MAD-PINN: A Decentralized Physics-Informed Machine Learning Framework for Safe and Optimal Multi-Agent Control

Manan Tayal*[1], Aditya Singh*[1], Shishir Kolathaya[1], Somil Bansal[2]

***Abstract*— Co-optimizing safety and performance in large-scale multi-agent systems remains a fundamental challenge. Existing approaches based on multi-agent reinforcement learning (MARL), safety filtering, or Model Predictive Control (MPC) either lack strict safety guarantees, suffer from conservatism, or fail to scale effectively. We propose *MAD-PINN*, a decentralized physics-informed machine learning framework for solving the multi-agent state-constrained optimal control problem (MASC-OCP). Our method leverages an epigraph-based reformulation of SC-OCP to simultaneously capture performance and safety, and approximates its solution via a physics-informed neural network. Scalability is achieved by training the SC-OCP value function on reduced-agent systems and deploying them in a decentralized fashion, where each agent relies only on local observations of its neighbours for decision-making. To further enhance safety and efficiency, we introduce an Hamilton-Jacobi (HJ) reachability-based neighbour selection strategy to prioritize safety-critical interactions, and a receding-horizon policy execution scheme that adapts to dynamic interactions while reducing computational burden. Experiments on multi-agent navigation tasks demonstrate that MAD-PINN achieves superior safety–performance trade-offs, maintains scalability as the number of agents grows, and consistently outperforms state-of-the-art baselines. Videos results can be viewed on the project webpage.**

I. Introduction

The deployment of autonomous systems in safety-critical domains such as aerial swarms [1], intelligent transportation [2] networks, and automated warehouses [3] has made multi-agent coordination a central problem in robotics and control. In these environments, multiple agents must operate in shared spaces and achieve collective objectives – such as routing, formation control, or exploration – while adhering to strict safety constraints. The joint requirement of balancing task performance with safety makes the synthesis of control policies in multi-agent systems a challenging problem.

Multi-agent reinforcement learning (MARL) [4]–[6] has emerged as a popular paradigm for policy learning in such settings. While MARL demonstrates strong performance in complex tasks, its safety treatment is insufficient. Safety is typically introduced via reward shaping, treating safety constraints as soft penalties. Constrained Markov Decision Process (CMDP) formulations [7], [8] provide a more principled approach, but they only ensure that constraint violations remain bounded on average, which is insufficient for safety-critical robotic applications where violations must be avoided at all times.

Control-theoretic methods such as Control Barrier Function (CBF) [9] and Hamilton-Jacobi (HJ) Reachability [10], [11] provide formal safety guarantees. These methods can act as safety filters [12] on top of existing nominal controllers, to minimally modify them to enforce safety. However, they suffer from scalability and conservatism in multi-agent settings. A popular approach is to compute pairwise safety filters and extend them to many agents; however, this approach remains ineffective, as the intersection of individual safe sets does not necessarily represent the true joint safe set [13]. Moreover, the myopic nature of safety filters often degrades task performance. Optimal control methods such as Model Predictive Control (MPC) [15], [16] and Model Predictive Path Integral control (MPPI) [17], [18] provide another line of solutions. These methods incorporate predictive look-ahead with explicit handling of hard constraints, and can flexibly handle nonlinear dynamics and diverse cost functions. These approaches have also been extended to multi-agent collision avoidance through the inclusion of modified cost terms or additional constraints [19]. Yet, despite their flexibility, MPC and MPPI do not offer formal safety guarantees. In interactive multi-agent scenarios, they may still yield unsafe trajectories, limiting their reliability in safety-critical domains.

A principled framework for unifying performance and safety is the state-constrained optimal control problem (SC-OCP), which formulates performance as cost minimization and safety as a strict state constraint. In the multi-agent setting, SC-OCP is particularly appealing because it directly encodes the dual objectives of collision-free coordination and task performance. However, solving SC-OCPs at scale is computationally formidable [22]. Epigraph-based reformulations recast the problem as a Hamilton-Jacobi-Bellman partial differential equation (HJB-PDE) [23], but the added dimensionality exacerbates the computational complexity. More critically, in centralized multi-agent scenarios, the joint state–action space grows exponentially with the number of agents, rendering classical numerical solvers impractical even for modest system sizes. Thus, while SC-OCP provides a theoretically sound foundation for safe multi-agent control, its direct application to large-scale systems remains infeasible without new strategies for decentralization and scalability.

To address these challenges, we propose *MAD-PINN*, a decentralized physics-informed machine learning framework for solving the multi-agent SC-OCP. At its core, MAD-PINN combines control-theoretic structure with neural approxima-

* Denotes equal contribution

[1] Authors are with the Center for Cyber-Physical Systems, Indian Institute of Science, Bangalore, India (email:{manantayal, adityasingh, shishirk}@iisc.ac.in)

[2] Author is with the Department of Aeronautics and Astronautics, Stanford University, USA (email:{somil@stanford.edu})

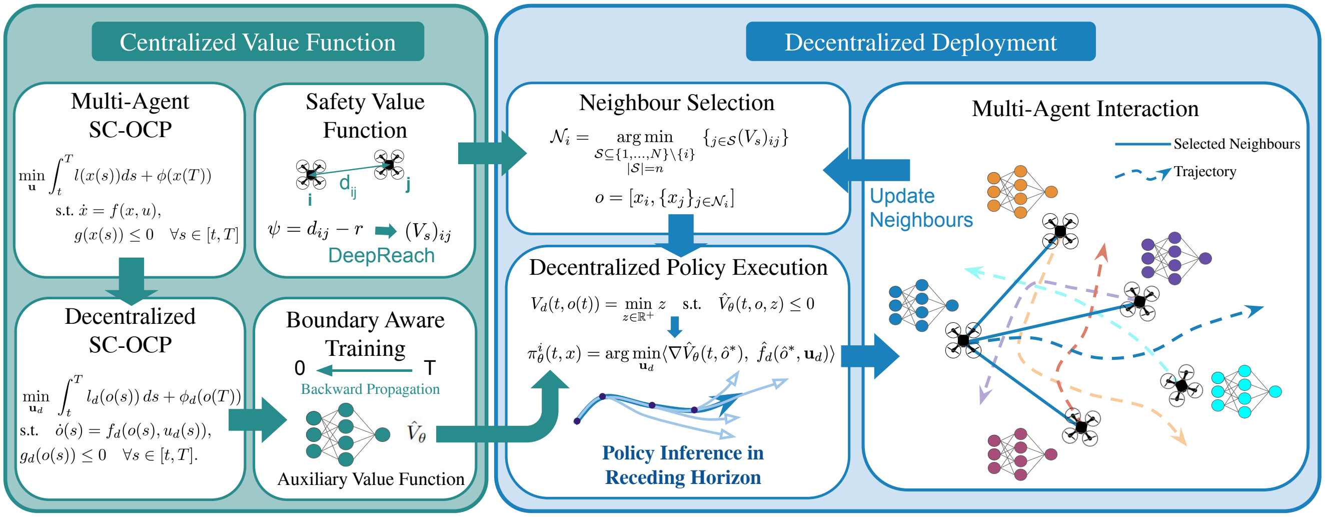
Centralized Value Function
Multi-Agent SC-OCP
$\min_{\mathbf{u}} \int_t^T l(x(s))ds + \phi(x(T))$
s.t. $\dot{x} = f(x,u)$, $g(x(s)) \leq 0 \quad \forall s \in [t,T]$
Safety Value Function
d_{ij}
i
j
$\psi = d_{ij} - r$ ➡ $(V_s)_{ij}$
DeepReach
Decentralized SC-OCP
$\min_{\mathbf{u}_d} \int_t^T l_d(o(s))\,ds + \phi_d(o(T))$
s.t. $\dot{o}(s) = f_d(o(s), u_d(s))$, $g_d(o(s)) \leq 0 \quad \forall s \in [t,T]$.
Boundary Aware Training
0
T
Backward Propagation
$\hat{V}_\theta$
Auxiliary Value Function
Decentralized Deployment
Neighbour Selection
$\mathcal{N}_i = \underset{\substack{\mathcal{S}\subseteq\{1,\dots,N\}\setminus\{i\}\\|\mathcal{S}|=n}}{\arg\min} \{_{j\in\mathcal{S}}(V_s)_{ij}\}$
$o = [x_i, \{x_j\}_{j\in\mathcal{N}_i}]$
Update Neighbours
Decentralized Policy Execution
$V_d(t, o(t)) = \min_{z\in\mathbb{R}^+} z \quad$ s.t. $\hat{V}_\theta(t,o,z) \leq 0$
$\pi_\theta^i(t,x) = \arg\min_{\mathbf{u}_d} \langle \nabla \hat{V}_\theta(t,\hat{o}^*),\ \hat{f}_d(\hat{o}^*, \mathbf{u}_d)\rangle$
Policy Inference in Receding Horizon
Multi-Agent Interaction
Selected Neighbours
Trajectory

Fig. 1: We propose **MAD-PINN** – a framework for safe and optimal multi-agent control. MAD-PINN is divided into two phases: (1) **Centralized training**, where we learn the auxiliary epigraph-based value function $\hat{V}_\theta$ using a boundary-aware PINN, and the pairwise safety value function V_s via DeepReach; (2) **Decentralized deployment**, where each agent selects its safety-critical neighbours using V_s and executes the policy in a receding-horizon manner using $\hat{V}_\theta$. This design enables tractable training, adaptive neighbour selection, and scalable execution in large multi-agent systems.

tion to achieve both formal safety guarantees and scalability. Specifically, MAD-PINN uses a physics-informed neural network (PINN) to approximate the epigraph-based value function of the SC-OCP, with boundary conditions encoded to ensure strict satisfaction of terminal safety constraints. Scalability is achieved by training these value functions on reduced-agent systems and deploying them in a decentralized fashion, where each agent makes decisions using only local observations of its most safety-critical neighbours. To support reliable real-world execution, MAD-PINN integrates two additional components: an HJ reachability-based neighbour selection strategy to identify critical interactions, and a receding-horizon policy execution scheme to adapt online to dynamic agent interactions. Our core contributions are:

- **MAD-PINN framework:** We propose a decentralized, boundary-aware physics-informed learning framework for multi-agent state-constrained optimal control (SC-OCP), which integrates performance optimization with strict safety guarantees.
- **Safety-aware neighbour selection:** We introduce a reachability-based neighbour selection strategy that prioritizes the most safety-critical interactions, thereby preserving safety while avoiding unnecessary conservatism in decentralized execution.
- **Empirical validation:** We evaluate MAD-PINN on multi-agent navigation tasks at varying scales, showing that it achieves superior safety–performance trade-offs, maintains high scalability as the number of agents grows, and outperforms state-of-the-art baselines.

II. Problem Setup

Consider a *homogeneous* multi-agent system comprising N agents. The state and control input for each agent i are defined as $x_i \in \mathcal{X}_i \subseteq \mathbb{R}^D$ and $u_i \in \mathcal{U}_i \subseteq \mathbb{R}^M$, respectively. Each agent's motion is governed by nonlinear dynamics expressed as: $\dot{x}_i(t) = f_i(x_i(t), u_i(t))$, where $f_i : \mathbb{R}^D \times \mathbb{R}^M \to \mathbb{R}^D$ is a locally Lipschitz continuous function. Although our framework assumes f_i is known, we note that this function could also be learned from data if a model is not available.

Using this premise, we define the joint state and control input vectors for the entire system as $x := [x_1, \dots, x_N] \in \mathcal{X} \subseteq \mathbb{R}^{N\times D}$ and $u := [u_1, \dots, u_N] \in \mathcal{U} \subseteq \mathbb{R}^{N\times M}$. Consequently, the collective system dynamics are described by the concatenated function $f = [f_1, f_2, \dots, f_N]$, yielding the global dynamics model: $\dot{x}(t) = f(x(t), u(t))$.

The primary safety objective is to ensure collision avoidance between all agents. We formalize this by defining a failure set $\mathcal{F} \subseteq \mathcal{X}$ of unsafe configurations:

$$\mathcal{F} = \{x \in \mathcal{X} : \min_{i\neq j} d_{ij}(x) \leq r\}. \tag{1}$$

Here, d_{ij} is the Euclidean distance between agents i and j, and r is a predefined minimum safe separation distance. To integrate this safety requirement into the optimal control framework, we define a constraint function $g(x)$ such that $\mathcal{F} = \{x \in \mathcal{X} \mid g(x) > 0\}$. The system's performance is quantified by a cost functional $C(t, x(t), u(\cdot))$ that accumulates a running cost and a terminal cost over a time horizon:

$$C(t, x(t), u(\cdot)) = \int_{s=t}^{T} l(x(s))ds + \phi(x(T)), \tag{2}$$

where $l : \mathcal{X} \to \mathbb{R}_{\geq 0}$ and $\phi : \mathcal{X} \to \mathbb{R}_{\geq 0}$ are non-negative, Lipschitz continuous functions. The control input is $u(\cdot) : [t, T) \to \mathcal{U}$. C could represent the time taken to reach the goal or fuel consumption, for instance. Furthermore, we assume the cost functions are separable and identical across the homogeneous agents.

Our goal is to synthesize an optimal control policy $\pi^* : [t, T) \times \mathcal{X} \to \mathcal{U}$ that minimizes this cost while guaranteeing that the state trajectory never enters the failure set $\mathcal{F}$. This leads to the following **State-Constrained Optimal Control Problem (SC-OCP)** with the value function $V(t, x)$:

$$\begin{aligned} V(t, x(t)) &= \min_{u(\cdot)} \int_t^T l(x(s))ds + \phi(x(T)) \\ \text{s.t.} \quad \dot{x}(s) &= f(x(s), u(s)), \quad g(x(s)) \le 0 \quad \forall s \in [t, T] \end{aligned} \tag{3}$$

Thus, the policy, π^*, derived from the solution of this SC-OCP co-optimizes safety and performance.

A. Decentralized Multi-Agent SC-OCP

As a direct consequence of its formulation, the centralized SC-OCP in (3) suffers from the curse of dimensionality; its computational complexity grows intractably with the number of agents N [24]. To address this scalability challenge, we introduce a decentralized reformulation of the problem.

In the decentralized framework, agents are constrained by a limited observation field. Rather than observing the global state x, each agent i must rely on a local observation o_i. This observation is constructed from the agent's own state and the states of a fixed number n of neighbouring agents within its observation radius r_{obs}, where $n \le N$. Formally, the observation is constructed by an operator $\mathcal{O}_i$ that selects and concatenates these relevant states, i.e., $o_i = \mathcal{O}_i(x) \subseteq \mathbb{R}^{D\times(n+1)}$, where: $o_i = [x_i, \{x_j\}_{j\in\mathcal{N}_i}]$ and $\mathcal{N}_i$ denotes the set of indices corresponding to the neighbors of agent i.

A key enabling factor for this approach is the homogeneity of the agent dynamics and objectives. This property ensures that the decentralized value function is identical for all agents, irrespective of their specific identity. We can therefore drop the agent subscript i and represent a generic local observation simply as o. This allows us to define a single, shared decentralized optimal control problem (**Decentralized SC-OCP**):

$$\begin{aligned} V_d(t, o(t)) &= \min_{\mathbf{u}_d} \int_t^T l_d(o(s))\, ds + \phi_d(o(T)) \\ \text{s.t. } \dot{o}(s) &= f_d(o(s), u_d(s)), \; g_d(o(s)) \le 0 \quad \forall s \in [t, T] \end{aligned} \tag{4}$$

Here, l_d and ϕ_d represent the running and terminal cost functions, respectively, aggregated over the subset of agents within o. The control sequence $\mathbf{u}_d$ is the joint input for these agents. The dynamics function f_d is derived by concatenating the individual agent dynamics, and the constraint function g_d enforces safety (e.g., collision avoidance) among the agents in the local observation.

This reformulation directly addresses the scalability issue by fixing the observation size to $n + 1$, rendering the problem dimension independent of the total agent count N. Consequently, a single value function V_d and its associated policy, computed once, can be used across agents. This provides a computationally tractable and globally consistent solution to the original, large-scale SC-OCP in (3).

B. Epigraph Reformulation

Directly solving the State-Constrained Optimal Control Problem (SC-OCP) in (4) is challenging due to the presence of hard state constraints. To circumvent this challenge, we adopt an epigraph reformulation [25] that transforms the original constrained problem into a more tractable, equivalent two-stage optimization formulation. The core of this reformulation is the introduction of a non-negative auxiliary variable $z \in \mathbb{R}^+$, representing a bound on the cost-to-go. The original SC-OCP is then equivalently expressed as:

$$V_d(t, o(t)) = \min_{z\in\mathbb{R}^+} z \quad \text{s.t.} \quad \hat{V}_d(t, o, z) \le 0, \tag{5}$$

where $\hat{V}_d$ denotes a newly defined auxiliary value function. Following the framework of [23], this function is given by:

$$\hat{V}_d(t, o(t), z) = \min_{\mathbf{u}_d} \max \left\{ C(t, o(t), \mathbf{u}_d) - z, \; \max_{s\in[t,T]} g_d(o(s)) \right\}. \tag{6}$$

This formulation captures the problem's dual objectives: minimizing cost and ensuring safety. Crucially, the condition $\hat{V}_d(t, o, z) < 0$ guarantees that $g_d(o(s)) < 0$ for all $s \in [t, T]$, meaning the system's trajectory remains within the safe set.

The optimal solution z^* to (5) represents the *minimum admissible cost* achievable without violating the state constraints. This interpretation provides an intuitive safety-performance trade-off: a choice of $z > z^*$ results in an overly conservative policy, while $z < z^*$ prioritizes performance at the potential expense of safety.

To further enable the application of dynamic programming principles, we treat the auxiliary variable z as a state variable with the simple dynamics $\dot{z}(t) = -l(o(t))$. This signifies that the admissible cost bound z is depleted by the stage cost $l(o)$ along the system's trajectory. This yields the following augmented system dynamics:

$$\dot{\hat{o}} = \hat{f}_d(t, \hat{o}, u) := \begin{bmatrix} f_d(t, o, u) \\ -l(o) \end{bmatrix}, \tag{7}$$

where $\hat{o} := [o, z]^\top \in \mathcal{X} \times \mathbb{R}$ is the augmented state.

Under standard assumptions A1–A4 from [23], the auxiliary value function $\hat{V}_d(t, \hat{o})$ is the unique continuous viscosity solution to the Hamilton-Jacobi-Bellman (HJB) partial differential equation:

$$\min \left(-\partial_t \hat{V}_d - \min_{\mathbf{u}_d} \left\langle \nabla_{\hat{o}} \hat{V}_d, \; \hat{f}_d(\hat{o}, u) \right\rangle, \; \hat{V}_d - g_d(o) \right) = 0, \tag{8}$$

for all $t \in [0, T)$ and $\hat{o} \in \mathcal{X} \times \mathbb{R}$, with the terminal condition:

$$\hat{V}_d(T, \hat{o}) = \max\left(\phi_d(o) - z, \; g_d(o)\right). \tag{9}$$

This HJB characterization provides a formal basis for computing the value function and the resulting optimal safe control policy. For notational brevity, we have dropped the subscript d and will hereafter use $\hat{V}(t, \hat{o})$ to denote this decentralized auxiliary value function.

III. Methodology

The solution to the SC-OCP formulated in Equation (3) hinges on the computation of the optimal value function V_d that minimizes cost under safety constraints. Our approach to obtaining this function proceeds in two primary stages as illustrated in Figure 1: an offline learning phase and an online deployment phase. First, we learn the auxiliary value function $\hat{V}$ using a physics-informed machine learning framework. Then, V_d is obtained from $\hat{V}$ using (5). For online decentralized deployment, a safety-aware clustering strategy is employed to determine the appropriate neighbours for each agent. The control policy for each agent is then derived based on V_d. We now discuss each step in detail.

A. Training the Auxiliary Value Function ($\hat{V}$)

The auxiliary value function $\hat{V}$ is characterized by the HJB-PDE in (8) (Section II-B). Traditional numerical methods for solving such PDEs rely on discretizing the state space over a grid [26], [27]. While accurate for low-dimensional systems, these methods are susceptible to the curse of dimensionality, as their computational cost scales exponentially with the number of states. To overcome this limitation, we leverage a physics-informed neural network (PINN) framework that uses PDE residuals to learn the value function and has shown promising results in solving high-dimensional HJB PDEs [28].

Auxiliary Value Function Parameterization: We approximate the auxiliary value function $\hat{V}(t, \hat{o})$ using a neural network, with parameters θ. A critical requirement is that the solution must satisfy the terminal boundary condition to adhere to the problem's safety constraints. To enforce this *exactly*, we structure our network output as:

$$\hat{V}_\theta(t, \hat{o}) = \max\left(\phi_d(o) - z,\ g_d(o)\right) + (T - t) \cdot R_\theta(t, \hat{o}),$$

where the first term encodes the terminal condition in (9), and the neural network, denoted $R_\theta(t, \hat{o})$, learns the residual evolution of the value function over time. This formulation, inspired by [29], guarantees that $\hat{V}_\theta(T, \hat{o}) = \max\left(\phi_d(o) - z,\ g_d(o)\right)$ for any state $\hat{o}$, irrespective of the network's output R_θ. There are two key advantages to the proposed structure of $\hat{V}_\theta$: (a) it eliminates the need to explicitly learn a complex boundary condition by hard-coding it into the network's forward pass, and (b) it reduces the learning problem to minimizing a single HJB-derived loss function (as we discuss next), thereby removing the necessity for a manually-tuned loss weighting scheme.

Loss Function and Training Scheme: The parameters θ of the network R_θ are learned by minimizing a loss function that penalizes the HJB PDE residual errors. Specifically, the loss function to learn the NN parameters is:

$$\begin{aligned}
&\mathcal{L}\left(t_k, \hat{o}_k|\theta\right) = \mathcal{L}_{pde}\left(t_k, \hat{o}_k|\theta\right) \\
&= \|\min\left\{-\partial_t \hat{V}_\theta\left(t_k, \hat{o}_k\right) - H(t_k, \hat{o}_k),\ \hat{V}_\theta\left(t_k, \hat{o}_k\right) - g_d\left(\hat{o}_k\right)\right\}\|, \\
&= \|\min\left\{(t_k - T)\partial_t R_\theta\left(\hat{o}_k, t_k\right) + R_\theta\left(\hat{o}_k, t_k\right) - H\left(\hat{o}_k, t_k\right),\right. \\
&\qquad \left. V_\theta\left(\hat{o}_k, t_k\right) - g_d\left(\hat{o}_k\right)\right\}\|,
\end{aligned} \tag{10}$$

where, $H(t, \hat{x}) = \min_{u \in \mathcal{U}} \langle \nabla V_\theta(\hat{o}_i, t), \hat{f}_d(\hat{o}_i, u)\rangle$. Typically, PINNs incorporate an additional loss term to enforce boundary conditions. In contrast, due to the structure of our formulation of $\hat{V}_\theta$, the boundary conditions are satisfied exactly. Consequently, the optimization reduces to minimizing a single loss term $\mathcal{L}_{pde}$, eliminating the need for auxiliary loss terms and the hyperparameters required for their weighting.

Curriculum Training: A key challenge in solving the HJB-PDE is its backward-in-time evolution; the solution at time t depends on the future time $t+\Delta t$. To manage this complexity during training, we employ a curriculum learning strategy similar to DeepReach [28]. Training begins by sampling time points near the terminal time T and progressively expanding the interval backward until it covers the entire horizon $[0, T]$, whereas the states are sampled uniformly across the state space at each training iteration. This allows the network to first learn the well-defined terminal condition accurately before learning to propagate the solution backward in time, governed by the PDE dynamics, yielding the auxiliary value function $\hat{V}_\theta$. We refer interested readers to [28], [29] for more details on the curriculum training scheme.

B. Neighbour Selection Strategy

During online deployment, each agent must efficiently identify the subset of neighbours most relevant for maintaining safety. Rather than treating all nearby agents equally, we group agents into clusters defined by their potential for safety-critical interactions. The cluster size is chosen to match the neighbourhood dimension used when training the decentralized auxiliary value function, ensuring consistency between training and deployment.

To determine which neighbours pose the greatest risk, we employ a principled criterion derived from Hamilton–Jacobi (HJ) reachability analysis. HJ reachability provides a rigorous way to quantify the likelihood of safety conflicts, enabling each agent to prioritize interactions that most directly impact safe operation. Specifically, while $\hat{V}$ and V_d encode cost and safety constraints in the SC-OCP, we additionally compute a pairwise safety value function V_s using HJ reachability to guide neighbour selection.

HJ Reachability [30], [31] characterizes the set of states from which the system can be driven into a failure set. Let $\psi : \mathbb{R}^n \to \mathbb{R}$ be a Lipschitz-continuous function whose sub-zero level set $\Psi = \{x : \psi(x) \leq 0\}$ represents pairwise collision states between the two agents. The corresponding safety value function is given by:

$$V_s(x, t) = \sup_{u(\cdot)} \min_{\tau \in [t, T]} \psi(\xi^u_{x,t}(\tau)), \tag{11}$$

where $\xi^u_{x,t}(\cdot)$ is the system trajectory. Intuitively, $V_s(x, t)$ measures how close the two agents are to a collision, even under optimal control. The unsafe set is precisely the sub-zero level set of V_s. The value function satisfies the

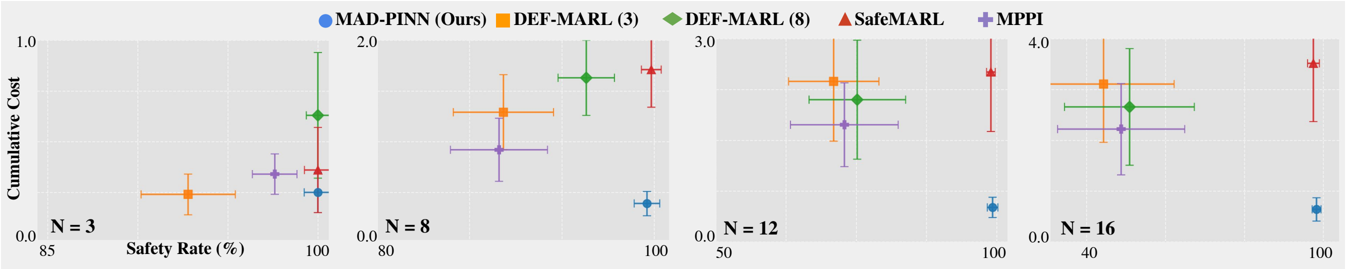

Fig. 2: Comparison of cumulative costs and safety rates across testing environments with 3, 8, 12, and 16 agents. Our method consistently appears in the bottom-right region, indicating **superior safety-performance co-optimization relative to all baselines**. Moreover, its degradation in performance and safety with increasing agent count is minimal, **demonstrating better scalability compared to the baselines**.

Hamilton-Jacobi-Bellman Variational Inequality (HJB-VI):

$$\begin{gathered}\min\{\partial_t V_s(x,t) + H_s(x,t), \psi(x) - V_s(x,t)\} = 0, \\ V_s(x,T) = \psi(x), \\ H_s(x,t) = \max_{u\in\mathcal{U}} \langle \nabla V_s(x,t)\ ,\ f_s(x,u)\rangle,\end{gathered} \tag{12}$$

where H_s is the Hamiltonian corresponding to the safety value function and f_s encodes the pairwise dynamics between agents.

Learning the Pairwise Safety Value Function (V_s): We approximate V_s using DeepReach [28], [29]. To characterize pairwise safety interactions, we consider each agent i in relation to another agent j. The target function is defined as $\psi = d_{ij} - r$, where d_{ij} denotes the distance between the two agents and r is the prescribed collision radius. By design, $\psi \leq 0$ corresponds to states in which agents i and j are in collision, thereby providing a natural safety signal. Training in this manner yields a value function that quantifies the relative safety risk posed by one agent to another.

Neighbour Selection: For agent i, the value $(V_s)_{ij}$ represents the degree of risk posed by agent j. Smaller values correspond to higher collision likelihoods. Specifically, if $(V_s)_{ij} < (V_s)_{ik}$, then agent j poses a higher safety risk to agent i compared to agent k, and thus should be prioritized in i's decision-making process. Hence, to select its n neighbours, agent i computes $(V_s)_{ij}$ for all agents j within its observation radius r_{obs}, ranks them, and selects the n agents with the lowest values. This process ensures that each agent focuses its decision-making on the most safety-critical interactions, while still keeping the neighbourhood size fixed for computational tractability.

C. Policy Synthesis

Once the neighbour set is determined, each agent synthesizes its policy by solving the optimization problem in (5). To enforce safety, we set $V_d(t,x) = +\infty$ whenever $\hat{V}_\theta(t,x,z) > 0$, since such states are unsafe and violate the safety constraint. For the remaining states, the optimization is solved via binary search over z. The resulting state-feedback policy for agent i, $\pi^i_\theta : \mathcal{X} \times [t,T] \to \mathcal{U}$, is given by

$$\pi^i_\theta(t,x) = \arg\min_{\mathbf{u}_d} \langle \nabla \hat{V}_\theta(t,\hat{o}^*),\ \hat{f}_d(\hat{o}^*, \mathbf{u}_d)\rangle,$$

where $\hat{o}^* = [o, z^*]^T$ is the augmented state corresponding to the optimal z^*.

Since interaction structures in multi-agent navigation evolve dynamically, the neighbour set $\mathcal{N}_i$ is updated online, motivating a receding-horizon execution of the policy. This ensures that policy continuously adapts by incorporating updated interaction information and anticipating future conflicts. In addition, frequent re-planning further enhances robustness to model mismatch, sensor noise, and external disturbances, enabling reliable decentralized navigation. Finally, this framework naturally extends to long-horizon tasks by repeatedly solving the SC-OCP over shorter horizons, allowing the proposed framework to co-optimize safety and performance in a practical and computationally efficient manner for real-world autonomous systems.

IV. Experiments

The goal of our experiments is to assess the effectiveness of the proposed framework in (i) co-optimizing safety and performance in multi-agent systems, (ii) scaling to larger environments with higher numbers of agents, and (iii) validating the proposed safety-aware neighbour selection strategy.

A. Baselines

To thoroughly assess our method, we benchmark it against baselines that represent the spectrum of safety integration techniques: 1) *Constrained Policy Synthesis:* **DEF-MARL** [32], which leverages multi-agent RL (MARL) to solve the discrete-time epigraph reformulation of the SC-OCP to synthesize safe and optimal policies; 2) *Safety Filtering:* **SafeMARL** [13], which uses a control barrier value function (CBVF)-based safety filter to provide safety for a nominal MARL policy; and 3) *Soft-Constrained Optimization:* **MPPI** [17], a sampling-based MPC method to solve (4) that penalizes constraint violations in its cost function using a Lagrangian approach.

B. Evaluation Metrics

To evaluate the trade-off between performance and safety, we use the following metrics: **(1) Cumulative Cost:** The total cost $\int_0^T l(x(s))ds + \varphi(x(T))$ accumulated along safe trajectories. **(2) Safety Rate:** The fraction of agents that remain collision-free for the entire horizon, quantifying per-agent safety. **(3) Safe Scenarios:** The fraction of scenarios in

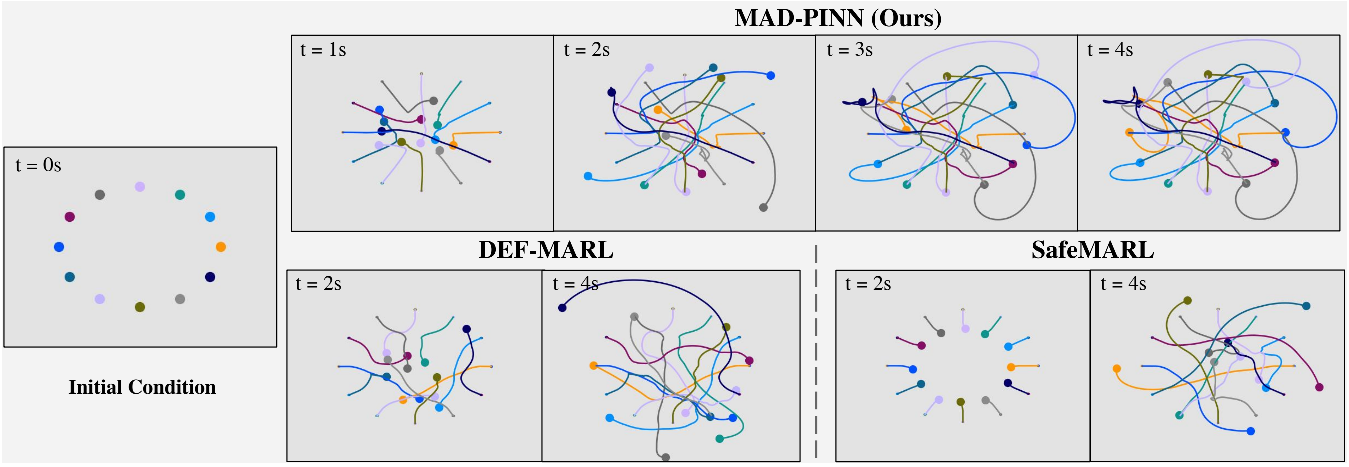

Fig. 3: Snapshots of multi-agent navigation trajectories at different times using MAD-PINN and baselines. Agents are represented as circles with radius R, indicating the minimum safe distance they must maintain from each other. Smaller dots mark their respective goals. MAD-PINN trajectories show that agents proactively **maintain long-horizon safety** by adjusting their paths to avoid close encounters, rather than enforcing safety reactively (SafeMARL), which could lead to suboptimal behaviours.

	3 Agents		8 Agents		12 Agents		16 Agents	
Method	Safety	Safe Sc.	Safety	Safe Sc.	Safety	Safe Sc.	Safety	Safe Sc.
Ours	100% ± 0.0%	100% ± 0.0%	99.5% ± 0.4%	98% ± 1.2%	99.3% ± 0.6%	96% ± 1.8%	98.3% ± 0.8%	86% ± 2.7%
DEF-MARL (3)	94% ± 2.8%	91% ± 3.4%	89.8% ± 2.5%	63% ± 4.2%	71% ± 3.8%	31% ± 4.8%	44.3% ± 3.4%	7% ± 1.5%
DEF-MARL (8)	100% ± 0.0%	100% ± 0.0%	95.4% ± 2.1%	82% ± 3.2%	75.2% ± 3.5%	37% ± 4.5%	50.9% ± 2.9%	13% ± 2.1%
SafeMARL	100% ± 0.0%	100% ± 0.0%	99.8% ± 0.2%	99% ± 0.8%	99% ± 0.5%	96% ± 1.7%	97.5% ± 1.0%	90% ± 2.2%
MPPI	98% ± 1.1%	97% ± 1.4%	89.5% ± 2.8%	72% ± 3.5%	72.9% ± 3.7%	35% ± 4.9%	48.8% ± 3.6%	9% ± 2.4%

TABLE I: Average safety rates and safe-scenario percentages across varying numbers of agents. Our method maintains consistently high values, indicating collision-free execution of every agent for nearly all initial configurations and demonstrating **effective safety-performance co-optimization at the agent and system level**.

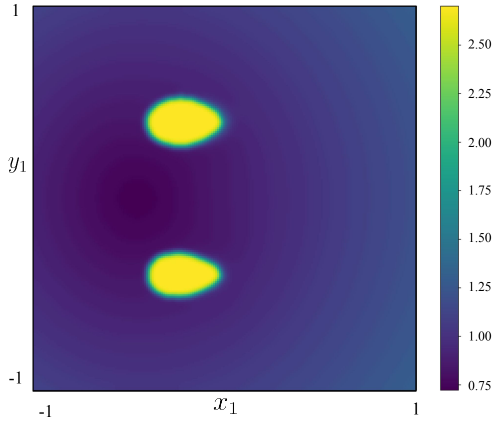

Fig. 4: **Heatmap of the learned value function with respect to the ego agent's position coordinates.** The other two agents are at $[-0.3,\ 0.4]$ and $[-0.3,\ -0.4]$, both moving with velocity $[-1,\ 0]$, while the ego agent moves with velocity $[1,\ 0]$ toward its goal at $[-0.5,\ 0]$.

which all agents remain collision-free for the entire horizon, reflecting collective safety guarantees.

C. Experimental Setup

We study a multi-agent **Drone Navigation** problem where each agent follows double-integrator dynamics with state $\mathbf{s}_i = [x, y, v_x, v_y]^\top$, constrained to $(x, y) \in [-1, 1]^2$ and $(v_x, v_y) \in [-4, 4]^2$. The control input is $\mathbf{u}_i = [a_x, a_y]^\top \in [-4, 4]^2$, representing acceleration. Each agent is assigned a parameterized goal (x_g, y_g), and its dynamics are given by $\dot{x} = v_x$, $\dot{y} = v_y$, $\dot{v}_x = a_x$, $\dot{v}_y = a_y$. Agents must reach their goals while maintaining a safety distance $r = 0.1$. The running cost for training the decentralized value function is:

$$l_d = \sum_{j=1}^{|\mathcal{N}_i|} \|(x^j(t), y^j(t))^\top - (x_g^j(t), y_g^j(t))^\top\|, \qquad (13)$$

where $\mathcal{N}_i$ denotes the neighbor set. We train the auxiliary value function with 3 agents (so $|\mathcal{N}_i| = 2$), a time horizon of 0.2s, and an observation radius $r_{obs} = 0.5$, and deploy the same value function for all agents across all environments. The residual component of the auxiliary value function R_θ is approximated as a multi-layer perceptron (MLP) with three hidden layers of 256 neurons each, using sine activation functions. The network is trained with the Adam optimizer at a learning rate of 2×10^{-5}. We evaluate all algorithms over 100 distinct initial conditions across 5 seeds.

Figure 4 illustrates the heatmap of the learned value function as a function of the ego agent's position coordinates. Regions of high value (yellow) correspond to unsafe states overlapping with the positions of other agents, thus encoding

Number of Agents	Cumulative Cost	Safety Rate	Safe Scenarios
64	1.09	100%	100%
128	1.68	98.75%	95%
256	2.73	96.25%	85%

TABLE II: Effect of increasing agent count and environment size. Our method **co-optimizes safety and performance even in large multi-agent environments**, despite training on smaller environments with fewer agents.

Method	Cumulative Cost	Safety Rate	Safe Scenarios
Value-based (Ours)	0.51	99.33%	96%
Nearest	0.82	83%	45%
Random	1.55	33%	4%

TABLE III: Ablation study on the impact of the proposed neighbor selection strategy. By effectively **capturing safety-critical inter-agent interactions among neighbors**, it enables improved safety–performance co-optimization.

safety constraints. Conversely, regions of low value (dark purple) are concentrated around the goal, reflecting the task objective. This demonstrates that the value function integrates both safety and performance considerations, enabling the derived policy to effectively co-optimize these objectives.

D. Results

1) Effectiveness in Co-optimizing Safety and Performance: To gauge the ability of our method in co-optimizing safety and performance, we compare it with the baselines on 3, 8, 12, and 16 agents, with environment size fixed so that density increases as the number of agents grows. As shown in Figure 2, our method consistently achieves the best trade-off, attaining the lowest cumulative cost while maintaining near-perfect safety rates, thereby validating its ability to co-optimize both objectives. In contrast, DEF-MARL, trained with 3 and 8 agents, performs competitively when evaluated on the same agent counts but suffers pronounced degradation in both safety and performance as agent density increases, highlighting the limited generalization of RL-based co-optimization approaches. Safety-filtering baselines such as SafeMARL effectively prevent collisions, but their overly conservative behavior results in substantially higher costs. Penalty-based methods such as MPPI fare even worse, exhibiting both high costs and poor safety rates due to the soft enforcement of safety constraints. Table I further illustrates these differences through the stringent *safe scenarios* metric, which requires collision-free execution across all agents for the entire horizon. Our method consistently achieves a very high percentage of safe scenarios, indicating not only strong safety rates but also collision-free execution in nearly all cases. This highlights the effectiveness of decentralization, where each agent can co-optimize safety and performance individually. By contrast, non-safety filtering baselines such as DEF-MARL and MPPI, while achieving moderate safety rates, perform poorly on the safe scenarios metric as the number of agents grows, suggesting their inability to co-optimize safety and performance at the agent level. SafeMARL, while maintaining high safe-scenario rates, does so at the cost of severely degraded performance. Taken together, these results show that our approach inherits the strengths of safety-filtering methods (high safety) and performance-driven methods (low cost), while avoiding their drawbacks. Moreover, the performance degradation of our approach remains minimal when scaling from 3 to 16 agents despite the increase in density, demonstrating that it can effectively co-optimize safety and performance using only local observations; a property that makes it well-suited for real-world autonomous systems, where agents typically have access only to local state information. Figure 3 further substantiates our claims by demonstrating that the proposed method guarantees long-horizon safety while allowing all agents to successfully reach their respective goals without collisions. In contrast, baseline approaches either adopt reactive safety strategies (leading to over-conservatism) or fail to preserve safety, highlighting their limitations in co-optimizing safety and performance.

2) Scalability with Agent Count and Environment Size: To evaluate the scalability of our approach, we conduct experiments with 20 distinct initial conditions and a substantially larger number of agents, namely 64, 128, and 256. In addition, the environment size is increased from $[-1, 1]^2$ to $[-4, 4]^2$ to test the method in a more extensive and populated setting compared to the training environment. From Table II, it can be observed that our approach consistently achieves very high safety rates even in large environments with up to 256 agents, while maintaining a high safe-scenario percentage. This indicates that each agent is able to co-optimize safety and performance effectively, even as the number of agents increases. It is also important to note that the safety rates and safe-scenario percentages remain comparable across Tables I and II because the increased environment size in this study results in a similar agent density in both setups. The cumulative costs are higher in this setting, as agents must travel longer distances to reach their respective goals due to the environment being scaled by a factor of 4 in each dimension. However, the increase in cost is approximately proportional to the increase in environment dimensions, suggesting that our method preserves the same level of performance as in Table I. These findings further confirm that, within a decentralized setting, our method can effectively address extremely large-scale multi-agent problems by training policies using only local information in a smaller environment and subsequently applying the same policy independently to each agent. This highlights the practicality of our framework for deployment in large-scale multi-agent systems.

3) Effectiveness of Neighbour Selection: Finally, we conduct an ablation study to evaluate the effect of the proposed HJ Reachability-based neighbour selection strategy. We compare it against two alternative strategies: (1) a distance-based strategy, where each agent selects its two nearest neighbours, and (2) a random strategy, where each agent selects two neighbours uniformly at random within its observation radius. All strategies are evaluated on the same

12-agent environment used in Table I. As shown in Table III, our method consistently achieves higher safety rates and lower costs, indicating that prioritizing neighbours based on the pairwise safety value functions V_s effectively captures the most critical interactions. In contrast, the distance-based strategy leads to a myopic neighbour selection, as it ignores safety-critical information (e.g., agents' relative velocities), thereby reducing safety rates and drastically lowering the safe scenario fraction. The random selection strategy performs the worst, as it fails to incorporate any information about the surrounding agents. These results highlight the necessity of a principled neighbour selection framework that accounts for critical inter-agent interaction information to ensure reliable decentralized multi-agent navigation.

V. Conclusion and Future Work

We presented a physics-informed machine learning framework for scalable multi-agent safe and optimal control. The key contribution lies in the reformulation of the large-scale SC-OCP into a decentralized formulation with fixed observation size, enabling tractability while preserving global consistency. To address the curse of dimensionality in solving the associated HJB-PDE, we introduced a principled neighbour selection strategy based on reachability analysis, ensuring that agents account for the most safety-critical interactions. Extensive experiments demonstrated that our approach effectively co-optimizes safety and performance, scales favourably with the number of agents, and significantly outperforms existing performance-driven and safety-driven baselines.

Future work will focus on extending the framework to heterogeneous agent systems, incorporating dynamics and model uncertainty, and evaluating performance on real-world robotic platforms. We believe this work represents a step toward practical, safe, and scalable multi-agent autonomy. In addition, we plan to quantify the approximation error of the auxiliary value function [21], [24], [33] and investigate its impact on both safety and performance.

References

[1] E. Soria, F. Schiano, and D. Floreano, "Predictive control of aerial swarms in cluttered environments," *Nature Machine Intelligence*, vol. 3, no. 6, pp. 545–554, 2021.

[2] R. Donatus, K. Ter, O.-O. Ajayi, and D. Udekwe, "Multi-agent reinforcement learning in intelligent transportation systems: A comprehensive survey," 2025. [Online]. Available: https://arxiv.org/abs/2508.20315

[3] A. Kattepur, H. K. Rath, A. Simha, and A. Mukherjee, "Distributed optimization in multi-agent robotics for industry 4.0 warehouses," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing*, 2018, pp. 808–815.

[4] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," *Advances in neural information processing systems*, vol. 30, 2017.

[5] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *Advances in neural information processing systems*, vol. 35, pp. 24 611–24 624, 2022.

[6] S. Nayak, K. Choi, W. Ding, S. Dolan, K. Gopalakrishnan, and H. Balakrishnan, "Scalable multi-agent reinforcement learning through intelligent information aggregation," in *International Conference on Machine Learning*. PMLR, 2023, pp. 25 817–25 833.

[7] E. Altman, *Constrained Markov Decision Processes*, ser. Stochastic Modeling Series. Taylor & Francis, 1999. [Online]. Available: https://books.google.co.in/books?id=3X9S1NM2iOgC

[8] S. Gu, J. G. Kuba, Y. Chen, Y. Du, L. Yang, A. Knoll, and Y. Yang, "Safe multi-agent reinforcement learning for multi-robot control," *Artificial Intelligence*, vol. 319, p. 103905, 2023.

[9] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[10] J. Borquez, K. Chakraborty, H. Wang, and S. Bansal, "On safety and liveness filtering using hamilton–jacobi reachability analysis," *IEEE Transactions on Robotics*, vol. 40, pp. 4235–4251, 2024.

[11] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.

[12] K.-C. Hsu, H. Hu, and J. F. Fisac, "The safety filter: A unified view of safety-critical control in autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, no. Volume 7, 2024, pp. 47–72, 2024. [Online]. Available: https://www.annualreviews.org/content/journals/10.1146/annurev-control-071723-102940

[13] J. J. Choi, J. J. Aloor, J. Li, M. G. Mendoza, H. Balakrishnan, and C. J. Tomlin, "Resolving conflicting constraints in multi-agent reinforcement learning with layered safety," in *Proceedings of Robotics: Science and Systems*, Los Angeles, USA, June 2025. [Online]. Available: https://arxiv.org/abs/2505.02293

[14] M. Tayal, H. Zhang, P. Jagtap, A. Clark, and S. Kolathaya, "Learning a formally verified control barrier function in stochastic environment," in *Conference on Decision and Control (CDC)*. IEEE, 2024.

[15] C. E. García, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0005109889900022

[16] L. Grüne, J. Pannek, L. Grüne, and J. Pannek, *Nonlinear model predictive control*. Springer, 2017.

[17] G. Williams, P. Drews, B. Goldfain, J. M. Rehg, and E. A. Theodorou, "Information-theoretic model predictive control: Theory and applications to autonomous driving," *IEEE Transactions on Robotics*, vol. 34, no. 6, pp. 1603–1622, 2018.

[18] L. Streichenberg, E. Trevisan, J. J. Chung, R. Siegwart, and J. Alonso-Mora, "Multi-agent path integral control for interaction-aware motion planning in urban canals," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1379–1385.

[19] ——, "Multi-agent path integral control for interaction-aware motion planning in urban canals," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 1379–1385.

[20] M. Tayal, A. Singh, P. Jagtap, and S. Kolathaya, "Semi-supervised safe visuomotor policy synthesis using barrier certificates," *arXiv preprint arXiv:2409.12616*, 2024.

[21] ——, "Cp-ncbf: A conformal prediction-based approach to synthesize verified neural control barrier functions," *arXiv preprint arXiv:2503.17395*, 2025.

[22] H. M. Soner, "Optimal control with state-space constraint i," *SIAM Journal on Control and Optimization*, vol. 24, no. 3, pp. 552–561, 1986. [Online]. Available: https://doi.org/10.1137/0324032

[23] A. Altarovici, O. Bokanowski, and H. Zidani, "A general hamilton-jacobi framework for non-linear state-constrained control problems," *ESAIM: Control, Optimisation and Calculus of Variations*, vol. 19, no. 2, pp. 337–357, 2013.

[24] M. Tayal, A. Singh, S. Kolathaya, and S. Bansal, "A physics-informed machine learning framework for safe and optimal control of autonomous systems," in *Forty-second International Conference on Machine Learning*, 2025.

[25] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[26] I. Mitchell, "A toolbox of level set methods," *http://www. cs. ubc. ca/mitchell/ToolboxLS/toolboxLS.pdf*, 2004.

[27] E. Schmerling, "hj_reachability: Hamilton-Jacobi reachability analysis in JAX," *https://github.com/StanfordASL/hj_reachability*, 2021.

[28] S. Bansal and C. J. Tomlin, "Deepreach: A deep learning approach to high-dimensional reachability," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1817–1824.

[29] A. Singh, Z. Feng, and S. Bansal, "Exact imposition of safety boundary conditions in neural reachable tubes," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*, 2025. [Online]. Available: https://arxiv.org/abs/2404.00814

[30] I. M. Mitchell, A. M. Bayen, and C. J. Tomlin, "A time-dependent hamilton-jacobi formulation of reachable sets for continuous dynamic games," *IEEE Transactions on automatic control*, vol. 50, no. 7, pp. 947–957, 2005.

[31] J. Lygeros, "On reachability and minimum cost optimal control," *Automatica*, vol. 40, no. 6, pp. 917–927, 2004.

[32] S. Zhang, O. So, M. Black, Z. Serlin, and C. Fan, "Solving multi-agent safe optimal control with distributed epigraph form MARL," in *Proceedings of Robotics: Science and Systems*, 2025.

[33] A. Lin and S. Bansal, "Verification of neural reachable tubes via scenario optimization and conformal prediction," in *Proceedings of the 6th Annual Learning for Dynamics & Control Conference*, ser. Proceedings of Machine Learning Research, A. Abate, M. Cannon, K. Margellos, and A. Papachristodoulou, Eds., vol. 242. PMLR, 15–17 Jul 2024, pp. 719–731. [Online]. Available: https://proceedings.mlr.press/v242/lin24a.html