velocity, body orientation, goal direction, local terrain features, and contact or stability indicators. The high level action is a locomotion command such as $a = (v_x, v_y, \omega)$, which is executed by a low level controller.

C.3 Task Costs and Safety Events

All methods use the same base task cost. For a transition (s_t, a_t, s_{t+1}),

$$c(s_t, a_t, s_{t+1}) = w_{\text{step}} + w_g \|p_{t+1} - p_g\|_2 + w_u \|a_t\|_2^2 + w_{\text{obs}} \mathbf{1}\{\text{contact}_t\}.$$

The terms encourage short routes and progress while penalizing aggressive commands and obstacle contacts. If the robot does not reach $B(g)$ by $T_{\max}$, it receives the terminal failure cost c_{fail}.

The evaluation violation indicator is

$$\chi^\star(s_t, a_t) = \mathbf{1}\{\text{collision} \vee \text{slip} \vee \text{fall} \vee \text{large tilt} \vee \text{intervention}\}.$$

These violations are used for evaluation and for local evidence labels. The planner does not receive the hidden hazard map.

CoGeL uses

$$\tilde{c}_\psi(s_t, a_t, s_{t+1}) = c(s_t, a_t, s_{t+1}) + \lambda \hat{\chi}_\psi(s_t, a_t).$$

The safety weight λ has the same units as the one step task cost because $\hat{\chi}_\psi \in [0, 1]$. If the base task cost is rescaled, λ should be retuned.

Table 5: **Task cost coefficients.** The values are normalized so that the learned safety penalty is comparable to the local motion and progress terms.

Term	Value	Role
w_{step}	0.05	discourages unnecessarily long routes
w_g	1.00	encourages progress toward the goal
w_u	0.01	discourages aggressive commands
w_{obs}	2.00	penalizes contact events
c_{fail}	10.00	terminal failure cost
λ	1.00	default learned safety penalty weight

C.4 Training and Evaluation

Training goals are sampled within a short local horizon. Evaluation goals are sampled at $1\times$, $2\times$, $4\times$, and $8\times$ the training horizon. Each seed is trained for 1.0 million environment interactions. The learning curves generally stabilize after about 0.6 to 0.8 million interactions. All reported values are averaged over five seeds.

Methods that require a safety signal receive the same learned score $\hat{\chi}_\psi$. The methods also use the same evidence budget, replay buffer size, goal sampling choices, and evaluation seeds. Planner based comparisons use the same waypoint graph, search procedure, node density, and local goal horizon. This keeps the comparison focused on how each method uses the same local safety evidence.

C.5 Metrics

We report success rate

$$\text{SR} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{1}\{\text{episode } i \text{ reaches the goal}\},$$

constraint violations per episode

$$\text{CV} = \frac{1}{N} \sum_{i=1}^{N} \sum_{t=0}^{\tau_g^{(i)}-1} \chi^\star(s_t^{(i)}, a_t^{(i)}),$$

and success weighted by path length

$$\text{SPL} = \frac{1}{N} \sum_{i=1}^{N} S_i \frac{\ell_i^\star}{\max(\ell_i, \ell_i^\star)}.$$

Here S_i is the success indicator, ℓ_i is the executed path length, and $\ell_i^\star$ is the shortest feasible reference path length. We also report success at $4\times$ and $8\times$ the training horizon.

C.6 Networks and Optimization

We use multilayer perceptrons for the learned components. The safety estimator takes (s, a) and outputs a violation likelihood through a sigmoid head. It has two hidden layers with 256 units in each layer and ReLU activations. The directed distance model $d_\theta(s, g)$ uses separate state and goal encoders followed by an MLP head and a softplus output. Separate encoders allow $d_\theta(s, g)$ and $d_\theta(g, s)$ to take different values.

For continuous control, the actor $\pi_\phi(a \mid s, g)$ uses the same state and goal representation and outputs a squashed Gaussian action distribution. The one step model $\hat{f}_\omega(s, a)$ predicts the next state from an observed state and action. It is trained on replay transitions with

$$\mathcal{L}_{\text{dyn}}(\omega) = \mathbb{E}_{(s,a,s')\sim\mathcal{B}} \left[\|\hat{f}_\omega(s, a) - s'\|_2^2 \right].$$

For an actor action $a_\phi \sim \pi_\phi(\cdot \mid s, g)$, we set $\hat{s}' = \hat{f}_\omega(s, a_\phi)$ and optimize

$$\mathcal{L}_\pi(\phi) = \mathbb{E}_{s,g} \left[\tilde{c}_\psi(s, a_\phi, \hat{s}') + \gamma d_\theta(\hat{s}', g) \right].$$

The one-step model is used only for the actor update. The Bellman target for d_θ always uses the observed replay successor s'.

C.7 Waypoint Planner

For long horizon navigation, CoGeL builds a directed graph $\mathcal{G} = (V, E)$ over feasible waypoint states. Each node $v_i \in V$ represents a candidate waypoint, and the learned distance defines the basic directed edge cost

$$w_0(i, j) = d_\theta(v_i, v_j).$$

Because d_θ is directed, the cost of moving from v_i to v_j need not equal the cost of moving from v_j to v_i. The safety evidence has already affected this edge cost through the Bellman learning of d_θ.

Unless otherwise stated, the reported CoGeL results use $d_\theta(v_i, v_j)$ as the graph edge cost without an additional edge level penalty.

For deployment, we optionally consider an additional edge level safety guard when a short local rollout ξ_{ij} is available. We compute the maximum predicted violation likelihood along the rollout as

$$\hat{q}_\psi(i, j) = \max_{(s,a)\in\xi_{ij}} \hat{\chi}_\psi(s, a).$$

The planner may then use

$$w(i, j) = d_\theta(v_i, v_j) + \kappa \hat{q}_\psi(i, j),$$

where $\kappa \geq 0$ controls the additional penalty assigned to an edge with high predicted risk. An edge may also be removed when $\hat{q}_\psi(i, j)$ exceeds a conservative threshold.

This optional guard is separate from the main constraint to geometry mechanism. In CoGeL, the learned safety signal has already shaped d_θ during Bellman learning. The additional term $\kappa\hat{q}_\psi$ is only an optional deployment check when a more conservative edge level decision is desired.

Figure 8 illustrates the planning process. The direct route from the start toward the goal is shorter, but it passes through a region with high predicted risk. The selected route instead moves through w_1 and w_2, avoiding the risky region at the cost of a slightly longer path.

The planner applies Dijkstra or A* to the directed graph and selects a waypoint sequence

$$s = w_0, w_1, \ldots, w_N = g.$$

The local actor executes $\pi_\phi(\cdot \mid s_t, w_{i+1})$ until the next waypoint region is reached.

Figure 9 illustrates the waypoint planning process. The green route follows intermediate waypoints around a region with high predicted risk, while the red dashed connection shows a shorter but riskier alternative.

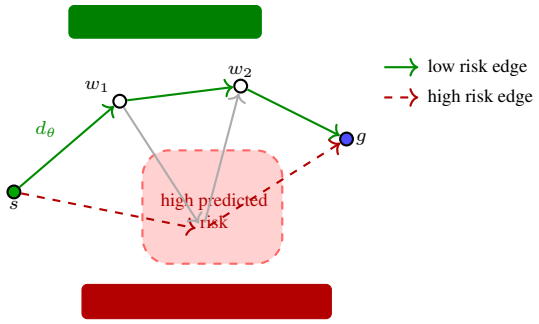


Figure 8: **Waypoint planning with a safety aware directed distance.** CoGeL builds a graph over feasible waypoints and uses the learned directed distance $d_\theta(v_i, v_j)$ as the main edge cost. This distance already reflects the learned safety signal through Bellman training. When a local rollout ξ_{ij} is available, an optional deployment guard can further penalize an edge using its predicted violation likelihood $\hat{q}_\psi(i, j)$. The green path shows the selected waypoint route, while the red dashed path shows a shorter route through a region with high predicted risk.

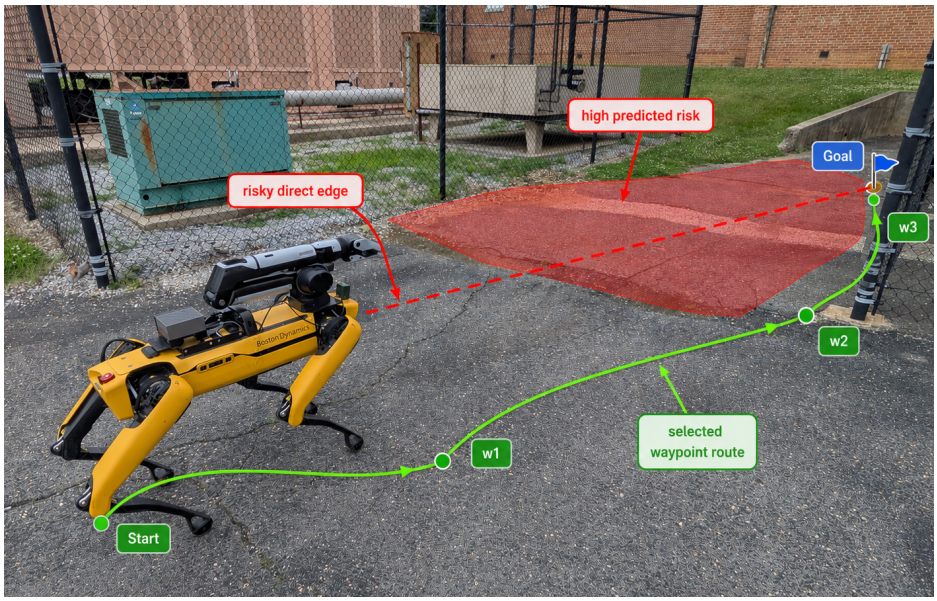


Figure 9: **Waypoint planning with the learned directed distance.** The green path shows the selected route through intermediate waypoints. The red dashed connection crosses a region with high predicted risk. The learned geometry makes the safer route preferable even when the direct route is shorter.

C.8 Training Settings

All methods are trained with the same replay buffer capacity, goal-sampling procedure, evidence budget, and evaluation seeds unless a baseline requires its own optimizer-specific parameters. For methods that use a safety signal, we provide the same learned violation likelihood $\hat{\chi}_\psi$ so that the comparison focuses on how the safety evidence is used rather than on differences in data collection or label quality. Each seed is trained for 1.0×10^6 environment interactions. The learning curves typically begin to stabilize after roughly 0.6–0.8×10^6 interactions. Table 6 summarizes the default settings used throughout the experiments.

Table 6: **Default training settings.**

Quantity	Value
Number of seeds	5
Discount factor γ	0.99
Replay buffer capacity	10^6 transitions
Batch size	256
Optimizer	Adam
Learning rate	3×10^{-4}
Target update rate ρ	5×10^{-3}
Distance network	2 hidden layers, 256 units each
Safety estimator	2 hidden layers, 256 units each, sigmoid head
Default safety weight λ	1.0
Route consistency weight β	0.1
Route slack margin m	0
Pre-event labeling window K	5 steps
Evaluation horizons	$1\times, 2\times, 4\times, 8\times$ training horizon
Training interactions per seed	1.0×10^6

D Additional Experiments and Ablations

D.1 Stair Ascent with a Local Slippery Region

Figure 10 shows a stair ascent setting with a local slippery region near the base of the stairs. A direct route to the goal passes over the floor mat and can reduce foothold stability during the transition from flat ground to the stairs. CoGeL assigns a higher learned cost to this region and selects an entry path on the right side. The example shows how local safety evidence changes the planning distance and can make a longer route preferable.

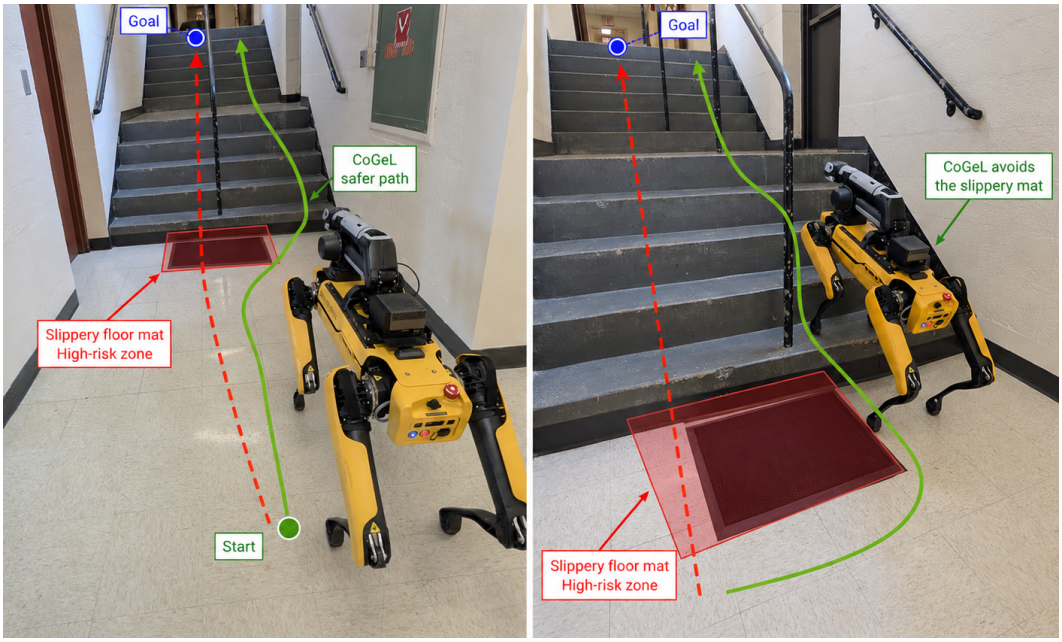


Figure 10: **Stair ascent with a local slippery hazard.** The red dashed path follows the shorter route over the slippery floor mat. The green path shows the CoGeL route around the risky region before the robot enters the stairs.

D.2 Main Ablations

No safety term sets $\lambda = 0$, which reduces CoGeL to directed distance learning without safety shaping.

Hard safety mask thresholds $\hat{\chi}_\psi(s, a)$ and removes actions or edges above the threshold.

No route consistency removes $\mathcal{L}_\triangle$.

Symmetric distance replaces the directed distance with a symmetric metric.

No waypoint planner applies the local goal conditioned policy directly to long horizon goals.

Table 7 summarizes the effect of each component. Removing the safety term increases constraint violations, removing route consistency reduces long horizon success, and replacing the directed distance with a symmetric metric gives the largest drop on directional hazards.

Table 7: **Ablation results.** The safety term reduces violations, route consistency improves long horizon composition, and the directed distance improves performance on directional hazards.

Variant	SR ↑	CV ↓	LH-8× ↑	Directional hazard SR ↑
CoGeL	**0.91**	**0.72**	**0.76**	**0.84**
No safety term ($\lambda = 0$)	0.92	3.95	0.58	0.67
Hard safety mask	0.79	0.88	0.49	0.56
No route consistency	0.88	0.86	0.51	0.78
Symmetric distance	0.84	1.18	0.61	0.54
No waypoint planner	0.86	0.93	0.44	0.76

D.3 Sensitivity Analysis

We study the safety weight λ, the pre-event labeling window K, and the route consistency weight β. Small values of λ can leave risky actions under penalized. Large values reduce violations but can make the policy conservative. A short K can miss the action that caused a safety event, while a long K can label nearby safe behavior as risky. The weight β controls how strongly the distance is encouraged to compose through intermediate waypoints.

Table 8 summarizes these trends. Moderate values of λ, K, and β strike the best balance between safety, nominal success, and long-horizon performance.

Table 8: **Sensitivity to key choices.** Moderate values give the best balance between safety and long horizon success. Values are averaged over five seeds.

Setting	**SR ↑**	**CV ↓**	**LH-8× ↑**
Default ($\lambda = 1, K = 5, \beta = 0.1$)	0.91 ± 0.02	0.72 ± 0.12	0.76 ± 0.04
Low λ (0.25)	0.93 ± 0.02	2.40 ± 0.31	0.62 ± 0.05
High λ (4.0)	0.85 ± 0.03	0.45 ± 0.08	0.70 ± 0.04
Small K (1)	0.90 ± 0.03	1.20 ± 0.18	0.69 ± 0.05
Large K (10)	0.84 ± 0.04	0.60 ± 0.10	0.63 ± 0.06
Small β (0.01)	0.89 ± 0.03	0.78 ± 0.13	0.53 ± 0.05
Large β (1.0)	0.87 ± 0.03	0.69 ± 0.11	0.67 ± 0.05

D.4 Triplet Sampling

The route-consistency loss depends on triplets (s, w, g), where w serves as an intermediate state between the start s and goal g. We sample these triplets from several sources so that the learned distance is exposed to both local trajectory structure and the waypoint configurations used during long-horizon planning.

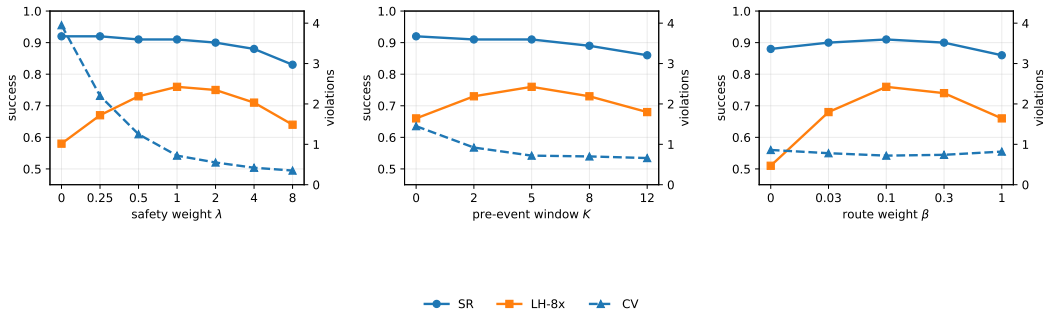


Figure 11: **Sensitivity to the main design choices.** Success rate and LH-8× use the left axis. Constraint violations use the right axis. The default values $\lambda = 1$, $K = 5$, and $\beta = 0.1$ provide a balanced setting.

Future-state triplets choose w and g from later states in the same trajectory. Graph-waypoint triplets use intermediate nodes from the waypoint graph. Random-nearby triplets select states that are spatially or geometrically close under the current representation. Our default setting mixes these sources during training.

Table 9 compares the different choices. Using only one source reduces the diversity of the consistency constraints, while mixed sampling gives the best route consistency and long-horizon success.

Table 9: **Triplet sampling ablation.** Mixed triplet sampling performs best because it includes both trajectory level composition and graph waypoint composition. Values are averaged over five seeds.

Triplet source	**SR** ↑	**Route error** ↓	**LH-8×** ↑
Future state only	0.89 ± 0.03	0.42 ± 0.05	0.64 ± 0.05
Graph waypoint only	0.88 ± 0.03	0.36 ± 0.04	0.69 ± 0.04
Random nearby only	0.86 ± 0.04	0.51 ± 0.06	0.58 ± 0.06
Mixed triplets	0.91 ± 0.02	0.24 ± 0.03	0.76 ± 0.04

D.5 Transition and History Dependent Safety Models

The main method uses the safety score $\hat{\chi}_\psi(s, a)$. Some hazards are better described by the next state or recent history. Examples include delayed slip, repeated obstacle contact, foothold collapse after load transfer, and interactions that depend on recent motion.

The same objective can use a transition based score $\hat{\chi}_\psi(s, a, s')$ or a recurrent score $\hat{\chi}_\psi(h_t, a_t)$, where h_t summarizes recent history. The augmented costs become

$$\tilde{c}_\psi(s_t, a_t, s_{t+1}) = c(s_t, a_t, s_{t+1}) + \lambda \hat{\chi}_\psi(s_t, a_t, s_{t+1})$$

and

$$\tilde{c}_\psi(h_t, a_t, s_{t+1}) = c(s_t, a_t, s_{t+1}) + \lambda \hat{\chi}_\psi(h_t, a_t).$$

The Bellman objective keeps the same form as long as the augmented cost is bounded.

D.6 Training Progress

Figure 12 shows success rate over environment interaction. Each curve is averaged over five seeds, and the shaded band shows one standard deviation. The long horizon and confined stepping settings improve more slowly because success depends on both local control and waypoint composition.

Figure 13 shows the corresponding violation trend. Methods that keep safety outside the learned distance retain higher violation counts in the longer tasks.

Table 10: **Safety score variants.** Transition and history based models reduce violations in these settings, while the state and action model remains the simplest default. Values are averaged over five seeds.

Safety model	**SR** ↑	**CV** ↓	**LH-8**× ↑
State and action $\hat{\chi}_\psi(s, a)$	0.91 ± 0.02	0.72 ± 0.12	0.76 ± 0.04
Transition based $\hat{\chi}_\psi(s, a, s')$	0.92 ± 0.02	0.65 ± 0.10	0.78 ± 0.04
History based $\hat{\chi}_\psi(h_t, a_t)$	0.90 ± 0.03	0.62 ± 0.09	0.77 ± 0.05

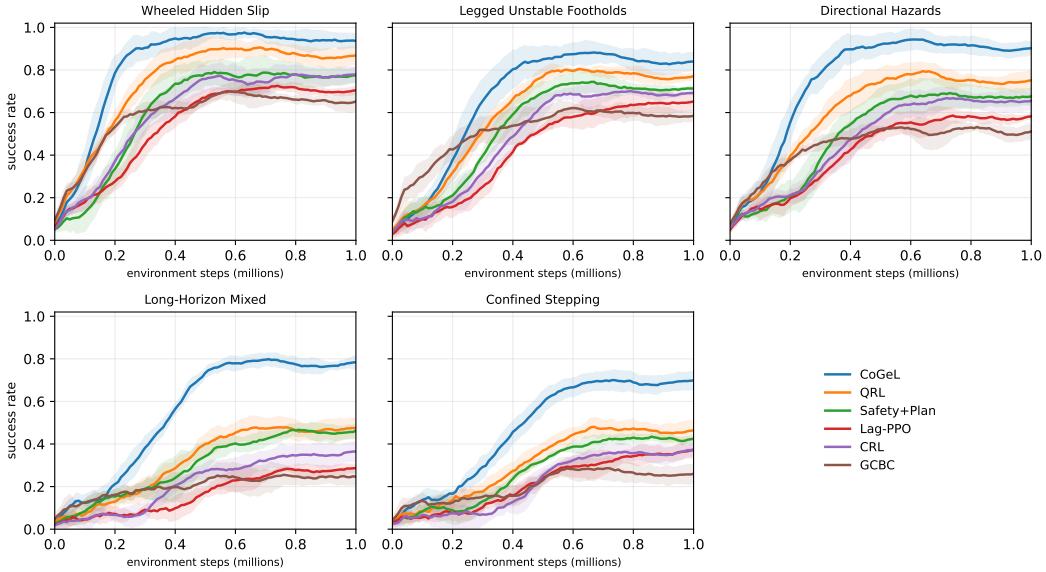


Figure 12: **Success rate learning curves.** Curves show the mean over five seeds with one standard deviation. CoGeL reaches higher final success in the directional, long horizon, and confined stepping settings.

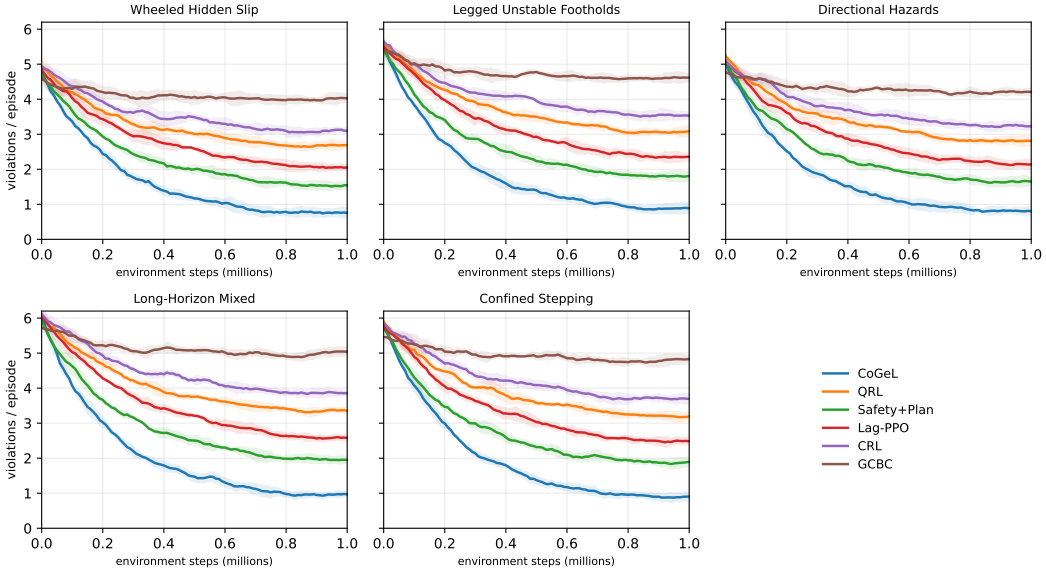


Figure 13: **Violation learning curves.** Violations per episode are shown over environment interaction. CoGeL reaches lower violation rates than the safety blind distance baseline and the planner side safety baseline, especially when early local choices affect later waypoint composition.

E Additional Failure Cases

CoGeL can fail when the collected evidence does not cover the relevant hazard boundary. For example, if the robot observes a directional hazard only from one traversal direction, the learned safety score may underestimate risk in the reverse direction. The method can also become conservative when the pre-event labeling window assigns risk to nearby states that were not unsafe. This can make narrow safe passages look risky and lead to unnecessary detours.

A second failure mode comes from distribution shift. If terrain friction, contact dynamics, lighting, payload, or robot calibration changes substantially, the learned safety score may no longer be well calibrated. The learned distance can then favor unsafe shortcuts or avoid regions that are actually traversable.

Long horizon performance also depends on the waypoint graph. If the graph misses a narrow safe corridor or includes waypoints in poorly estimated regions, the local distance may be accurate while the global route still fails.

Figure 14 shows a confined stepping failure. The passage is physically traversable, but the feasible foothold region is small. A foot placed near the block boundary or outside the support region can produce unstable contact and stop the maneuver. This case shows that the learned planning geometry does not remove the need for accurate local foothold selection and contact control.



Figure 14: **Failure in a confined stepping setting.** The robot attempts to move through the narrow space between separated stepping blocks. The maneuver can fail when a foot lands near the block boundary or outside the support region.

F Notes on Related Baselines

Constrained waypoint search. Safety+Plan uses the same learned safety score as CoGeL but applies it only during graph planning. The distance itself is trained without the safety term. A related alternative would train separate distance and safety critics and use the safety critic to penalize or prune graph edges. CoGeL differs because the learned safety score enters the Bellman cost before the distance is learned.

Risk sensitive reinforcement learning. CVaR and distributional critics place more weight on rare but severe failures. These methods change the risk objective used for optimization. CoGeL instead studies how local safety evidence changes the directed distance used for long horizon planning. A useful extension is to study both effects together so that the geometry reflects direction dependent cost and tail risk.

G Robot Platforms and Hardware Details

The simulation experiments use wheeled and legged robot families, and the main paper also reports physical navigation trials on a Clearpath Jackal and a Boston Dynamics Spot. The physical trials use matched start and goal conditions and matched hazard settings for CoGeL and Safety+Plan. The quantitative results are reported in Table 2.

G.1 Wheeled Robot Family

The wheeled simulation suite uses a differential drive robot at the scale of Jackal and Husky unmanned ground vehicles. The action is

$$a = (v, \omega),$$

where v is forward velocity and ω is yaw rate. Hidden low friction regions, loose debris, and pothole like regions produce slip, contact, or stuck events.

The physical wheeled trials use a Clearpath Jackal. The logging interface records odometry, laser scan, IMU, commanded motion, and safety events through ROS2. These signals are converted to the same state, action, and evidence format used by the learner.

G.2 Legged Robot Family

The legged simulation suite uses a quadruped on rough terrain with elevation changes, unstable footholds, confined stepping regions, and tilt inducing slopes. The high level action is

$$a = (v_x, v_y, \omega),$$

and a low level locomotion controller executes the command. Violations include falls, large body tilt, slip, unstable foothold events, recovery triggers, and interventions.

The physical legged trials use a Boston Dynamics Spot. The logging interface records commanded motion, robot state, operator interventions, emergency stop events, recovery events, and available safety monitor outputs. These logs populate the same safe and risky evidence sets used by CoGeL.

G.3 Directional Hazard Setting

The directional hazard setting uses sloped or one way friction regions whose cost depends on traversal direction. Entering a region against the slope or from the low friction side has a higher expected violation cost than leaving it. This produces the directional cost gap studied in Section B.5.

H Compute and Runtime

Table 11 lists the compute and platform information used for the reported simulation sweeps and implementation checks.

Table 11: **Compute and platform summary.**

Item	Value
GPU server	$4\times$ NVIDIA GeForce RTX 4090
GPU memory	24 GB per GPU $(24564\ \mathrm{MiB})$
Driver version	NVIDIA driver 535.309.01
CUDA version	CUDA 12.2
GPU use during sweeps	One GPU per seed or method run, with parallel runs when GPUs are available
Reference code checks	CPU compatible smoke tests, theory checks, and short hidden hazard runs
Simulation tools	Unity and NVIDIA Isaac based scene generation and rollout evaluation
Learning curves	Five seed success and violation curves over simulated environment interaction
Hardware utilities	Spot SDK logging and Jackal ROS2 logging