# SAFE AND GENERALIZABLE HIERARCHICAL MULTI-AGENT RL VIA CONSTRAINT MANIFOLD CONTROL

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

## ABSTRACT

Multi-agent systems are widely used in safety-critical applications that require coordinated behavior under strict safety constraints. Existing approaches face a fundamental trade-off: learning-based methods achieve strong empirical performance but lack theoretical safety guarantees, while control-theoretic methods enforce safety but often lead to overly conservative and inefficient behaviors. We propose a hierarchical multi-agent reinforcement learning framework that enforces hard safety constraints under mild assumptions at a low level via a constraint manifold while enabling effective coordination through high-level policy learning. Our approach provides theoretical safety guarantees in the multi-agent setting and yields stationary learning dynamics, thereby enabling stable and efficient training. Empirically, our method achieves competitive performance while maintaining nearly perfect safety rates, and generalizes effectively to varying numbers of agents and obstacles.

## 1 INTRODUCTION

Multi-agent systems (MAS) have seen significant development in real-world applications such as warehouse robotics (Kattepur et al., 2018), autonomous vehicles (Zhang et al., 2024), traffic routing (Wu et al., 2020), and drone swarm coordination (Batra et al., 2022). A common characteristic of these scenarios is that each agent must not only accomplish its own task but also collaborate efficiently with other agents while maintaining safety such as collision avoidance. To address these challenges, various methods have been explored, ranging from Lagrangian-based Constrained Markov Decision Processes (CMDPs) (Gu et al., 2023; Liu et al., 2021; Ding et al., 2023; Lu et al., 2021; Geng et al., 2023; Zhao et al., 2023) that penalize unsafe behaviors, to safety-filter methods such as Control Barrier Functions (CBFs) (Zhang et al., 2025b;a), traditional control-theoretic approaches like Model Predictive Control (MPC) (Goarin et al., 2025).

Recent learning-based CBF approaches for multi-agent systems (Zhang et al., 2025b;a) have demonstrated strong empirical performance by integrating learning with implicit planning capabilities. This enables more effective and scalable cooperative policies in safety-critical tasks such as multi-agent navigation and collision avoidance. Compared to traditional model-based CBF methods with fixed analytical controllers, these approaches further improve adaptability and performance. However, similar to CMDP-based approaches, these methods rely on learned policies and therefore generally lack formal safety guarantees.

In addition to learning-based methods, control-theoretic approaches (Goarin et al., 2025) are proposed to guarantee safety in MAS under hard constraints. However, in complex MAS with multiple obstacles, they often lead to poor coordination among agents and suboptimal path selection, which can result in deadlocks. To overcome this challenge, safe hierarchical MARL approaches (Ahmad et al., 2025) decompose control into a high-level policy that learns to select or switch between skills, and a low-level controller that executes each skill via a parameterized CBF-quadratic programming (QP) controller. In this framework, CBF parameters are learned, but the safety constraints retain a model-based analytical form.

The need to solve a QP problem at every timestep introduces significant computational overhead, leading to reduced training efficiency and suboptimal performance. In the single-agent setting, constraint manifold approach (Liu et al., 2025) ensures safety via a safe action space, while avoiding



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.

QP solves and thus enabling efficient per-timestep computation. However, it is unclear whether this approach can be extended to multi-agent settings.

Existing methods fail to simultaneously strong task performance, theoretical safety guarantees, and training efficiency. To bridge these gaps, we propose a novel framework **H**ierarchical **M**anifold **M**ulti-Agent PPO (**HMM**), which decouples cooperation and safety across two levels: a learnable policy for coordination at high-level and a model-based controller for enforcing hard constraints at low-level. The high-level policy operates under the Centralized Training with Decentralized Execution (CTDE) fashion to sequentially generate subgoals for each agent, handling multi-agent coordination and path planning. At the low level, the controller guarantees safety by embedding hard constraints into a differentiable constraint manifold and restricting actions to its tangent space. Compared to CBF-based methods that require solving a QP at every timestep, our **HMM** only involves efficient tangent space projections, while still providing structural guarantees of constraint satisfaction at each environmental timestep under mild assumptions. Our main contributions are summarized as follows:

- We propose **HMM**, a hierarchical MARL framework based on a constraint manifold formulation, enabling coordinated multi-agent control while providing formal safety guarantees at every timestep during both training and execution.
- Our approach provides theoretical safety guarantees in the multi-agent setting, and induces a stationary learning process with stable convergence behavior.
- Empirically, **HMM** outperforms other safety MARL baselines on Lidar benchmarks, achieving state-of-the-art performance. It also demonstrates strong generalization across task scales: policies trained with only 3 agents and 3 obstacles generalize effectively to scenarios involving up to 21 agents or obstacles, while maintaining almost perfect safety rates and high task success rates.

## 2 Related Work

Our method lies at the intersection of safe RL and MARL, hierarchical MARL, and constraint manifold based safe control. We review these areas below and highlight that no existing approach provides hard safety guarantees, avoids per step QP solves, and scales to multi agent hierarchical settings. This gap motivates our work.

**Safe RL and MARL.** In safe RL, CMDP serves as a classic framework (Zhao et al., 2023; Altman, 2021; Gu et al., 2023) for maximizing cumulative reward while satisfying safety constraints. Existing methods can be broadly categorized into primal approaches (Xu et al., 2021; Chow et al., 2018; 2019; Liu et al., 2020), which directly enforce constraints; primal-dual approaches (He et al., 2023; Borkar, 2005; Ding et al., 2020; Huang et al., 2023; Tessler et al., 2018) that introduce Lagrange multipliers; and trust-region-based approaches (Achiam et al., 2017; He et al., 2023). Among these, Lagrange-multiplier-based approaches (Gu et al., 2023; Liu et al., 2021; Ding et al., 2023; Lu et al., 2021; Geng et al., 2023; Zhao et al., 2023) have been particularly popular and have been extended to the multi-agent setting. However, these Lagrangian-based approaches, whether in single-agent or multi-agent settings, suffer from problems such as training instability, slow convergence, and sensitivity to hyperparameters (Zanon & Gros, 2020; He et al., 2023; So & Fan, 2023; Ganai et al., 2023). Recently, CBF have been integrated into the RL training process as a safety filter (Tearle et al., 2021; Hsu et al., 2023; Hailemichael et al., 2023). This approach has been explored in both single-agent (Cheng et al., 2019; Emam et al., 2022; Hailemichael et al., 2023) and multi-agent settings (Pereira et al., 2021; 2022; Zhang et al., 2025a). However, existing CBF-based methods either require solving a QP at each timestep, leading to high computational cost, or rely on learning a parameterized CBF, which fail to provide theoretical safety guarantees.

**Hierarchical MARL.** Hierarchical reinforcement learning (HRL) (Pertsch et al., 2021) addresses complex, long-horizon tasks by decomposing decision-making into multiple levels of temporal abstraction. HRL has been studied extensively in both single-agent (Parr & Russell, 1997; Dietterich, 2000; Sutton et al., 1999) and multi-agent settings (Xu et al., 2023; Ghavamzadeh et al., 2006; Son et al., 2019; Dietterich, 2000; Ahilan & Dayan, 2019; Tessler et al., 2017). However, these hierarchical frameworks generally lack explicit safety guarantees. To address this, HMARL-CBF (Ahmad et al., 2025) uses CBF-based low-level controllers for safety. However, it requires solving a QP at each timestep, leading to inefficiency and suboptimal performance compared to DGPPO (Zhang et al., 2025a) on the same benchmarks, as reported in Ahmad et al. (2025).

**Constraint Manifold-Based Safe Control.** Constraint manifold-based methods (Liu et al., 2022; 2025) offer an alternative approach to safety enforcement without requiring QP solvers in RL. This approach has been extended to dynamic high-dimensional robotic tasks (Liu et al., 2023), stochastic constraint manifolds (Gu et al., 2024), and long-term safety under unknown constraints (Günster et al., 2024). However, existing works are focused on single-agent settings. In this work, we extend constraint manifolds to multi-agent settings within a hierarchical framework. This enables efficient and scalable safety enforcement while preserving formal guarantees in multi-agent environments.

## 3 Preliminaries and Problem Formulation

### 3.1 Constrained Semi-Markov Decision Process

We formulate the multi-agent safe control problem as a constrained semi-Markov decision process (CSMDP) (Makar et al., 2001), which allows temporally extended actions. In our setting, we consider the special case where each action is executed for a fixed duration $\tau$. The CSMDP is defined by the tuple $\langle \mathcal{N}, \mathcal{S}, \mathcal{Z}, \mathcal{O}, \mathcal{P}, r, \mathcal{H}, \gamma \rangle$, where $\mathcal{N} = \{1, \dots, N\}$ denotes the set of agents. The global state space is denoted by $\mathcal{S}$. The global state at timestep $t$ is $\mathbf{s}^t = \{\mathbf{s}_i^t\}_{i \in \mathcal{N}}$, where $\mathbf{s}_i^t \in \mathbb{R}^S$ denotes the state of agent $i$. The global state is not directly observable by individual agents; instead, each agent $i$ receives a local observation $o_i^t \in \mathcal{O}_i$. The joint action and observation spaces are $\mathcal{Z} = \mathcal{Z}_1 \times \cdots \times \mathcal{Z}_N$ and $\mathcal{O} = \mathcal{O}_1 \times \cdots \times \mathcal{O}_N$, respectively. Each agent $i \in \mathcal{N}$ has $L$ associated constraint functions defined on local observations, with the joint set defined as $\mathcal{H} := \{h_i^{(\ell)} : \mathcal{O}_i \to \mathbb{R} \mid i \in \mathcal{N},\ \ell = 1, \dots, L\}$. We require hard constraint satisfaction at every timestep $t$:

$$h_i^{(\ell)}(o_i^t) \leq 0, \quad \forall i \in \mathcal{N},\ \ell = 1, \dots, L,\ \forall t \geq 0. \tag{1}$$

The multistep transition kernel $\mathcal{P} : \mathcal{S} \times \mathcal{Z} \to \Delta(\mathcal{S})$, where $\mathcal{P}(\mathbf{s}' \mid \mathbf{s}, \mathbf{z})$ denotes the probability of transitioning to $\mathbf{s}'$ after $\tau$ timesteps under joint action $\mathbf{z}$. The reward function $r : \mathcal{S} \times \mathcal{Z} \to \mathbb{R}$ denotes the cumulative reward collected over the $\tau$ timesteps induced by the joint action.

We distinguish between timesteps and decision epochs: the system evolves at the finer timestep index $t$, while decisions are made at discrete decision epochs indexed by $k$. Each decision epoch $k$ corresponds to the execution of a joint action $\mathbf{z}^k$, which is held fixed for $\tau$ consecutive timesteps, i.e., from $t = k\tau$ to $t = (k+1)\tau - 1$. We use $\mathbf{s}^k := \mathbf{s}^{k\tau}$ to denote the state at the start of epoch $k$. The objective is to find decentralized policies $\pi_i : \mathcal{O}_i \to \mathcal{Z}_i$ defined at decision epochs that maximize the expected discounted return while satisfying the safety constraints:

$$\max_{\pi_1, \dots, \pi_N} \mathbb{E}_{\boldsymbol{\pi}} \left[ \sum_{k=0}^{\infty} \gamma^{k\tau}\, r(\mathbf{s}^k, \mathbf{z}^k) \right] \quad \text{s.t.} \quad h_i^{(\ell)}(o_i^t) \leq 0,\ \forall i \in \mathcal{N},\ \ell = 1, \dots, L,\ \forall t \geq 0. \tag{2}$$

The value function under joint policy $\boldsymbol{\pi}$ in the CSMDP is defined as

$$V^{\boldsymbol{\pi}}(\mathbf{s}) = \mathbb{E}_{\boldsymbol{\pi}} \left[ \sum_{k=0}^{\infty} \gamma^{k\tau}\, r(\mathbf{s}^k, \mathbf{z}^k) \,\middle|\, \mathbf{s}^0 = \mathbf{s} \right]. \tag{3}$$

This yields an effective discount factor of $\gamma^\tau$ at the decision level.

### 3.2 Single Agent Safe Control on the Constraint Manifold

In this work, we consider a control-affine system that is locally Lipschitz continuous and defined as follow:

$$\dot{\mathbf{s}} = f(\mathbf{s}) + G(\mathbf{s})\mathbf{u}_s, \tag{4}$$

where $\mathbf{s} \in \mathcal{S} \subset \mathbb{R}^S$ denotes the system state, $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^U$ denotes the control input, and $f : \mathcal{S} \to \mathbb{R}^S$ and $G : \mathcal{S} \to \mathbb{R}^{S \times U}$ are Lipschitz continuous mappings. We review the key concepts of manifold constraint method (Liu et al., 2022; 2025) for single agent, which forms the basis of our low-level safe controller. Additional details on manifolds are provided in Appendix B.2.

**Constraint manifold.** We consider that the safety of the single agent setting requirements is specified by a set of $L$ inequality constraints $h(\mathbf{s}) \leq \mathbf{0}$ with $h : \mathcal{S} \to \mathbb{R}^L$. Given the inequality constraint definition, we define the safe set as the sublevel set $\mathcal{C} := \{\mathbf{s} \in \mathcal{S} \subset \mathbb{R}^S : h(\mathbf{s}) \leq \mathbf{0}\}$. Slack variables $\boldsymbol{\mu} \in [0, +\infty)^L$ are introduced to rewrite the inequality constraints as equality constraints:

$$c(\mathbf{s}, \boldsymbol{\mu}) := h(\mathbf{s}) + \boldsymbol{\mu} = \mathbf{0}. \tag{5}$$

The Jacobian of $c(\mathbf{s}, \boldsymbol{\mu})$ is given by $J_c(\mathbf{s}, \boldsymbol{\mu}) = [J_h(\mathbf{s}) \;\; \mathbb{I}_L]$, where $\mathbb{I}_L$ denotes the $L$-dimensional identity matrix. The equality constraint formulation defines the constraint manifold $\mathcal{M} := \{(\mathbf{s}, \boldsymbol{\mu}) \in \mathcal{D} : c(\mathbf{s}, \boldsymbol{\mu}) = \mathbf{0}\}$, in the augmented state space $\mathcal{D} := \mathcal{S} \times [0, +\infty)^L \subset \mathbb{R}^{S+L}$. The safe set $\mathcal{C}$ is the projection of $\mathcal{M}$ onto the original state space: for any $(\mathbf{s}, \boldsymbol{\mu}) \in \mathcal{M}$, $h(\mathbf{s}) = -\boldsymbol{\mu} \leq \mathbf{0}$, so $\mathbf{s} \in \mathcal{C}$. More preliminaries of manifold are provided in Appendix B.1

**Augmented dynamics.** To make the slack variables controllable, a controlled system is introduced for each slack variable $\mu_\ell$:

$$\dot{\mu}_\ell = \alpha_\ell(\mu_\ell)\, u_{\mu,\ell}, \quad \ell = 1, \ldots, L, \tag{6}$$

where $\alpha_l$ is a class-$\mathcal{K}$[1] function and $u_{\mu,\ell}$ is the virtual control input for the $\ell$-th slack variable. Combining the original system dynamics equation 4 with the slack variable dynamics equation 6, we obtain the augmented system:

$$\begin{bmatrix} \dot{\mathbf{s}} \\ \dot{\boldsymbol{\mu}} \end{bmatrix} = \begin{bmatrix} f(\mathbf{s}) \\ \mathbf{0} \end{bmatrix} + \begin{bmatrix} G(\mathbf{s}) & \mathbf{0} \\ \mathbf{0} & A(\boldsymbol{\mu}) \end{bmatrix} \begin{bmatrix} \mathbf{u}_s \\ \mathbf{u}_\mu \end{bmatrix}, \tag{7}$$

where $A(\boldsymbol{\mu}) \in \mathbb{R}^{L \times L}$ is a diagonal matrix with entries $A_{\ell\ell} = \alpha_\ell(\mu_\ell)$, and $\mathbf{u}_\mu = [u_{\mu,1}, \ldots, u_{\mu,L}]^\top$ collects all virtual control inputs.

**Safe controller.** Then we can have a controller that keeps the augmented state $(\mathbf{s}, \boldsymbol{\mu})$ on the constraint manifold $\mathcal{M}$. Since any point on $\mathcal{M}$ projects to a safe state in the original state space, staying on $\mathcal{M}$ guarantees that the system remains within the safe set $\mathcal{C}$.

To keep the augmented state on $\mathcal{M}$, its velocity must lie in the tangent space, which is equivalent to enforcing

$$\dot{c}(\mathbf{s}, \boldsymbol{\mu}) = J_c(\mathbf{s}, \boldsymbol{\mu}) \begin{bmatrix} \dot{\mathbf{s}} \\ \dot{\boldsymbol{\mu}} \end{bmatrix} = \mathbf{0}, \tag{8}$$

where $J_c(\mathbf{s}, \boldsymbol{\mu}) = [J_h(\mathbf{s}) \;\; \mathbb{I}_L]$ is the Jacobian of $c$. Substituting the augmented dynamics equation 7 into equation 8 yields the linear equation

$$\boldsymbol{\psi}(\mathbf{s}) + J_u(\mathbf{s}, \boldsymbol{\mu}) \begin{bmatrix} \mathbf{u}_s \\ \mathbf{u}_\mu \end{bmatrix} = \mathbf{0}, \tag{9}$$

where $J_u(\mathbf{s}, \boldsymbol{\mu}) = [J_h(\mathbf{s})G(\mathbf{s}) \;\; A(\boldsymbol{\mu})]$ is the control Jacobian and $\boldsymbol{\psi}(\mathbf{s}) = J_h(\mathbf{s})f(\mathbf{s})$ is the *constraint drift* induced by the system drift $f(\mathbf{s})$.

Solving equation 9 gives the general form of the constraint manifold safe controller:

$$\begin{bmatrix} \mathbf{u}_s \\ \mathbf{u}_\mu \end{bmatrix} = \underbrace{-J_u^\dagger \boldsymbol{\psi}}_{\substack{\text{drift} \\ \text{compensation}}} \; \underbrace{-\lambda J_u^\dagger c}_{\substack{\text{manifold} \\ \text{contraction}}} + \underbrace{B_u \mathbf{u}}_{\substack{\text{tangential} \\ \text{action}}}, \tag{10}$$

where $J_u^\dagger$ is the pseudoinverse of $J_u$, $B_u$ is the tangent space basis satisfying $J_u B_u = \mathbf{0}$, $\lambda > 0$ is a contraction gain, and $\mathbf{u}$ is an arbitrary task-level control signal. The three terms play distinct roles: the *drift compensation* term cancels the natural drift of the system, the *manifold contraction* term retracts the augmented state back to $\mathcal{M}$ whenever numerical errors cause deviation, and the *tangential action* term allows free exploration along the tangent space without violating constraints. Any task-level control signal $\mathbf{u}$ mapped through equation 10 produces a safe control input $\mathbf{u}_s$, ensuring that the system state remains within the safe set at every timestep.

[1] A continuous function $\alpha : [0, \infty) \to [0, \infty)$ is of class $\mathcal{K}$ if it is strictly increasing and $\alpha(0) = 0$.

# 4 METHODOLOGY

## 4.1 OVERVIEW

We propose a hierarchical framework for safe MARL, illustrated in Figure 1. The framework consists of a high-level cooperative planner with a low-level safe controller built on the constraint manifold. The high-level policy generates subgoals for each agent every $\tau$ timesteps to enable coordination. At the low level, each agent independently applies a constraint manifold safe controller that projects low level actions onto the tangent space of its constraint manifold, structurally guaranteeing safety at every timestep. As a result, our framework enables efficient multi-agent coordination while guaranteeing safety by construction at every timestep.

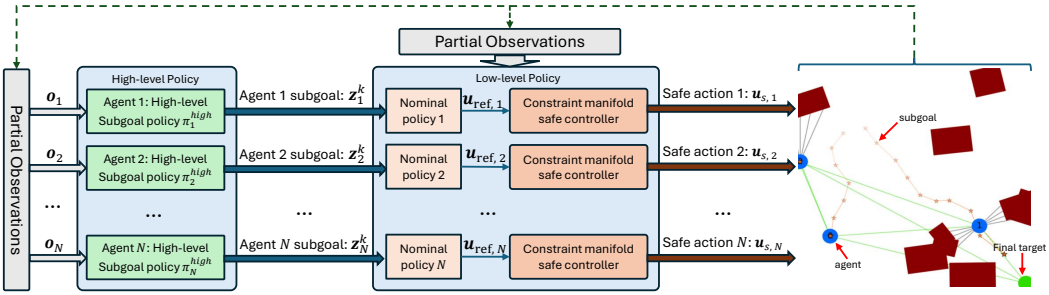



Figure 1: Overview of the hierarchical safe MARL framework during decentralized execution. Each agent's high-level policy generates a subgoal $\mathbf{z}_i^k$, which is converted into a nominal action $\mathbf{u}_{\text{ref},i}$ and then projected onto the constraint manifold to produce a safe action $\mathbf{u}_{s,i}$.

## 4.2 HIGH-LEVEL SUBGOAL POLICY LEARNING

The high-level planner is responsible for generating temporally extended subgoals that coordinate cooperative behavior across agents. At each decision epoch $k$, each agent $i$ uses its high-level policy $\pi_i^{high}$, conditioned on its observation $o_i^k$, to generate a subgoal $\mathbf{z}_i^k \in \mathbb{R}^d$. The subgoal represents a target position relative to agent $i$'s current state, where $d$ is the spatial dimension of the environment. This subgoal serves as a short-horizon waypoint that the low-level safe controller drives the agent toward over the subsequent $\tau$ environment timesteps.

We adopt a MARL framework under the centralized training with decentralized execution (CTDE) paradigm for high-level coordination. The high-level policy is based on MAPPO (Yu et al., 2022) and incorporates graph-based information aggregation inspired by InforMARL (Nayak et al., 2023). Each agent observes its own state, as well as the relative states of nearby agents and obstacles within its sensing radius, forming a local interaction graph. We parameterize the high-level policy $\pi_i^{high}$ using a graph neural network (GNN) (Veličković et al., 2017; Nayak et al., 2023; Zhang et al., 2025b;a) with attention-based message passing to aggregate information from neighboring agents and obstacles. This design ensures permutation invariance and allows the policy to generalize across varying agents and obstacles sizes. Following the CSMDP formulation with fixed epoch duration $\tau$, the high-level objective is

$$J(\boldsymbol{\pi}^{high}) = \mathbb{E}_{\boldsymbol{\pi}^{high}}\left[\sum_{k=0}^{\infty} \gamma^{k\tau}\, r(\mathbf{s}^k, \mathbf{z}^k)\right]. \tag{11}$$

The semi-MDP Bellman residual with effective per-epoch discount $\gamma^\tau$ is $\delta^k = r(\mathbf{s}^k, \mathbf{z}^k) + \gamma^\tau V_\phi(\mathbf{s}^{k+1}) - V_\phi(\mathbf{s}^k)$, and advantages are estimated via Generalized Advantage Estimation: $\hat{A}^k = \sum_{j=0}^{\infty} (\gamma^\tau \lambda_{\text{GAE}})^j\, \delta^{k+j}$, where $V_\phi$ is the centralized value function and $\lambda_{\text{GAE}} \in (0, 1)$ is the GAE trace-decay parameter.

## 4.3 LOW-LEVEL SAFE CONTROLLER

The low-level controller executes the subgoals generated by the high-level policy while strictly satisfying safety constraints at every timestep. Building on the constraint manifold method reviewed in Section 3.2, the safe controller is applied independently to each agent, exploiting the natural

decoupling of control inputs across agents. Given a subgoal $\mathbf{z}_i^k$ from the high-level policy, the low-level controller first computes a nominal reference action $\mathbf{u}_{\text{ref},i} = \pi_{\text{ref}}(\mathbf{s}_i, \mathbf{z}_i^k)$ that drives agent $i$ toward the subgoal. This nominal action does not account for safety constraints; the constraint manifold controller then projects $\mathbf{u}_{\text{ref},i}$ onto the tangent space of the local constraint manifold to produce a safe control input $\mathbf{u}_{s,i}$ for each agent $i$.

**Per-agent constraint formulation.** For each agent $i$, pairwise collision-avoidance constraints are defined with the $L$ nearest entities (agents and obstacles). Each such neighbor $j$ instantiates a per-agent constraint $h_i^{(\ell)}$ from the CSMDP formulation (with $\ell \equiv j$):

$$h_{ij} = (r_{ij}^{\text{safe}} + \delta_{ij})^2 - \|\mathbf{p}_i - \mathbf{p}_j\|^2 \leq 0, \tag{12}$$

where $\mathbf{p}_i, \mathbf{p}_j \in \mathbb{R}^d$ are positions and $r_{ij}^{\text{safe}}$ is the safety radius. The velocity-dependent margin $\delta_{ij} = d_0 + v_{\text{approach}}^2/(2a_{ij})$ accounts for braking distance, where $v_{\text{approach}}$ is the closing speed along the collision axis and $a_{ij}$ is the effective deceleration capacity. For agent-agent pairs, symmetry doubles the collision radius and braking capacity ($r_{ij} = 2r$, $a_{ij} = 2a_{\max}$); agent-obstacle pairs use single-agent values.

**Safe control via tangent space projection.** Each agent $i$ independently projects its nominal action $\mathbf{u}_{\text{ref},i}$ onto the tangent space of its local constraint manifold $\mathcal{M}_i$, following the constraint manifold safe controller in equation 10:

$$\mathbf{u}_{s,i} = \underbrace{-J_{u,i}^\dagger \boldsymbol{\psi}_i}_{\substack{\text{drift} \\ \text{compensation}}} \underbrace{-\lambda J_{u,i}^\dagger c_i}_{\substack{\text{manifold} \\ \text{contraction}}} + \underbrace{B_{u,i}\, \mathbf{u}_{\text{ref},i}}_{\substack{\text{tangential} \\ \text{tracking}}}, \tag{13}$$

where $J_{u,i}$, $\boldsymbol{\psi}_i$, $c_i$, and $B_{u,i}$ are computed locally from agent $i$'s state and the states of its neighbors. The resulting safe control $\mathbf{u}_{s,i}$ is guaranteed to keep agent $i$'s augmented state on $\mathcal{M}_i$, ensuring constraint satisfaction at every timestep.

## 5 Theoretical Guarantees

### 5.1 Safety Guarantee

**Theorem 1** (Compositional Safety)**.** *Consider $N$ agents with decoupled control-affine dynamics $\dot{\mathbf{s}}_i = f_i(\mathbf{s}_i) + G_i(\mathbf{s}_i)\mathbf{u}_i$, each applying the constraint manifold controller equation 10 on its local constraint set $\mathbf{h}_i(\mathbf{s}_i, \mathbf{s}_{\mathcal{N}_i}) \leq \mathbf{0}$, where $\mathcal{N}_i = \mathcal{E}_i \cup \mathcal{A}_i$ denotes neighboring obstacles and agents. Under the standard assumptions (Assumptions 1–4 in Appendix B.4), if all agents start on their local constraint manifolds, the joint trajectory remains in the global safe set for all time.*

*Proof sketch (full proof in Appendix B.5).* Each agent $i$ defines a constraint residual $\mathbf{c}_i = \mathbf{h}_i(\mathbf{s}_i, \mathbf{s}_{\mathcal{N}_i}) + \boldsymbol{\mu}_i$ and a Lyapunov function $V_i = \frac{1}{2}\|\mathbf{c}_i\|^2$. Expanding $\dot{\mathbf{c}}_i$ via equation 8 and equation 9 gives

$$\dot{\mathbf{c}}_i = \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_i}\dot{\mathbf{s}}_i + \underbrace{\sum_{e \in \mathcal{E}_i} \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_e}\dot{\mathbf{s}}_e}_{\text{obstacle motion}} + \underbrace{\sum_{j \in \mathcal{A}_i} \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_j}\dot{\mathbf{s}}_j}_{\text{other agents}} + \dot{\boldsymbol{\mu}}_i = \boldsymbol{\psi}_i + J_{u,i}\begin{bmatrix}\mathbf{u}_{s,i} \\ \mathbf{u}_{\mu,i}\end{bmatrix}, \tag{14}$$

where the drift $\boldsymbol{\psi}_i$ absorbs all terms not controlled by agent $i$: its autonomous dynamics, obstacles, and the states of other agents. In the single-agent setting, only the obstacle term remains and the multi-agent extension adds the inter-agent term, but crucially both enter only the drift $\boldsymbol{\psi}_i$.

Substituting equation 14 into $\dot{V}_i = \mathbf{c}_i^\top \dot{\mathbf{c}}_i$ and expanding (see equation 35), the drift term $\mathbf{c}_i^\top(\boldsymbol{\psi}_i - J_{u,i}J_{u,i}^\dagger \boldsymbol{\psi}_i)$ vanishes since Assumption 4 implies $J_{u,i}J_{u,i}^\dagger = I$. The null-space term $\mathbf{c}_i^\top J_{u,i} B_{u,i}\mathbf{u}_{\text{ref},i} = 0$ by construction. The remaining contraction term yields

$$\dot{V}_i = -\lambda\, \mathbf{c}_i^\top J_{u,i} J_{u,i}^\dagger\, \mathbf{c}_i \leq 0. \tag{15}$$

Since $V_i(0) = 0$ and $\dot{V}_i \leq 0$, we have $V_i(t) = 0$ for all $t$, so every agent remains on its local constraint manifold and satisfies $\mathbf{h}_i(t) \leq \mathbf{0}$ (Remark 1). Because every pairwise inter-agent constraint $h_{ij}$ is enforced by at least one agent's local guarantee, intersecting all local safe sets recovers global safety. □

### 5.2 Reduction to Stationary MARL

A well-known challenge in hierarchical RL is that the high-level transition kernel $P_{high}$ depends on the evolving low-level policy, causing non-stationarity (Sutton et al., 1999; Nachum et al., 2018). Consequently, the induced high-level process violates the stationarity assumption of Markov decision processes, resulting in non-stationary transition dynamics, biased value estimation, and lower training stability. In our framework, this problem does not arise due to the following property:

**Proposition 1** (Stationarity by Construction)**.** *Given that the low-level policy $\pi_{low}$ contains no learnable parameters, the induced high-level transition kernel $P_{high}(s' \mid s, z)$ and reward function $r(s, z)$ are time-invariant and uniquely determined by the environment dynamics.*

*Proof.* The high-level dynamics $P_{high}$ depend only on the environment dynamics $P$, the nominal policy $\pi_{\text{nom}}$, and the constraint manifold projection $\Pi_{\mathcal{M}}$. As all these components are stationary and independent of the high-level policy's learning process, the transition kernel remains constant throughout training. □

Therefore, our framework effectively reduces the hierarchical safe MARL problem to solving an unconstrained and stationary high-level MDP. This reduction provides significant advantages for training stability. By offloading safety to a fixed lower level, we eliminate the non-stationarity and distributional shift typical of hierarchical RL. Consequently, the high-level policy interacts with a consistent environment, allowing standard MARL algorithms like MAPPO (Yu et al., 2022). These algorithms inherit their original convergence guarantees, ensuring stable optimization and stationary throughout the entire learning process.

## 6 Experiments

In this section, we provide a comprehensive evaluation and analysis of our **HMM** across four key dimensions: overall performance, safety throughout the training process, computational efficiency, and generalization capability. By comparing our approach with several baselines, we demonstrate its superior effectiveness in achieving better performance, strictly maintaining safety constraints even during the early stages of learning, and exhibiting stronger generalization capabilities compared to learning-based baselines.

### 6.1 Settings

**Environments.** We evaluate our algorithm across a diverse set of tasks. Specifically, we conduct experiments in LiDAR-based environments (Keyumarsi et al., 2023; Zhang et al., 2025a;b), including both 2D environments (LidarSpread, LidarTarget, LidarLine, LidarBicycle) and 3D environments (LinearDrone, CrazyFlie) (See Figures 5). To comprehensively assess performance against baselines, we adopt two metrics: safety and success rate. The safety rate measures the ratio of agents that remain safe throughout the entire trajectory. The success rate is defined as the ratio of agents that both remain safe and reach their goals.

**Baselines.** We compare against DGPPO (Zhang et al., 2025a), InforMARL (Nayak et al., 2023), and MAPPO-Lagrangian (Gu et al., 2021; 2023). In InforMARL, constraint violations are incorporated via a penalty term $\beta \cdot \max\{0, \max_m h^{(m)}\}$ with varying penalty weights (denoted `Penalty(`$\beta$`)` in figures for $\beta \in \{0.02, 0.1, 0.5\}$), and a scheduled variant (`Schedule`) where $\beta$ is increased during training is also evaluated. For MAPPO-Lagrangian, a GNN backbone is used and different Lagrange multiplier learning rates are considered (`Lagr(1)`, `Lagr(5)` for multiplier rates 1 and 5, and `Lagr(lr)` for increasing the learning rate to 0.1). Our method is denoted **HMM** in all figures.

### 6.2 Main Results

**Overall Performance and Stability.** We evaluate the performance of different algorithms and visualize their results in terms of safety rate and success rate in a two-dimensional plot. For each method, we report both the mean and standard deviation (std) in the figure. As shown in Figure 2, our algorithm consistently outperforms all other baselines in both safety rate and success rate. Notably, our method achieves a highly stable safety rate across all environments, remaining close to 100%

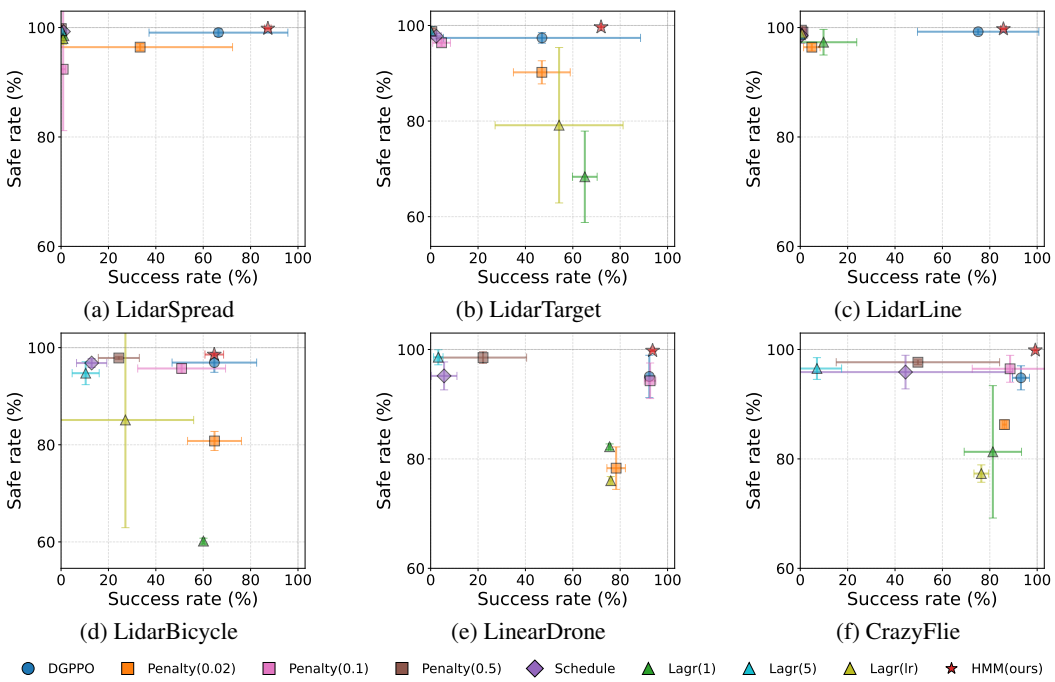



Figure 2: Safe rate and success rate (mean $\pm$ std) across six environments.

throughout. Moreover, the low std reported in Figure 2 indicates that our method exhibits consistently low variance across runs, demonstrating strong training stability. This suggests that our algorithm is not only effective but also robust during training.

**Persistent Safety throughout Training.** We present training curves of the safety rate during the learning process. As shown in Figure 3, other learning-based baselines are not able to guarantee safety at the early stage of training when the neural networks have not yet converged. In contrast, our **HMM** consistently maintains safety throughout the entire training process. The rare residual violations (<2%) stem from discrete-time integration, neighbor truncation, and partial observability, which fall outside our continuous-time guarantee (Theorem 1); see Appendix C for details.

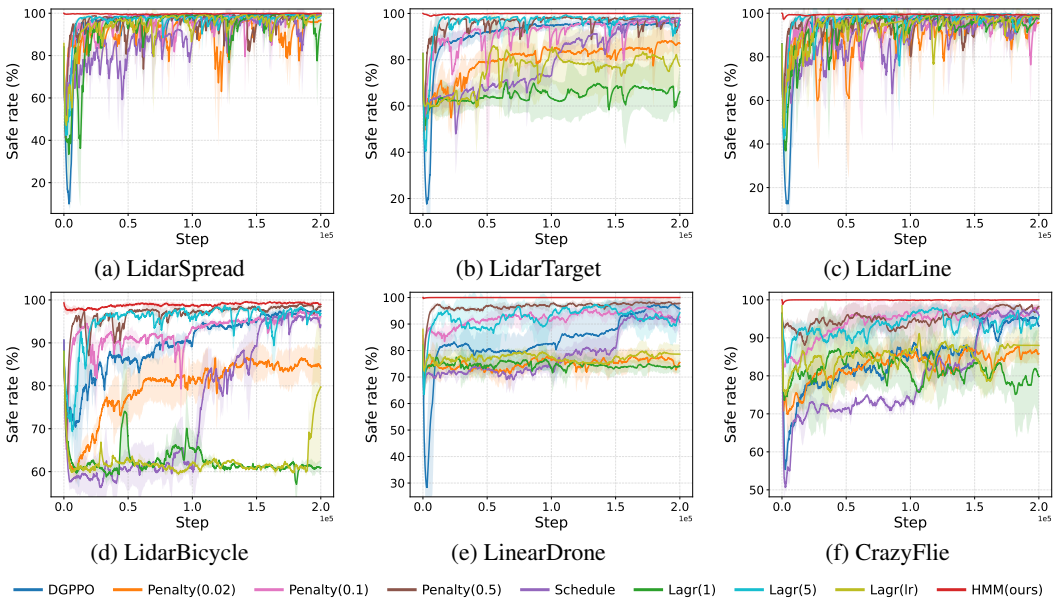



Figure 3: Training safe rate curves across six environments.

**Computational Efficiency.** We compare the training efficiency of our constraint manifold controller against the QP-based CBF baseline on a single NVIDIA H200 GPU. Benefiting from the closed-form pseudoinverse projection, our method achieves a significant speedup in training throughput. Specifically, based on the same training setting, while the CBF-QP baseline requires over 5 days to complete 200k steps, our **HMM** finishes the same training process within 8 to 10 hours. A detailed breakdown of the per-iteration cost and total training time is provided in Appendix E.1.

**Ablation Study.** We further compare three variants: the nominal policy, the nominal policy+constraint manifold, and our **HMM**. Table 2 shows that **HMM** achieves higher safety and success rates than both baselines, indicating that the high-level planning component effectively alleviates deadlock issues and improves coordination efficiency, while the low-level controller safe constraints.

### 6.3 Generalization

All our 2D environments are trained with 3 agents and 3 obstacles, while the 3D environments are trained with 3 agents and 6 obstacles. During evaluation, we increase the number of agents and obstacles while keeping the map size fixed. As shown in Figure 4, when the number of agents increases, all other baselines fail to maintain safety and often break down. This suggests that the other baselines struggle to generalize under distribution shifts, particularly as interaction complexity increases with larger numbers of agents and obstacles. In contrast, our **HMM** continues to guarantee safety even under increased numbers of agents or obstacles, and in most cases achieves an even higher success rate. Results for the remaining environments are provided in the appendix E.2.

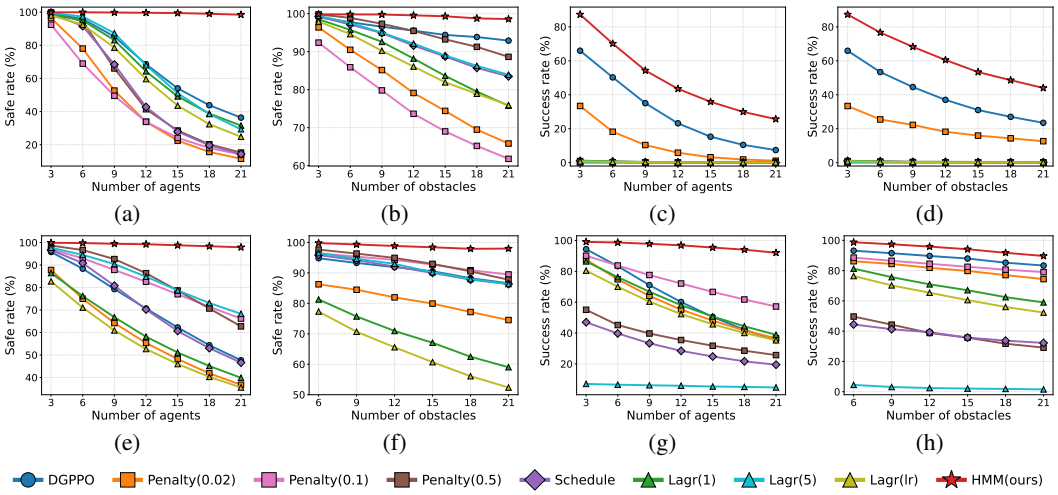



Figure 4: Generalization results. LidarSpread (a–d) and CrazyFlie (e–h). Left two columns: safe rate; right two columns: success rate. Models trained with 3 agents and 3/6 obstacles are evaluated with varying numbers of agents and obstacles.

## 7 Conclusion

We propose a hierarchical safe multi-agent reinforcement learning framework. The high-level policy adopts a CTDE MARL architecture to generate subgoals, while a fixed low-level constraint-manifold controller projects these subgoals onto the safe action space. We provide theoretical guarantees on both safety and convergence of the proposed approach. Empirically, our method achieves near-perfect safety rates across multiple multi-agent navigation tasks, while maintaining higher task success rates compared to other baselines. Furthermore, it demonstrates strong generalization across varying numbers of agents and obstacles. Overall, our results highlight the effectiveness of integrating learning-based coordination with control-theoretic safety mechanisms, offering a promising direction for developing scalable and reliable multi-agent systems.

**Limitation** The low-level policy of our **HMM** is based on control-theoretic safety mechanisms, which makes it less suitable for tasks involving complex physical interactions, such as contact-rich manipulation or object transportation. In particular, extending our framework to scenarios where agents must interact with external objects while ensuring their safety remains challenging.

## AI USE STATEMENT

We used AI tools to polish and edit the manuscript and to assist with debugging the code. AI tools were not directly involved in the development of the methodology. All AI-assisted content and code changes were reviewed and verified by the authors.

## REPRODUCIBILITY STATEMENT

Our experiments are designed to be reproducible. The main text clearly describes our methodology, environments, evaluation protocol, and baselines. Additional implementation details, model configurations, and hyperparameters required for reproduction are provided in the Appendix F.

## REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. Pmlr, 2017.

Sanjeevan Ahilan and Peter Dayan. Feudal multi-agent hierarchies for cooperative reinforcement learning. *arXiv preprint arXiv:1901.08492*, 2019.

HM Sabbir Ahmad, Ehsan Sabouni, Alexander Wasilkoff, Param Budhraja, Zijian Guo, Songyuan Zhang, Chuchu Fan, Christos Cassandras, and Wenchao Li. Hmarl-cbf–hierarchical multi-agent reinforcement learning with control barrier functions for safety-critical autonomous systems. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Sumeet Batra, Zhehui Huang, Aleksei Petrenko, Tushar Kumar, Artem Molchanov, and Gaurav S Sukhatme. Decentralized control of quadrotor swarms with end-to-end deep reinforcement learning. In *Conference on robot learning*, pp. 576–586. PMLR, 2022.

Vivek S Borkar. An actor-critic algorithm for constrained markov decision processes. *Systems & control letters*, 54(3):207–213, 2005.

Nicolas Boumal. *An introduction to optimization on smooth manifolds*. Cambridge University Press, 2023.

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.

Yinlam Chow, Ofir Nachum, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. A lyapunov-based approach to safe reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Yinlam Chow, Ofir Nachum, Aleksandra Faust, Edgar Duenez-Guzman, and Mohammad Ghavamzadeh. Lyapunov-based safe policy optimization for continuous control. *arXiv preprint arXiv:1901.10031*, 2019.

Thomas G Dietterich. Hierarchical reinforcement learning with the maxq value function decomposition. *Journal of artificial intelligence research*, 13:227–303, 2000.

Dongsheng Ding, Kaiqing Zhang, Tamer Basar, and Mihailo Jovanovic. Natural policy gradient primal-dual method for constrained markov decision processes. *Advances in Neural Information Processing Systems*, 33:8378–8390, 2020.

Dongsheng Ding, Xiaohan Wei, Zhuoran Yang, Zhaoran Wang, and Mihailo Jovanovic. Provably efficient generalized lagrangian policy optimization for safe multi-agent reinforcement learning. In *Learning for dynamics and control conference*, pp. 315–332. PMLR, 2023.

Yousef Emam, Gennaro Notomista, Paul Glotfelter, Zsolt Kira, and Magnus Egerstedt. Safe reinforcement learning using robust control barrier functions. *IEEE Robotics and Automation Letters*, 10(3):2886–2893, 2022.

Milan Ganai, Zheng Gong, Chenning Yu, Sylvia Herbert, and Sicun Gao. Iterative reachability estimation for safe reinforcement learning. *Advances in Neural Information Processing Systems*, 36:69764–69797, 2023.

Nan Geng, Qinbo Bai, Chenyi Liu, Tian Lan, Vaneet Aggarwal, Yuan Yang, and Mingwei Xu. A reinforcement learning framework for vehicular network routing under peak and average constraints. *IEEE Transactions on Vehicular Technology*, 72(5):6753–6764, 2023.

Mohammad Ghavamzadeh, Sridhar Mahadevan, and Rajbala Makar. Hierarchical multi-agent reinforcement learning. *Autonomous Agents and Multi-Agent Systems*, 13(2):197–229, 2006.

Manohari Goarin, Guanrui Li, Alessandro Saviolo, and Giuseppe Loianno. Decentralized nonlinear model predictive control for safe collision avoidance in quadrotor teams with limited detection range. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5387–5393, 2025. doi: 10.1109/ICRA55743.2025.11128389.

Shangding Gu, Jakub Grudzien Kuba, Munning Wen, Ruiqing Chen, Ziyan Wang, Zheng Tian, Jun Wang, Alois Knoll, and Yaodong Yang. Multi-agent constrained policy optimisation. *arXiv preprint arXiv:2110.02793*, 2021.

Shangding Gu, Jakub Grudzien Kuba, Yuanpei Chen, Yali Du, Long Yang, Alois Knoll, and Yaodong Yang. Safe multi-agent reinforcement learning for multi-robot control. *Artificial Intelligence*, 319: 103905, 2023.

Shangding Gu, Puze Liu, Alap Kshirsagar, Guang Chen, Jan Peters, and Alois Knoll. Roscom: Robust safe reinforcement learning on stochastic constraint manifolds. *IEEE Transactions on Automation Science and Engineering*, 22:5841–5851, 2024.

Jonas Günster, Puze Liu, Jan Peters, and Davide Tateo. Handling long-term safety and uncertainty in safe reinforcement learning. *arXiv preprint arXiv:2409.12045*, 2024.

Habtamu Hailemichael, Beshah Ayalew, and Andrej Ivanco. Optimal control barrier functions for rl based safe powertrain control. *IFAC-PapersOnLine*, 56(3):385–390, 2023.

Tairan He, Weiye Zhao, and Changliu Liu. Autocost: Evolving intrinsic cost for zero-violation reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 14847–14855, 2023.

Kai-Chieh Hsu, Haimin Hu, and Jaime F Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 7, 2023.

Weidong Huang, Jiaming Ji, Chunhe Xia, Borong Zhang, and Yaodong Yang. Safedreamer: Safe reinforcement learning with world models. *arXiv preprint arXiv:2307.07176*, 2023.

Ajay Kattepur, Hemant Kumar Rath, Anantha Simha, and Arijit Mukherjee. Distributed optimization in multi-agent robotics for industry 4.0 warehouses. In *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, pp. 808–815, 2018.

Shaghayegh Keyumarsi, Made Widhi Surya Atman, and Azwirman Gusrialdi. Lidar-based online control barrier function synthesis for safe navigation in unknown environments. *IEEE Robotics and Automation Letters*, 9(2):1043–1050, 2023.

John M Lee. Smooth manifolds. In *Introduction to smooth manifolds*, pp. 1–29. Springer, 2003.

Chenyi Liu, Nan Geng, Vaneet Aggarwal, Tian Lan, Yuan Yang, and Mingwei Xu. Cmix: Deep multi-agent reinforcement learning with peak and average constraints. In *Joint european conference on machine learning and knowledge discovery in databases*, pp. 157–173. Springer, 2021.

Puze Liu, Davide Tateo, Haitham Bou Ammar, and Jan Peters. Robot reinforcement learning on the constraint manifold. In *Conference on Robot Learning*, pp. 1357–1366. PMLR, 2022.

Puze Liu, Kuo Zhang, Davide Tateo, Snehal Jauhri, Zhiyuan Hu, Jan Peters, and Georgia Chalvatzaki. Safe reinforcement learning of dynamic high-dimensional robotic tasks: navigation, manipulation, interaction. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 9449–9456. IEEE, 2023.

Puze Liu, Haitham Bou-Ammar, Jan Peters, and Davide Tateo. Safe reinforcement learning on the constraint manifold: Theory and applications. *IEEE Transactions on Robotics*, 41:3442–3461, 2025. doi: 10.1109/TRO.2025.3567477.

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.

Songtao Lu, Kaiqing Zhang, Tianyi Chen, Tamer Başar, and Lior Horesh. Decentralized policy gradient descent ascent for safe multi-agent reinforcement learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 8767–8775, 2021.

Rajbala Makar, Sridhar Mahadevan, and Mohammad Ghavamzadeh. Hierarchical multi-agent reinforcement learning. In *Proceedings of the fifth international conference on Autonomous agents*, pp. 246–253, 2001.

Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. *Advances in neural information processing systems*, 31, 2018.

Siddharth Nayak, Kenneth Choi, Wenqi Ding, Sydney Dolan, Karthik Gopalakrishnan, and Hamsa Balakrishnan. Scalable multi-agent reinforcement learning through intelligent information aggregation. In *International conference on machine learning*, pp. 25817–25833. PMLR, 2023.

Ronald Parr and Stuart Russell. Reinforcement learning with hierarchies of machines. *Advances in neural information processing systems*, 10, 1997.

Marcus Pereira, Ziyi Wang, Ioannis Exarchos, and Evangelos Theodorou. Safe optimal control using stochastic barrier functions and deep forward-backward sdes. In *Conference on robot learning*, pp. 1783–1801. PMLR, 2021.

Marcus A Pereira, Augustinos D Saravanos, Oswin So, and Evangelos A Theodorou. Decentralized safe multi-agent stochastic optimal control using deep fbsdes and admm. *arXiv preprint arXiv:2202.10658*, 2022.

Karl Pertsch, Youngwoon Lee, and Joseph Lim. Accelerating reinforcement learning with learned skill priors. In *Conference on robot learning*, pp. 188–204. PMLR, 2021.

Oswin So and Chuchu Fan. Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning. *arXiv preprint arXiv:2305.14154*, 2023.

Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning. In *International conference on machine learning*, pp. 5887–5896. PMLR, 2019.

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.

Ben Tearle, Kim P Wabersich, Andrea Carron, and Melanie N Zeilinger. A predictive safety filter for learning-based racing control. *IEEE Robotics and Automation Letters*, 6(4):7635–7642, 2021.

Chen Tessler, Shahar Givony, Tom Zahavy, Daniel Mankowitz, and Shie Mannor. A deep hierarchical approach to lifelong learning in minecraft. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31, 2017.

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

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Tong Wu, Pan Zhou, Kai Liu, Yali Yuan, Xiumin Wang, Huawei Huang, and Dapeng Oliver Wu. Multi-agent deep reinforcement learning for urban traffic light control in vehicular networks. *IEEE Transactions on Vehicular Technology*, 69(8):8243–8256, 2020.

Tengyu Xu, Yingbin Liang, and Guanghui Lan. Crpo: A new approach for safe reinforcement learning with convergence guarantee. In *International Conference on Machine Learning*, pp. 11480–11491. PMLR, 2021.

Zhiwei Xu, Yunpeng Bai, Bin Zhang, Dapeng Li, and Guoliang Fan. Haven: Hierarchical cooperative multi-agent reinforcement learning with dual coordination mechanism. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11735–11743, 2023.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems*, 35:24611–24624, 2022.

Mario Zanon and Sébastien Gros. Safe reinforcement learning using robust mpc. *IEEE Transactions on Automatic Control*, 66(8):3638–3652, 2020.

Ruiqi Zhang, Jing Hou, Florian Walter, Shangding Gu, Jiayi Guan, Florian Röhrbein, Yali Du, Panpan Cai, Guang Chen, and Alois Knoll. Multi-agent reinforcement learning for autonomous driving: A survey. *arXiv preprint arXiv:2408.09675*, 2024.

Songyuan Zhang, Oswin So, Mitchell Black, and Chuchu Fan. Discrete gcbf proximal policy optimization for multi-agent safe optimal control. *arXiv preprint arXiv:2502.03640*, 2025a.

Songyuan Zhang, Oswin So, Kunal Garg, and Chuchu Fan. Gcbf+: A neural graph control barrier function framework for distributed safe multiagent control. *IEEE Transactions on Robotics*, 41: 1533–1552, 2025b.

Youpeng Zhao, Yaodong Yang, Zhenbo Lu, Wengang Zhou, and Houqiang Li. Multi-agent first order constrained optimization in policy space. *Advances in Neural Information Processing Systems*, 36: 39189–39211, 2023.

# A Appendix

# B Safety Proofs

## B.1 Manifold Preliminaries

We review relevant concepts from manifold theory here; see (Lee, 2003; Boumal, 2023) for details.

**Manifold** An $n$-dimensional manifold is a topological space that is locally homeomorphic to $\mathbb{R}^n$. Therefore, every point has a neighborhood that is homeomorphic to an open subset of $\mathbb{R}^n$. A differentiable manifold(also called a smooth manifold) is a manifold equipped with an atlas of coordinate charts whose transition maps are all differentiable In this work, we consider differentiable manifolds.

**Tangent Space** Let $\mathcal{M}$ be an $n$-dimensional differentiable manifold and $p \in \mathcal{M}$. The tangent space at $p$, denoted by $\mathrm{T}_p\mathcal{M}$, is an $n$-dimensional vector space of all tangent vectors at $p$, providing a local linear approximation of $\mathcal{M}$ near $p$. The *tangent bundle* $\mathrm{T}\mathcal{M} = \sqcup_{p\in\mathcal{M}} \mathrm{T}_p\mathcal{M}$ is the disjoint union of all tangent spaces over $\mathcal{M}$.

**Smooth Map and Rank** Let $\mathcal{M}$ and $\mathcal{N}$ be smooth manifolds. A smooth map $\Phi : \mathcal{M} \to \mathcal{N}$ induces, at each point $p \in \mathcal{M}$, a linear map $\mathrm{D}\Phi_p : \mathrm{T}_p\mathcal{M} \to \mathrm{T}_{\Phi(p)}\mathcal{N}$, called the differential of $\Phi$ at $p$. The *rank* of $\Phi$ at $p$ is defined as the rank of $\mathrm{D}\Phi_p$.

**Embedded Submanifold** An *embedded submanifold* is a subset $\mathcal{S} \subseteq \mathcal{M}$ that itself forms a manifold, with the *codimension* given by $\dim \mathcal{M} - \dim \mathcal{S}$.

**Theorem 2** (Constant-Rank Level Set Theorem)**.** *Let $\mathcal{M}$ and $\mathcal{N}$ be smooth manifolds, and let $\Phi : \mathcal{M} \to \mathcal{N}$ be a smooth map with constant rank $r$. Then every level set of $\Phi$ is a properly embedded submanifold of $\mathcal{M}$ with codimension $r$.*

As a consequence, consider a smooth function $f : \mathcal{E} \to \mathbb{R}^k$, where $\mathcal{E}$ is a Euclidean space of dimension $d > k$ equipped with an inner product $\langle \cdot, \cdot \rangle$ and the induced norm $\| \cdot \|$. If the differential $Df(\boldsymbol{x})$ has full rank $k$ at every $\boldsymbol{x}$ in the level set

$$\mathcal{M} = \{\boldsymbol{x} \in \mathcal{E} : f(\boldsymbol{x}) = \mathbf{0}\}, \tag{16}$$

then $\mathcal{M}$ is a smooth *embedded submanifold* of $\mathcal{E}$ with dimension $d - k$. Its tangent space at any point $\boldsymbol{x} \in \mathcal{M}$ is characterized by

$$\mathrm{T}_{\boldsymbol{x}}\mathcal{M} = \{\boldsymbol{v} \in \mathcal{E} : \langle \operatorname{grad} f_i(\boldsymbol{x}),\, \boldsymbol{v} \rangle = 0,\ i \in \{1, \ldots, k\}\}, \tag{17}$$

which has dimension $\dim \mathrm{T}_{\boldsymbol{x}}\mathcal{M} = d - k$. Intuitively, the tangent space consists of all directions orthogonal to every constraint gradient.

**LaSalle's Invariance Principle.** Consider the autonomous system $\dot{\mathbf{x}} = f(\mathbf{x})$ with $f : \mathcal{D} \to \mathbb{R}^n$ locally Lipschitz on a domain $\mathcal{D} \subseteq \mathbb{R}^n$.

### B.2 Single Agent Constraint Manifold Construction

**Slack variable augmentation.** Introduce slack variables $\boldsymbol{\mu} \in [0, +\infty)^K$ and rewrite the inequality constraints as equalities:

$$\mathbf{c}(\mathbf{x}, \boldsymbol{\mu}) := \mathbf{h}(\mathbf{x}) + \boldsymbol{\mu} = \mathbf{0}. \tag{18}$$

The augmented state is $(\mathbf{x}, \boldsymbol{\mu}) \in \mathcal{D} := \mathcal{X} \times [0, +\infty)^K \subseteq \mathbb{R}^N$, where $N = S + K$.

**Definition 1** (Constraint Manifold)**.** *The constraint manifold is defined as:*

$$\mathcal{M} := \{(\mathbf{x}, \boldsymbol{\mu}) \in \mathcal{D} : \mathbf{c}(\mathbf{x}, \boldsymbol{\mu}) = \mathbf{0}\}. \tag{19}$$

**Proposition 2** (Manifold Structure)**.** *Under Assumption 2 below, $\mathcal{M}$ is an $S$-dimensional submanifold embedded in $\mathbb{R}^N$.*

*Proof.* The Jacobian of $\mathbf{c}$ is $J_c(\mathbf{x}, \boldsymbol{\mu}) = [J_h(\mathbf{x}) \quad \mathbb{I}_K]$, which has constant rank $K$ due to the identity block. By Theorem 2, $\mathcal{M}$ is an embedded submanifold of codimension $K$, hence dimension $N - K = S$. □

**Remark 1** (Projection to safe set)**.** *The safe set $\mathcal{C}$ is the projection of $\mathcal{M}$ onto the original state space: for any $(\mathbf{x}, \boldsymbol{\mu}) \in \mathcal{M}$, we have $\mathbf{h}(\mathbf{x}) = -\boldsymbol{\mu} \leq \mathbf{0}$, hence $\mathbf{x} \in \mathcal{C}$.*

**Tangent space.** The tangent space of $\mathcal{M}$ at $(\mathbf{x}, \boldsymbol{\mu})$ is:

$$\mathrm{T}_{(\mathbf{x},\boldsymbol{\mu})}\mathcal{M} = \ker J_c(\mathbf{x}, \boldsymbol{\mu}) = \{\mathbf{v} \in \mathbb{R}^N : J_c(\mathbf{x}, \boldsymbol{\mu})\, \mathbf{v} = \mathbf{0}\}. \tag{20}$$

A basis of this tangent space can be expressed in matrix form as $B(\mathbf{x}, \boldsymbol{\mu}) \in \mathbb{R}^{N \times S}$, satisfying $J_c\, B = \mathbf{0}$.

### B.3 Augmented Dynamics and Safe Control Law

**Slack variable dynamics.** We endow each slack variable with controlled dynamics:

$$\dot{\mu}_c = \alpha_c(\mu_c)\, u_{\mu,c}, \quad c \in \{1, \ldots, K\}, \tag{21}$$

where $\alpha_c : [0, +\infty) \to [0, +\infty)$ is a class-$\mathcal{K}$ function (continuous, strictly increasing, $\alpha_c(0) = 0$) that is locally Lipschitz, and $u_{\mu,c}$ is a virtual control input. In practice, we use the exponential form $\alpha(\mu) = \exp(\beta\mu) - 1$.

[Positivity of Slack Variables] Consider $\dot{\mu} = \alpha(\mu)u_\mu$ with $\alpha$ of class $\mathcal{K}$ and locally Lipschitz, $u_\mu \in [u_\mu^-, u_\mu^+]$ with $u_\mu^- < 0 < u_\mu^+$. For every initial state $\mu(0) > 0$, there exists $\epsilon > 0$ such that $\mu(t) \geq \epsilon$ for all $t \geq 0$.

*Proof.* Since $\alpha$ is $L$-Lipschitz with $\alpha(0) = 0$, we have $|\alpha(\mu)| \leq L\mu$. For any $u_\mu \in [u_\mu^-, u_\mu^+]$: $\dot{\mu} = \alpha(\mu)u_\mu \geq \alpha(\mu)u_\mu^- \geq Lu_\mu^-\mu$, where the second inequality uses $\alpha(\mu) > 0$ and $u_\mu^- < 0$. Setting $L' = Lu_\mu^-$, the comparison principle gives $\mu(t) \geq \mu(0)e^{L't} > 0$ for all $t \geq 0$. □

**Augmented system.** Combining the original dynamics equation 4 with the slack dynamics equation 21:

$$\begin{bmatrix}\dot{\mathbf{x}}\\ \dot{\boldsymbol{\mu}}\end{bmatrix} = \begin{bmatrix}f(\mathbf{x})\\ \mathbf{0}\end{bmatrix} + \begin{bmatrix}G(\mathbf{x}) & \mathbf{0}\\ \mathbf{0} & A(\boldsymbol{\mu})\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ \mathbf{u}_\mu\end{bmatrix}, \tag{22}$$

where $A(\boldsymbol{\mu}) = \mathrm{diag}(\alpha_1(\mu_1), \ldots, \alpha_K(\mu_K))$.

**Deriving the safe control law.** To ensure the augmented state remains on $\mathcal{M}$, we require $\dot{\mathbf{c}} = \mathbf{0}$:

$$\dot{\mathbf{c}} = J_c\begin{bmatrix}\dot{\mathbf{x}}\\ \dot{\boldsymbol{\mu}}\end{bmatrix} = \underbrace{J_h(\mathbf{x})f(\mathbf{x})}_{\boldsymbol{\psi}(\mathbf{x})} + \underbrace{[J_h(\mathbf{x})G(\mathbf{x}) \quad A(\boldsymbol{\mu})]}_{J_u(\mathbf{x},\boldsymbol{\mu})}\begin{bmatrix}\mathbf{u}\\ \mathbf{u}_\mu\end{bmatrix} = \mathbf{0}. \tag{23}$$

Here $\boldsymbol{\psi}(\mathbf{x}) = J_h(\mathbf{x})f(\mathbf{x})$ is the *constraint drift* induced by the system drift. The general solution is:

$$\begin{bmatrix}\mathbf{u}\\ \mathbf{u}_\mu\end{bmatrix} = -J_u^\dagger\boldsymbol{\psi} + B_u\mathbf{u}_{\mathrm{ref}}, \tag{24}$$

where $J_u^\dagger$ is the Moore–Penrose pseudoinverse, $B_u$ is a basis of $\ker J_u$ (i.e., $J_uB_u = \mathbf{0}$), and $\mathbf{u}_{\mathrm{ref}}$ is an arbitrary reference input (from the task policy or subgoal tracking controller).

Adding a contraction term to ensure convergence to the manifold from nearby states, the control law is:

$$\begin{bmatrix}\mathbf{u}_s\\ \mathbf{u}_\mu\end{bmatrix} = \underbrace{-J_u^\dagger\boldsymbol{\psi}}_{\text{drift compensation}} \underbrace{-\lambda J_u^\dagger\mathbf{c}}_{\text{contraction}} + \underbrace{B_u\mathbf{u}_{\mathrm{ref}}}_{\text{tangential}}, \tag{25}$$

where $\lambda > 0$ is the contraction gain. The first term compensates for the constraint drift, the second drives the state toward $\mathcal{M}$ when $\mathbf{c} \neq \mathbf{0}$, and the third enables task-directed motion along the manifold.

### B.4 Single Agent Safety Guarantees

This section establishes the main safety result: under mild assumptions, the constraint manifold controller equation 25 renders the constraint manifold $\mathcal{M}$ attractive, so that every trajectory starting sufficiently close to $\mathcal{M}$ converges to it and, once on $\mathcal{M}$, remains there indefinitely. Since $\mathcal{M}$ projects onto the safe set $\mathcal{C}$ (Remark 1), this guarantees constraint satisfaction $\mathbf{h}(\mathbf{x}(t)) \leq \mathbf{0}$ for all $t \geq 0$.

**Proof strategy.** The argument has three ingredients:

1. A Lyapunov-like function $V(\mathbf{x}, \boldsymbol{\mu}) = \frac{1}{2}\mathbf{c}^\top\mathbf{c}$ that measures the distance from $\mathcal{M}$: $V = 0$ if and only if $(\mathbf{x}, \boldsymbol{\mu}) \in \mathcal{M}$.
2. A direct computation showing that the constraint manifold controller yields $\dot{V} \leq 0$, so $V$ is non-increasing along trajectories.
3. An application of LaSalle's Invariance Principle (Theorem B.1) to conclude that trajectories actually converge to $\mathcal{M}$, not just to some lower level set of $V$.

We first state the assumptions needed to make this argument rigorous, then define the region over which convergence is guaranteed, and finally present the theorem and its proof.

**Assumption 1** (Compact State Space)**.** *The state space $\mathcal{X}$ is compact and positively invariant with respect to equation 4.*

**Assumption 2** (Smooth Constraints)**.** *The constraint function $\mathbf{h}$ is of class $C^1$.*

**Assumption 3** (Non-empty Safe Set)**.** *The constraint manifold $\mathcal{M}$ defined in equation 19 is non-empty.*

**Assumption 4** (Constraint Qualification)**.** *For all $(\mathbf{x}, \boldsymbol{\mu}) \in \partial\mathcal{M}$, the submatrix $(J_h(\mathbf{x})G(\mathbf{x}))_{[\iota_{h=0},:]}$ has rank $|\iota_{h=0}(\mathbf{x})|$, where $\iota_{h=0}(\mathbf{x}) = \{c : h_c(\mathbf{x}) = 0\}$ is the index set of active constraints.*

Assumptions 1–3 are standard. Assumption 4 is the key constraint qualification: it ensures that $J_u$ has full row rank on $\mathcal{M}$, which in turn guarantees that the linear system equation 23 is solvable and that $J_u J_u^\dagger = \mathbb{I}_K$.

**Remark 2.** *Assumption 4 implies that the number of simultaneously active constraints cannot exceed the control dimension: $|\iota_{h=0}(\mathbf{x})| \leq U$. For the 2D navigation agents with $U = 2$, at most two constraints can be simultaneously active at the boundary.*

**Region of contraction.** LaSalle's Invariance Principle requires a compact positively invariant set on which to apply the analysis. We construct such a set using sublevel sets of the Lyapunov function, but must first exclude a pathological set where the contraction mechanism can fail.

**Definition 2** (Singular Set)**.** *The* singular set *is*

$$\mathcal{Y} := \{(\mathbf{x}, \boldsymbol{\mu}) \in \mathcal{D} : J_u^\top \mathbf{c} = \mathbf{0},\ \|\boldsymbol{\mu}\| = 0,\ \|\mathbf{c}\| \neq 0\}.$$

At points in $\mathcal{Y}$, the contraction term $-\lambda J_u^\dagger \mathbf{c}$ vanishes even though $\mathbf{c} \neq \mathbf{0}$, so the controller cannot drive the state back toward $\mathcal{M}$. Crucially, when $\mathcal{Y}$ is non-empty, it is disjoint from $\mathcal{M}$ with positive distance: $\operatorname{dist}(\mathcal{M}, \mathcal{Y}) > 0$.

Define the Lyapunov-like function

$$V(\mathbf{x}, \boldsymbol{\mu}) = \tfrac{1}{2} \mathbf{c}^\top \mathbf{c}, \tag{26}$$

and choose $\eta$ such that $0 < \eta < \inf\{V(\mathbf{x}, \boldsymbol{\mu}) : (\mathbf{x}, \boldsymbol{\mu}) \in \mathcal{Y}\}$ (or $\eta$ arbitrarily large if $\mathcal{Y} = \emptyset$). The *region of contraction*

$$\Omega_\eta := \{(\mathbf{x}, \boldsymbol{\mu}) \in \mathcal{D} : V(\mathbf{x}, \boldsymbol{\mu}) \leq \eta\} \tag{27}$$

is compact, contains $\mathcal{M}$ (since $V = 0$ on $\mathcal{M}$), and is disjoint from $\mathcal{Y}$ by construction.

**Main safety theorem.** The proof of the main theorem relies on the following elementary lemma about pseudoinverses, which allows us to identify the set where $\dot{V}$ vanishes.

[Pseudoinverse Kernel] Let $X \in \mathbb{R}^{m \times n}$ with $\operatorname{rank}(X) \leq m \leq n$, and let $\mathbf{x} \in \mathbb{R}^m$. If $\mathbf{x}^\top X X^\dagger \mathbf{x} = 0$, then $X^\top \mathbf{x} = \mathbf{0}$.

*Proof.* Let $X = U_1 \Sigma_1 V_1^\top$ be the compact SVD, where $\Sigma_1$ contains the positive singular values. The pseudoinverse is $X^\dagger = V_1 \Sigma_1^{-1} U_1^\top$, so $X X^\dagger = U_1 U_1^\top$ is the orthogonal projector onto the column space of $X$. The assumption $\mathbf{x}^\top U_1 U_1^\top \mathbf{x} = \|U_1^\top \mathbf{x}\|^2 = 0$ gives $U_1^\top \mathbf{x} = \mathbf{0}$. Since $X^\top = V_1 \Sigma_1 U_1^\top$, we conclude $X^\top \mathbf{x} = V_1 \Sigma_1 (U_1^\top \mathbf{x}) = \mathbf{0}$. □

**Theorem 3** (Per-Agent Safety (Liu et al., 2025))**.** *Consider the augmented system equation 22 equipped with the constraint manifold controller equation 25. Under Assumptions 1–4, and provided a feasible control $\mathbf{u}_s \in \mathcal{U}$ satisfying equation 23 exists throughout $\Omega_\eta$, the following hold:*

1. (Attractivity) *Every trajectory starting in $\Omega_\eta$ approaches the constraint manifold $\mathcal{M}$ as $t \to +\infty$.*
2. (Forward invariance) *If $(\mathbf{x}(0), \boldsymbol{\mu}(0)) \in \mathcal{M}$, then $(\mathbf{x}(t), \boldsymbol{\mu}(t)) \in \mathcal{M}$ for all $t \geq 0$, and consequently $\mathbf{h}(\mathbf{x}(t)) \leq \mathbf{0}$ for all $t \geq 0$.*

*Proof.* We apply LaSalle's Invariance Principle on the compact set $\Omega_\eta$.

**Step 1: Lyapunov decrease.** Define the Lyapunov-like function $V(\mathbf{s}, \boldsymbol{\mu}) = \frac{1}{2}\|\mathbf{c}\|^2$, where $\mathbf{c} = \mathbf{h}(\mathbf{s}) + \boldsymbol{\mu}$. Its time derivative is

$$\dot{V} = \mathbf{c}^\top \dot{\mathbf{c}} = \mathbf{c}^\top \Big[ \boldsymbol{\psi} + J_u \begin{bmatrix} \mathbf{u}_s \\ \mathbf{u}_\mu \end{bmatrix} \Big]. \tag{28}$$

Substituting the constraint manifold control law equation 25:

$$\begin{aligned} \dot{V} &= \mathbf{c}^\top \Big[ \boldsymbol{\psi} + J_u \big( - J_u^\dagger \boldsymbol{\psi} - \lambda\, J_u^\dagger \mathbf{c} + B_u\, \mathbf{u}_{\mathrm{ref}} \big) \Big] \\ &= \mathbf{c}^\top \big( \underbrace{\boldsymbol{\psi} - J_u J_u^\dagger \boldsymbol{\psi}}_{(a)} \underbrace{- \lambda\, J_u J_u^\dagger \mathbf{c}}_{(b)} + \underbrace{J_u B_u\, \mathbf{u}_{\mathrm{ref}}}_{(c)} \big). \end{aligned} \tag{29}$$

Term (a) vanishes when the tangency condition equation 23 holds, i.e., $\boldsymbol{\psi} - J_u J_u^\dagger \boldsymbol{\psi} = \mathbf{0}$. Term (c) vanishes since $J_u B_u = \mathbf{0}$ by construction of the null-space basis. The remaining term gives

$$\dot{V} = -\lambda\, \mathbf{c}^\top J_u J_u^\dagger\, \mathbf{c} \leq 0, \tag{30}$$

since $J_u J_u^\dagger$ is positive semidefinite and $\lambda > 0$.

**Step 2: Positive invariance of $\Omega_\eta$.** Since $\dot{V} \leq 0$ throughout $\Omega_\eta$ and $\Omega_\eta = \{(\mathbf{s}, \boldsymbol{\mu}) \in \mathcal{D} : V(\mathbf{s}, \boldsymbol{\mu}) \leq \eta\}$ is compact (Assumption 1), the sublevel set $\Omega_\eta$ is positively invariant.

**Step 3: Attractivity via LaSalle.** By LaSalle's Invariance Principle, every trajectory starting in $\Omega_\eta$ converges to the largest invariant set contained in

$$\mathcal{E} := \{(\mathbf{s}, \boldsymbol{\mu}) \in \Omega_\eta : \dot{V} = 0\} = \{(\mathbf{s}, \boldsymbol{\mu}) \in \Omega_\eta : \mathbf{c}^\top J_u J_u^\dagger\, \mathbf{c} = 0\}. \tag{31}$$

By Lemma B.4, $\mathbf{c}^\top J_u J_u^\dagger\, \mathbf{c} = 0$ implies $J_u^\top \mathbf{c} = \mathbf{0}$. Under Assumption 4, $J_u$ has full row rank on $\mathcal{M}$, so $J_u^\top \mathbf{c} = \mathbf{0}$ with $\mathbf{c} \in \mathrm{row}(J_u)$ forces $\mathbf{c} = \mathbf{0}$. Therefore $\mathcal{E} \cap \{\boldsymbol{\mu} \geq \mathbf{0}\} = \mathcal{M}$, and $\mathcal{M}$ is the largest invariant set in $\mathcal{E}$. This establishes attractivity.

**Step 4: Forward invariance of $\mathcal{M}$.** If $(\mathbf{s}(0), \boldsymbol{\mu}(0)) \in \mathcal{M}$, then $\mathbf{c}(0) = \mathbf{0}$ and $V(0) = 0$. Since $V \geq 0$ and $\dot{V} \leq 0$:

$$0 \leq V(t) = V(0) + \int_0^t \dot{V}(\tau)\, d\tau = \int_0^t \underbrace{\dot{V}(\tau)}_{\leq 0}\, d\tau \leq 0. \tag{32}$$

Hence $V(t) = 0$ for all $t \geq 0$, which implies $\mathbf{c}(t) = \mathbf{0}$. By the projection property (Remark 1), $\mathbf{h}(\mathbf{s}(t)) = -\boldsymbol{\mu}(t) \leq \mathbf{0}$ for all $t \geq 0$. □

### B.5 Multi-agent Safety Guarantees

**Setup.** Consider $N$ agents with decoupled control-affine dynamics $\dot{\mathbf{s}}_i = f_i(\mathbf{s}_i) + G_i(\mathbf{s}_i)\mathbf{u}_i$. Each agent $i$ maintains a local constraint $\mathbf{h}_i(\mathbf{s}_i, \mathbf{s}_{\mathcal{N}_i}) \leq \mathbf{0}$, where $\mathcal{N}_i = \mathcal{E}_i \cup \mathcal{A}_i$ comprises obstacles $\mathcal{E}_i$ and neighboring agents $\mathcal{A}_i$. The constraint vector stacks pairwise collision terms with all entities in $\mathcal{N}_i$, and each agent applies the constraint manifold safe controller equation 10 independently on its local constraint manifold $\mathcal{M}_i$. The global safe set is $\mathcal{C}_{\text{global}} := \bigcap_{i \in \mathcal{N}} \mathcal{C}_i$, where $\mathcal{C}_i = \{\mathbf{S} : \mathbf{h}_i \leq \mathbf{0}\}$.

**Assumptions.** Our analysis requires four conditions, all naturally satisfied by MAS with distance-based collision constraints:

(i) The single-agent safety assumptions (Assumptions 1–4) hold for each agent's local system.
(ii) *Decoupled control:* each $\mathbf{u}_i$ affects only agent $i$'s dynamics.
(iii) *Symmetric inter-agent constraints:* $h_{ij} = h_{ji}$ for every agent pair $(i, j)$ with $j \in \mathcal{A}_i$.
(iv) *Local observability:* each agent observes the positions and velocities of all entities in $\mathcal{N}_i$.

**Theorem 4** (Compositional Safety)**.** *Under assumptions (i)–(iv), if every agent starts on its local constraint manifold, i.e., $(\mathbf{s}_i(0), \mathbf{s}_{\mathcal{N}_i}(0), \boldsymbol{\mu}_i(0)) \in \mathcal{M}_i$ for all $i \in \mathcal{N}$, then every agent remains on its local manifold for all $t \geq 0$, and the joint trajectory satisfies $\mathbf{S}(t) \in \mathcal{C}_{\text{global}}$ for all $t \geq 0$.*

*Proof.* The proof extends the single-agent safety result (Theorem 3) to the multi-agent setting via a per-agent Lyapunov argument.

**Step 1: Multi-agent constraint dynamics.** Each agent $i$ defines a constraint residual $\mathbf{c}_i = \mathbf{h}_i(\mathbf{s}_i, \mathbf{s}_{\mathcal{N}_i}) + \boldsymbol{\mu}_i$ and a local Lyapunov function $V_i = \frac{1}{2}\|\mathbf{c}_i\|^2$. Differentiating $\mathbf{c}_i$ with respect to time:

$$\dot{\mathbf{c}}_i = \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_i}\dot{\mathbf{s}}_i + \underbrace{\sum_{e \in \mathcal{E}_i} \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_e}\dot{\mathbf{s}}_e}_{\text{obstacle motion}} + \underbrace{\sum_{j \in \mathcal{A}_i} \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_j}\dot{\mathbf{s}}_j}_{\text{other agents}} + \dot{\boldsymbol{\mu}}_i = \boldsymbol{\psi}_i + J_{u,i} \begin{bmatrix} \mathbf{u}_{s,i} \\ \mathbf{u}_{\mu,i} \end{bmatrix}, \tag{33}$$

where the drift and control Jacobian are

$$\boldsymbol{\psi}_i = \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_i} f_i(\mathbf{s}_i) + \sum_{e \in \mathcal{E}_i} \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_e}\dot{\mathbf{s}}_e + \sum_{j \in \mathcal{A}_i} \frac{\partial \mathbf{h}_i}{\partial \mathbf{s}_j}\dot{\mathbf{s}}_j, \qquad J_{u,i} = \begin{bmatrix} \dfrac{\partial \mathbf{h}_i}{\partial \mathbf{s}_i} G_i(\mathbf{s}_i) & \mathbb{I} \end{bmatrix}. \tag{34}$$

Since control is decoupled, $J_{u,i}$ involves only agent $i$'s own control input $\mathbf{u}_i$ and slack rate $\dot{\boldsymbol{\mu}}_i$. The drift $\boldsymbol{\psi}_i$ absorbs all terms not controlled by agent $i$: its autonomous dynamics, obstacle motions ($\dot{\mathbf{s}}_e = 0$ for static obstacles, or known trajectories for dynamic ones), and the velocities of other agents produced by their own policies. In the single-agent setting $\mathcal{A}_i = \emptyset$ and only the obstacle term remains; the multi-agent extension adds the inter-agent term, but crucially both enter only the drift. Agent $i$ can evaluate $\boldsymbol{\psi}_i$ at each instant via local observation (Assumption (iv)).

**Step 2: Per-agent Lyapunov decrease.** Substituting the constraint manifold control law equation 10 into $\dot{V}_i = \mathbf{c}_i^\top \dot{\mathbf{c}}_i$:

$$\dot{V}_i = \mathbf{c}_i^\top \Big( \boldsymbol{\psi}_i + J_{u,i} \Big( - J_{u,i}^\dagger \boldsymbol{\psi}_i - \lambda \, J_{u,i}^\dagger \mathbf{c}_i + B_{u,i} \, \mathbf{u}_{\mathrm{ref},i} \Big) \Big). \tag{35}$$

Expanding yields three terms: (a) the drift residual $\mathbf{c}_i^\top (\boldsymbol{\psi}_i - J_{u,i} J_{u,i}^\dagger \boldsymbol{\psi}_i)$ vanishes because Assumption 4 ensures $J_{u,i}$ has full row rank, so $J_{u,i} J_{u,i}^\dagger = I$ and the drift is cancelled exactly—independently of the obstacle and inter-agent velocity terms in $\boldsymbol{\psi}_i$; (b) the tangential term $\mathbf{c}_i^\top J_{u,i} B_{u,i} \, \mathbf{u}_{\mathrm{ref},i} = 0$ since $B_{u,i}$ spans the null space of $J_{u,i}$; (c) the remaining contraction term gives

$$\dot{V}_i = -\lambda \, \mathbf{c}_i^\top J_{u,i} J_{u,i}^\dagger \, \mathbf{c}_i \leq 0, \tag{36}$$

since $J_{u,i} J_{u,i}^\dagger$ is positive semidefinite and $\lambda > 0$.

**Step 3: Forward invariance.** Each agent starts on its local constraint manifold, so $\mathbf{c}_i(0) = \mathbf{0}$ and $V_i(0) = 0$. Combined with $V_i \geq 0$ and $\dot{V}_i \leq 0$:

$$0 \leq V_i(t) = V_i(0) + \int_0^t \dot{V}_i(\tau) \, d\tau = \int_0^t \underbrace{\dot{V}_i(\tau)}_{\leq 0} \, d\tau \leq 0. \tag{37}$$

Hence $V_i(t) = 0$ for all $t \geq 0$, which implies $\mathbf{c}_i(t) = \mathbf{0}$ and, by the projection property (Remark 1), $\mathbf{h}_i(t) = -\boldsymbol{\mu}_i(t) \leq \mathbf{0}$ for every agent $i$.

**Step 4: Global safety via symmetry.** For any agent pair $(i, j)$ with $j \in \mathcal{A}_i$, the collision constraint $h_{ij}$ appears in agent $i$'s local constraint set $\mathbf{h}_i$, and $h_{ji} = h_{ij}$ appears in agent $j$'s set $\mathbf{h}_j$. Since every agent satisfies its local constraints (Step 3), every pairwise collision constraint is enforced, yielding $\mathbf{S}(t) \in \bigcap_i \mathcal{C}_i = \mathcal{C}_{\mathrm{global}}$ for all $t \geq 0$. □

## C Analysis of Residual Safety Violations

Our compositional safety guarantee (Theorem 1) is established in continuous time, under a fixed constraint set and on-manifold initialization. Several implementation factors cause mild deviations from these conditions, accounting for the small ($< 2\%$) violations observed in practice.

**(1) Discrete-time integration.** The control law guarantees $\dot{V} \leq 0$ in continuous time, but each discrete integration step induces a drift in the residual $c$ on the order of the step size, momentarily violating $h$; the magnitude shrinks as the step size decreases.

**(2) Neighbor truncation (top-$k$).** Each agent constrains only its $k$ nearest entities. Entities beyond $k$ are excluded from $h_i$ and receive no guarantee, and the top-$k$ set switches as agents move, so newly activated constraints need not satisfy $c = 0$ at the switching instant.

**(3) Degeneracy of Assumption 4.** When the number of active constraints exceeds the control dimension $U$, $J_u$ loses full row rank and the projection yields a least-squares rather than an exact drift cancellation, leaving a small residual drift.

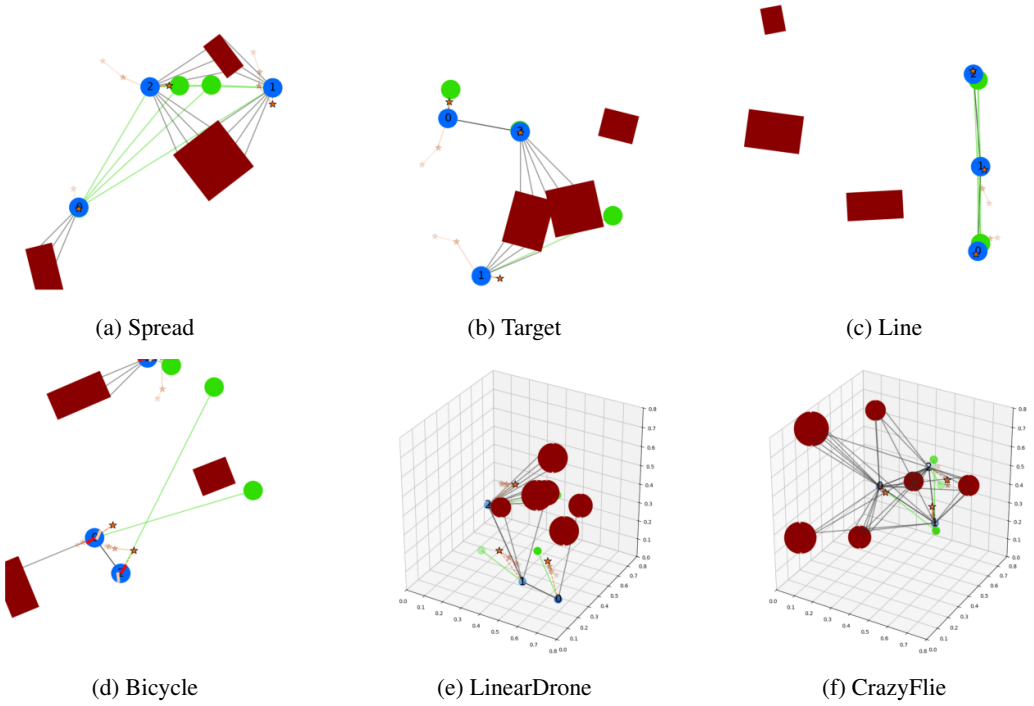
**(4) Partial observability.** Neighbor states are estimated from local observations, and the estimation error enters the drift $\psi_i$, making the tangent-space projection slightly inexact.

## D Environment Descriptions

We evaluate our method on six multi-agent environments with LiDAR-based obstacle perception, covering a spectrum of dynamics ranging from simple 2D point-mass systems to higher-dimensional

non-holonomic and quadrotor platforms. LidarSpread and LidarTarget adopt 2D double-integrator dynamics (state $[x, y, \dot{x}, \dot{y}]$, action $[a_x, a_y]$), differing in their goal-assignment semantics. In LidarSpread, $N$ agents are required to collectively cover $N$ goal locations (any-to-any matching), whereas in LidarTarget each agent is assigned a fixed goal ($i \leftrightarrow i$ matching). LidarLine extends this setup by replacing discrete goal nodes with two landmarks, between which $N$ goal positions are linearly interpolated, requiring agents to form an evenly spaced line. LidarBicycleTarget retains the target-based assignment but introduces non-holonomic kinematic bicycle dynamics (5D state including heading and forward velocity).

LinearDrone models a 3D quadrotor with linear dynamics, using a 6D state $[x, y, z, \dot{x}, \dot{y}, \dot{z}]$ and 3D acceleration control, with aerodynamic damping along each axis. CrazyFlie represents a more challenging setting with full nonlinear quadrotor dynamics and realistic physical parameters. Each agent has a 12D state including position, orientation, and velocities.



(a) Spread (b) Target (c) Line

(d) Bicycle (e) LinearDrone (f) CrazyFlie

Figure 5: Visualization of different LiDAR-based multi-agent environments.

# E Additional Experiments

## E.1 Computational efficiency

We compare the per-iteration training cost of the QP-based CBF controller and the constraint manifold controller on a single NVIDIA H200 GPU, with all training hyperparameters held constant. As shown in Table 1, the constraint manifold controller achieves approximately $14.4\times$ higher throughput than the CBF baseline. This speedup arises because the manifold controller computes safety corrections via a closed-form pseudoinverse projection, whereas the CBF controller requires solving a quadratic program (QP) at every timestep. Over the full 200k training iterations, this translates to an estimated training time of $\sim$9.4 hours for the manifold controller versus $\sim$136.8 hours for the CBF controller.

## E.2 Additional generalization results

We evaluate generalization across all environments. As shown in Figure 6, **HMM** achieves the best generalization in safety rate: it maintains near-100% safety even with a large number of agents or

Table 1: Per-iteration training speed comparison between the QP-based CBF controller and the constraint manifold controller on a single NVIDIA H200 GPU with identical training hyperparameters.

| **Method** | **Speed** | **Est. Total (200k iters)** |
|---|---|---|
| QP-based CBF | $\sim$0.4 it/s | $\sim$136.8 hours |
| Constraint Manifold | $\sim$5.9 it/s | $\sim$9.4 hours |
| **Speedup** | | **14.4**$\times$ |

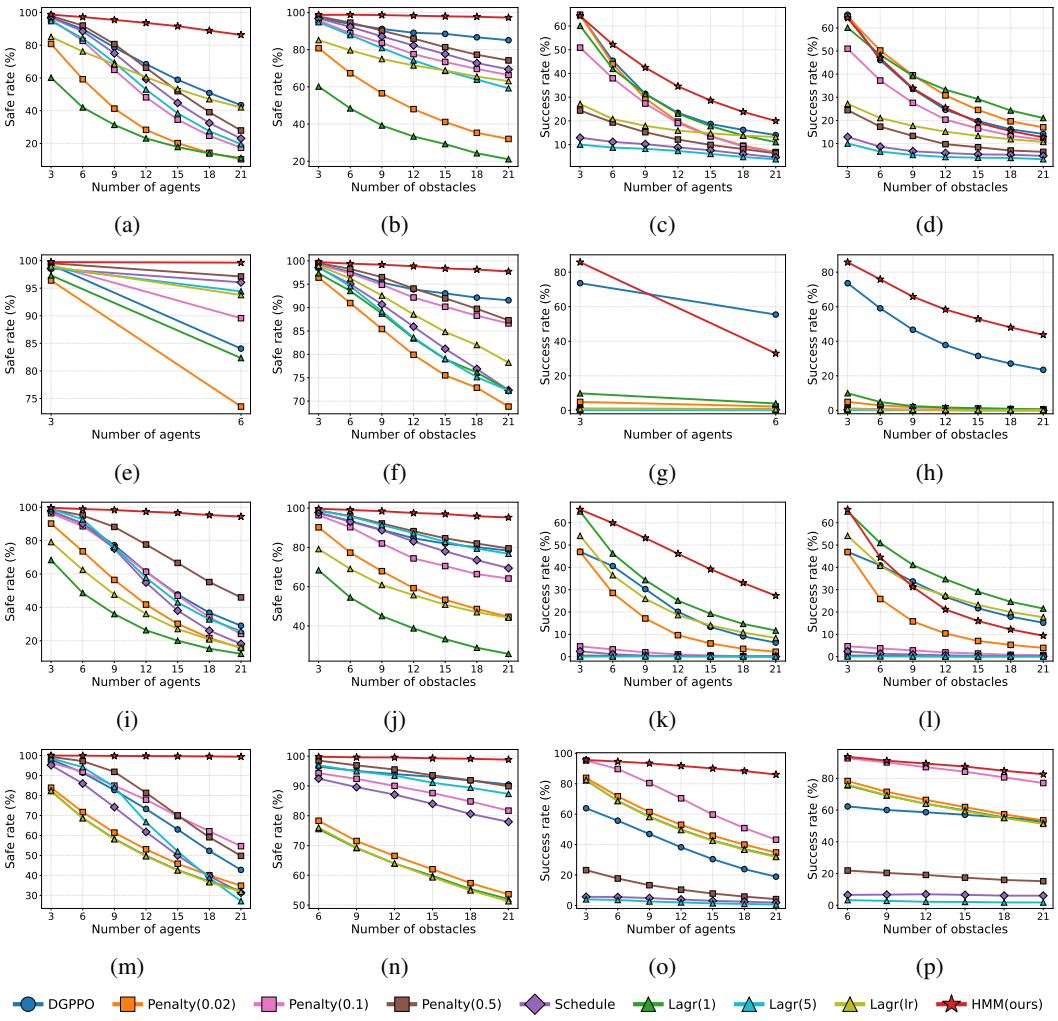



Figure 6: Generalization results for remaining environments. LidarBicycleTarget (a–d), LidarLine (e–h), LidarTarget (i–l), and LinearDrone (m–p). Left two columns: safe rate; right two columns: success rate.

obstacles, whereas other baselines degrade significantly. In terms of success rate, HMM also performs best in most environments.

### E.3 Ablation Study

To validate the contribution of each component in our framework, we compare three variants: (1) Nominal only, which uses a simple proportional controller to drive agents toward their goals without any safety mechanism; (2) Nominal + Manifold, which applies the constraint manifold

Table 2: Comparison of nominal-only, nominal+manifold, and our HMM method across six environments. All values are percentages (mean ± std across seeds, where applicable). Bold marks the best safe rate and success rate per environment.

| Environment | Method | Safe rate (%) ↑ | Success rate (%) ↑ |
|---|---|---|---|
| LidarSpread | Nominal only | 32.23 | 32.17 |
| | Nominal + Manifold | 97.60 | 49.17 |
| | **HMM (ours)** | **99.83 ± 0.06** | **87.29 ± 1.35** |
| LidarTarget | Nominal only | 62.43 | 59.90 |
| | Nominal + Manifold | 98.77 | 70.97 |
| | **HMM (ours)** | **99.67 ± 0.05** | **71.93 ± 0.02** |
| LidarLine | Nominal only | 33.30 | 33.13 |
| | Nominal + Manifold | 96.87 | 48.20 |
| | **HMM (ours)** | **99.73 ± 0.03** | **85.75 ± 1.15** |
| LidarBicycle | Nominal only | 60.97 | 59.17 |
| | Nominal + Manifold | 97.17 | 54.67 |
| | **HMM (ours)** | **98.58 ± 0.31** | **64.67 ± 3.93** |
| LinearDrone | Nominal only | 74.43 | 40.00 |
| | Nominal + Manifold | 99.57 | 42.87 |
| | **HMM (ours)** | **99.80 ± 0.14** | **93.67 ± 0.80** |
| CrazyFlie | Nominal only | 74.20 | 74.20 |
| | Nominal + Manifold | 98.63 | 94.93 |
| | **HMM (ours)** | **99.87 ± 0.00** | **99.22 ± 0.44** |

safe controller on top of the nominal controller but without learned high-level subgoal planning; and (3) HMM (ours), the full framework with both learned high-level subgoals and the low-level manifold controller. As shown in Table 2, the nominal-only baseline achieves poor safety across all environments, confirming that naive goal-tracking is insufficient in multi-agent settings. Adding the manifold controller significantly improves safety (above 96% in all cases), but without high-level coordination, agents frequently get stuck or take suboptimal paths, resulting in limited success rates. Our full HMM framework achieves the highest performance on both metrics, demonstrating that the learned high-level planner is essential for effective coordination while the low-level manifold controller provides the safety foundation.

## F HYPERPARAMETERS

We summarize the key hyperparameters used in our experiments. Table 3 lists the training hyperparameters for all environments, including PPO settings, subgoal configuration, and environment specifications. Table 4 lists the constraint manifold controller hyperparameters. All values are fixed across environments unless otherwise noted.

Table 3: Training hyperparameters.

| **Parameter** | **Value** |
|---|---|
| Discount $\gamma$ | 0.99 |
| GAE $\lambda$ | 0.95 |
| PPO clip $\epsilon$ | 0.25 |
| Entropy coefficient | $1 \times 10^{-2}$ |
| Actor learning rate | $3 \times 10^{-4}$ |
| Critic learning rate | $1 \times 10^{-3}$ |
| Total environment steps | $2 \times 10^{5}$ |
| Subgoal interval $\tau$ | 8 |
| Relative subgoal max $\Delta$ | 0.2 |
| Episode length $T$ | 128 (3D env: 256) |
| Number of agents | 3 |
| Number of obstacles | 3 (3D env: 6) |

Table 4: Constraint manifold controller hyperparameters.

| **Parameter** | **Symbol** | **Value** |
|---|---|---|
| Top-$k$ nearest neighbors | $k$ | 3 |
| Viability gain | $K$ | 0.5 |
| Error correction gain | $K_c$ | 30.0 |
| Null-space control bound | $\alpha_{\max}$ | 3.0 |
| Constraint activation threshold | $g_{\text{act}}$ | 0.02 |
| Safety margin | – | 0.02 |
| Slack weight | $w_s$ | 10.0 |
| Slack lower bound | $s_{\min}$ | 0.1 |
| Configuration-space dimension | $d_q$ | 2 (drones: 3) |