2024 IEEE 63rd Conference on Decision and Control (CDC)
December 16-19, 2024. MiCo, Milan, Italy

2024 IEEE 63rd Conference on Decision and Control (CDC) | 979-8-3503-1633-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/CDC56724.2024.10886205

Learning Differentiable and Safe Multi-Robot Control for Generalization to Novel Environments using Control Barrier Functions

Vivek Sharma*, Negar Mehr** and Naira Hovakimyan*

***Abstract*— Ensuring safety in the navigation of multi-robot systems using control barrier functions has traditionally involved the utilization of a *pre-tuned* class-$\mathcal{K}$ function specifically tailored to a given environment. However, these pre-tuned class-$\mathcal{K}$ functions struggle to generalize to different environments. In this work, we address these challenges for control-affine systems with actuation constraints. Our key insight is that incorporating environment-specific information implicitly into the class-$\mathcal{K}$ function can enhance generalization to unseen environments. We introduce a parameterization of the class-$\mathcal{K}$ functions for multi-robot systems using a Graph Neural Network (GNN). We formulate safety conditions and safe control using control barrier functions utilizing this GNN-based class-$\mathcal{K}$ function, which is optimized with both environmental information and information perceived by the robot in its local neighborhood leading to decentralized execution. To enable end-to-end learning of class-$\mathcal{K}$ functions and decentralized control policy, we employ a differentiable optimization layer, facilitating the embedding of optimization problem for computing safe control policies jointly with class-$\mathcal{K}$ functions using environment information and information perceived by the robot in its local neighborhood. We show through simulation results the effectiveness of our proposed method in generating scalable and generalizable safe control policies which are adaptable to novel environments.**

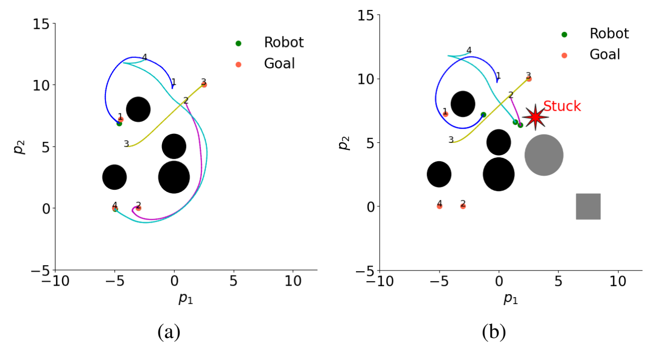


Fig. 1: Motivating example demonstrating the generalization challenge of CBF-based multi-robot safe control to novel environments using unicycle dynamics. The heuristically tuned class-$\mathcal{K}$ function optimized for Environment (a) with four obstacles (in black) is used in Environment (b) featuring two additional novel obstacles (in grey). As can be seen, the pre-tuned class-$\mathcal{K}$ function fails to generalize in Environment (b) with robots 2 and 4 getting stuck and struggling to advance to their goals safely.

I. INTRODUCTION

While multi-robot systems (MRS) are great for boosting task efficiency in contrast to single-robot systems [1], their use is limited in safety-critical scenarios [2] due to their lack of safety guarantees.

For single-robot systems, control barrier functions (CBF) [3] have been used for synthesizing control policies that can guaranteeing safety. CBF acts as a safety filter for unsafe actions and guarantees the safety of the system through forward invariance of the safe set. CBF conditions for set invariance are employed as constraints within a quadratic program (QP) [4] to compute a safe control action by modifying a high-level controller [5] that achieves goal-reaching objectives in a minimally invasive manner. Formulating such conditions often entails choosing a unique class-$\mathcal{K}$ function that maximizes the overall control performance for a specific environment [6].

For MRS, the number of CBF constraints increases linearly with the number of robots and obstacles in an environment. A safe set for each robot with respect to other robots and obstacles, within the CBF necessitates a distinct class-$\mathcal{K}$ function, tailored to maximize the overall control performance within a specific environment [6]. In real-world applications, especially those involving safety-critical tasks, prior knowledge of the environment is often lacking. Additionally, robots may encounter various environmental conditions during their deployment. Consequently, pre-tuning a class-$\mathcal{K}$ function for each possible environment to reconcile performance and safety for each robot becomes a challenging endeavor. See motivating example in Fig. 1

Related Work. We present a non-exhaustive list of related works pertinent to learning based safe multi-robot control. For a more comprehensive review, readers are referred to [7].

CBF based methods. Despite their safety guarantees, finding a CBF a-priori is a challenging task, which prompted the exploration of learning-based frameworks for jointly learning CBF certificates and control policies [8] for single-robot systems. Such learning-based frameworks for single-robot systems can be broadly categorized into model-based [9]–[11] and model-free approaches [12]–[14]. Typically, all such frameworks rely on a heuristically selected class-$\mathcal{K}$ function tailored to a specific environment. Consequently, these methods often exhibit limited generalizability and suffer from performance degradation when the environment undergoes changes. To address these challenges, [6], [15] recently introduced a technique that optimizes the class-

*V. Sharma and N. Hovakimyan are with the Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Champaign, IL 61820 USA (e-mail: `viveks4@illinois.edu`)
**N. Mehr is with the Department of Mechanical Engineering, University of California Berkeley, Berkeley, CA 94720-1740, USA

function using environmental data, aiming to enhance the generalizability of control barrier certificates to novel environments, but for single-robot systems.

Extending such learning frameworks from single-robot systems to MRS presents challenges. Recent approaches to certificate-based learning for MRS involves the parameterization of CBF and policies using feedforward neural networks [16], as well as GNN [17]. While these frameworks excel in decentralization and scalability of safe control policies, a limitation arises due to the inherent conservatism [6], [15] of the classical CBF conditions, primarily stemming from the use of a fixed class- function in enforcing these conditions using data collected from different environments. The performance of the control policy can be degraded when environmental conditions change. To address feasibility issues in computing control with a fixed class- function, [18] proposed a model-free reinforcement learning method using adaptive policies parameterized by a GNN that dynamically adjusts class- functions online, leveraging locally perceived information. Unlike our approach, this work focuses on the guaranteed feasibility of computing a safe control action. Moreover, our proposed framework is end-to-end trainable. Other notable works [19]–[21] employ GNN to learn complex multi-robot behaviors for safe motion planning in multi-robot scenarios, albeit without directly incorporating environmental information into the learning process.

Statement of Contributions. For MRS with control affine dynamics subject to control input constraints, given a CBF (or safety constraints), our contributions include:

1) We introduce a novel parameterization of the class- functions using GNN, aiming to learn inter-robot interactions for improved generalizability of the CBF to changing environments while retaining the set-invariant guarantees associated with classic CBF conditions.
2) We present a method for the end-to-end learning of decentralized safe control policies, achieved jointly with the optimization of the GNN-based class- function using differentiable-optimization layers with environmental and local neighborhood information perceived by the robot.

To the best of our knowledge, *this is the first work for MRS that considers optimizing class- function in an end-to-end manner using environmental information for improving the generalizability of a given CBF candidate to novel environments.* The overall framework is shown in Fig. 2.

II. Preliminaries

Consider a multi-robot system with N robots denoted by $= A_i \, {}_{i=1}^{N}$ in an environment with M static obstacles denoted by $O_l \, {}_{l=1}^{M}$. Assume that each robot has a control-affine dynamics[1] of the form

$$\dot{x}_i = f(x_i) + g(x_i)u_i, \tag{1}$$

[1]While in this work we consider robots with same dynamics, it is possible to consider heterogeneous MRS where the dynamics of robots are different.

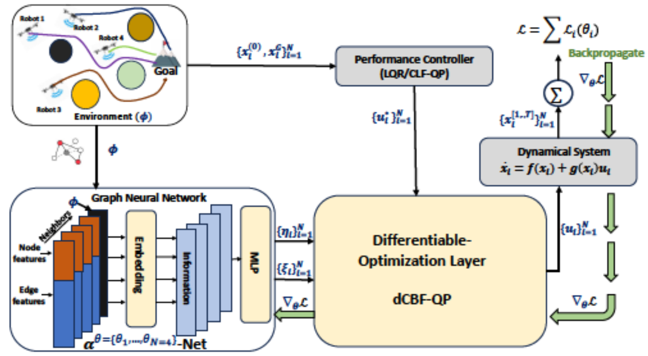


Fig. 2: Overview of our proposed framework. Our framework has three key components: a performance controller, GNN based class- function α^θ-Net, and a differentiable optimization layer CBF-QP (**dCBF-QP**) for end-to-end learning.

where $x_i \quad {}_i \quad \mathbb{R}^n$ is the state and $u_i \quad {}_i \quad \mathbb{R}^m$ is the control input of robot A_i. The vector fields f : $\mathbb{R}^n \quad \mathbb{R}^n$ and $g : \mathbb{R}^n \quad \mathbb{R}^m$ are assumed to be locally Lipschitz continuous. We omit the time dependence of state and control input in (1) for brevity. We define node set $= \; 1, 2, 3, ..., N$ as the set containing the index of robots. We use $x_i^{(t)}$ to denote the state of robot A_i for i at time t. Each robot A_i has a sensing radius $\sigma \quad \mathbb{R}^+$ that provides partial observation of the environment. We assume that each robot gets to observe the states of the other robots $x_j \; {}_{j \in \mathcal{N}_i}$ and the positions of the obstacles $p_{l,o} \; {}_{l \in \mathcal{N}_i}$ within the neighborhood of radius σ. Here, ${}_i$ is the neighbor set of A_i defined as ${}_i := \; j \quad p_j - p_i \quad \sigma, \quad j = i \bigcup l \;\; p_{l,o} - p_i \quad \sigma, \quad l = 1, .., M$, and p_i and $p_{l,o}$ refers to the position states of robot A_i and the obstacles O_l respectively. In this work, we will consider decentralized control policies similar to [18] of the form

$$\pi_i(u_i \quad x_i, \;\; x_j \;\; {}_{j \in \mathcal{N}_i}, \;\; p_{l,o} \;\; {}_{l \in \mathcal{N}_i}, \phi) \;\; i \tag{2}$$

where $\phi \quad \mathbb{R}^\mu$ is a vector with implicitly encoded environment information. We assume just like in prior works [6], [15] that information vector ϕ depends on the environment (e.g., size/shape of obstacles, velocity of dynamic obstacles, etc.) and is sampled from a distribution . In this work, we assume that start and goal positions are known to every robot and focus on designing control policy $\pi_i \; {}_{i=1}^{N}$ that drive robots from the initial states $x_i^{(0)} \; {}_{i=1}^{N}$ to goal states $x_i^G \; {}_{i=1}^{N}$ safely with local neighborhood information $\gamma_i := \;\; x_j \;\; {}_{j \in \mathcal{N}_i}, \;\; p_{l,o} \;\; {}_{l \in \mathcal{N}_i}$ in different environments.

We begin by reviewing the concept of CBF commonly employed in the literature for addressing safety requirements [22].

Definition 1: [23] For each robot A_i, consider a continuously differentiable function $h_i : \; {}_i \quad \mathbb{R}$ and a safe set ${}_i$ defined as superlevel set of h_i i.e. ${}_i := \; x_i \quad {}_i : h_i(x_i)$ 0 . Then h_i is a CBF if there exists a class- [24]function α such that for the control-affine system (1), we have

$$\sup_{u_i \in \mathcal{U}_i} L_f h_i(x_i) + L_g h_i(x_i) u_i \quad -\alpha(h_i(x_i)), \tag{3}$$

where ( ) := —— ( ) and ( ) := —— ( ). A quadratic program is proposed in [23] to compute safe control action by integrating (3) as a constraint in the following way

$$\textbf{(CBF-QP)} \quad = \arg\min \| \quad - \quad \|$$
$$\text{s.t.} \quad \sup \quad (\;) + \quad (\;) \quad \geq - \; (\; (\;)) \tag{4}$$

where is usually a high-level performance controller [5] for each robot that can achieve the goal-reaching performance objectives.

III. Problem Formulation

In this work, we assume that robots and obstacles are disk-shaped[2] with radii { } and { } respectively. Let { } and { } be the positions and goal states of robot . The goal is to move robots toward goal states while avoiding collision in a decentralized manner. The destination convergence is equivalent to the state convergence as lim = with being the maximal time instant for = 1 . The state of a robot for $\in \mathcal{V}$ at time is *safe* (collision free) when it belongs to the following sets, termed as *safe sets* given by,

$$\mathcal{C} \quad = \{ \quad \in \quad | \quad (\cdot) \geq 0 \quad \forall \quad \in \mathcal{V} \quad \neq \; \} \tag{5}$$

$$\mathcal{C} \quad = \{ \quad \in \quad | \quad (\cdot) \geq 0 \quad = 1 \;\; 2 \quad \} \tag{6}$$

where $(\cdot)$ is a CBF for collision avoidance between the robots and , and $(\cdot)$ is a CBF for collision avoidance between robot and obstacle . The combined safe set for robot is $\mathcal{C} \; = \mathcal{C} \; \cap \mathcal{C}$, with $\mathcal{C}$ and $\mathcal{C}$ being time-varying safe sets (for robot) w.r.t other robots and obstacles respectively. Due to the changing cardinality of the neighbor set $\mathcal{N}$ of robot with time , the safe sets are time-varying.

The definition of safe sets allows us to formulate the problem of multi-robot navigation as follows.

Problem 1 (Decentralized Safe Multi-Robot Navigation). For the MRS $\mathcal{A}$ with dynamics (1) and control input constraints, goal states { } , homogeneous sensing radius , environment-dependent vector and candidate control barrier functions { $(\cdot)$ $(\cdot)$} , design *decentralized safe control policies* , conditioned on local neighborhood and environment information, that guarantee safety and liveness for all $\in [0 \quad]$ defined as

$$\textbf{(Liveness)} \quad \lim \| \quad - \quad \| = 0$$
$$\text{s.t. } \textbf{(Safety)} \quad \in \mathcal{C} \quad \cap \mathcal{C} \quad = \mathcal{C} \tag{7}$$

while generalizing to a new environment.

To address liveness problem in Problem 1, we assume the presence of a high-level performance controller { ( )} similar to other works [5], [11] for each robot that fulfills the liveness property by steering robots from the initial states { } to the goal states { } but may not necessarily satisfy the safety property. We then design a decentralized control policy based on CBF to supplementarily ensure satisfaction of the safety property, especially in novel environments.

IV. Methodology

We propose addressing Problem 1 by employing a decentralized control policy, as described in (2), to ensure scalability for an arbitrary number of robots. We use CBF conditions to guarantee safety through the invariance of the safe set, leveraging local neighborhood information. To enhance the generalizability of classical decentralized CBF conditions, we propose parameterizing the class-$\mathcal{K}$ function using a GNN and implicitly embedding environment-dependent information into this function. We ensure by construction that the GNN-based parameterization satisfies the properties of class-$\mathcal{K}$ and thus maintains the set-invariance guarantees associated with classical CBF conditions [23] utilizing this GNN-based class-$\mathcal{K}$ function. We employ differentiable optimization layers [25] to incorporate CBF-based quadratic programming (**dCBF-QP**) for computing safe control actions into a neural network. We then propose a loss function to jointly optimize the parameters of the GNN and safe-control policies for each robot in an end-to-end manner. Specifically, at each time step for each robot , we formulate the dCBF-QP problem as

dCBF-QP

$$(\quad | \quad) = \quad \arg\min \quad \| \quad - \quad \| + (\quad)$$
$$\text{s.t.} \quad \sup \quad (\quad) + \quad (\quad)$$
$$+ \quad (\quad (\quad)) + (\quad) \;\; \geq 0 \;\; \forall \;\; \in \mathcal{N} \tag{8}$$

where, (|) is the safe policy parameterized by for each robot and conditioned on the high-level performance control input $(\cdot)$ which accomplishes the goal reaching objectives, environmental information , and local neighborhood information . We refer to $(\cdot)$ as **-Net**, which is a class-$\mathcal{K}$ function parameterized by for each robot . The parameter $\in$ is a penalty for slack variable $\in$. The slack variable guarantees the feasibility of the optimization problem. For each robot , $\mathcal{U} \; \subset$ denotes the control input constraint set. Here $(\cdot)$ is a CBF candidate (safety constraint) for robot w.r.t $\in \mathcal{N}$.

A. Structure of the -Net.

We aim to employ learning-based approaches to capture environment-dependent interactions among multiple robots. Specifically, for each robot , we consider a special structure for the class-$\mathcal{K}$ function (( )) := (( )) , where { } are parameters of CBFs constrained to be positive. We parameterize ({ } |) in (8) as a neural network with parameters conditioned on and .

[2]We can also consider non-circular obstacles. If the shape of the obstacle is non-circular, one can define as the minimum circular radius of a disc that fully encloses the entire obstacle.

Here denotes the local information set of robot (e.g. states of other robots and obstacles in the sensed region).

Modeling as a neural network, conditioned on the local neighborhood information vector and an environment-dependent vector , presents a challenge. The information in is structured as a graph, requiring the network to be *permutation invariant* (i.e., insensitive to the order of neighboring robots) and handle *variable input size* due to the changing cardinality of the neighbor set $\mathcal{N}$ for robot over time .

To tackle these challenges, we use GNN especially because of their permutation invariant characteristics [26]. Refer to [27] for preliminaries on GNN. These neural networks take node and edge features as input, accommodating varying input sizes. They also demonstrate the capacity to generalize across unseen graph topologies [28], [29] and allow for decentralized execution. We model the class-$\mathcal{K}$ function, parameterized by , using GNN. We exploit the observation that robot safety can be analyzed through relative information, such as the relative position of robot w.r.t other obstacles [30], and devise a GNN that is translation-invariant, employing *neural message passing* in which vector messages are exchanged between nodes in the graph to generate CBF parameters based on relative information.

For each robot with its state , the states of other neighbouring robots $\{\ \}$, environment vector and the positions of neighboring obstacles $\{\ \}$, our GNN generates the CBF parameters $\{\ \}$ as

$$\{\quad\} = \tag{9}$$

where denotes a differentiable, permutation invariant function (e.g. sum), and and denote differentiable functions such as MLPs (Multi-Layer Perceptrons) [31]. We use to denote edge features (e.g. relative distance) at time from $\rightarrow$, and are learnable parameters. We use the states of the robots , directly as node features in our GNN. The GNN-based parameterization allows for *decentralized execution, translation invariance, and permutation invariance*.

Remark 1: It is crucial to emphasize that our use of a GNN to parameterize the class-$\mathcal{K}$ function does not jeopardize the set-invariance guarantees inherent in classical CBF conditions. This is guaranteed by construction, as we ensure that the GNN satisfies the properties of a typical class-$\mathcal{K}$ function by constraining $\{\ \}$ to be positive.

B. End-to-End Offline Training

We jointly learn -Net and decentralized control policy for all robots with safety guarantees in an end-to-end manner by embedding the dCBF-QP in (8) as shown in Fig. 2. as a layer on the top of GNN -Net defined in (9) and optimize the learnable parameters $=\{\quad\}$ offline by minimizing the following loss function

$$\begin{aligned} \min \quad & \mathcal{L}(\quad) \\ \text{s.t.} \quad & \{\quad\} = \quad \cdot \quad \forall \\ & = (\ \mid \ )\ \forall \\ & = (\ ) + (\ ) \quad \forall \end{aligned} \tag{10}$$

where is the environment vector sampled from an environment distribution $\mathcal{D}$, is the performance control input provided by high-level performance controller and $=[\quad]$ represents trajectory rollouts for a time horizon using the safe control action generated by the policy for robot . We evaluate the loss at the state for each time step . Note that $\mathcal{L}$ is the cost along a trajectory instead of the cost at each time step, In general, a loss function can be designed with any performance evaluation metric such as $\mathcal{L} = \| - \|$. We also include slack variable penalty to address the infeasibility issue of solving the dCBF-QP in (8). Our loss function comprises two components, drawing inspiration from previous works on single-robot systems [6]

$$\mathcal{L}(\quad) = \mathcal{L}(\quad) + (\ ) \tag{11}$$

where represents the value of slack variable at each time step as defined in (8), and coefficient is a slack variable penalty. The computed performance loss over the joint trajectories of all robots with time duration is backpropagated through the dCBF-QP to the -Net, facilitating updates to the learnable parameters $=\{\quad\}$. The training is done offline and the GNN-based class-$\mathcal{K}$ function can be deployed to different environments sampled from $\mathcal{D}$.

V. Experiments

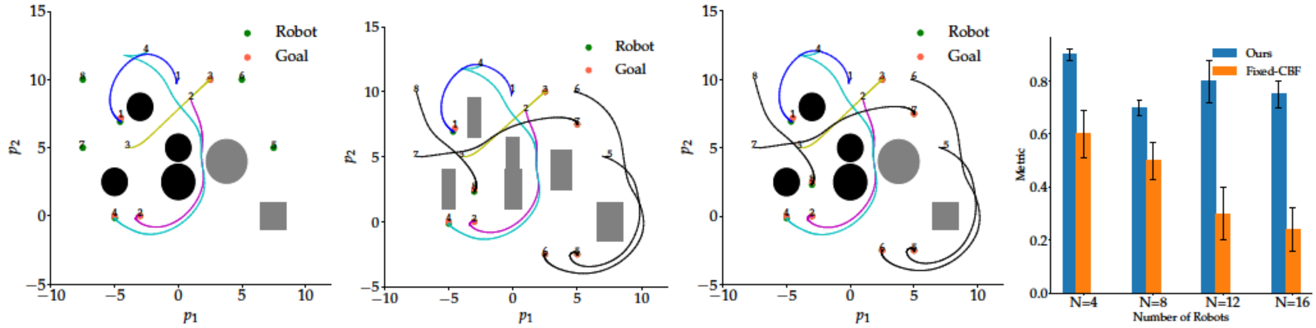
We conduct a multi-robot simulation with robots having the unicycle dynamics:

$$\tag{12}$$

where $=[\quad]$ denotes the position, denotes the orientation and $=[\quad]$ is the control input of robot . We consider static obstacles in the 2D-plane at positions $=[\quad]$ for $=1$. We consider CBF candidates (safety constraints) based on Euclidean distance

$$\begin{aligned} (\ ) &= \| - \| - (+) \\ (\ ) &= (-) + (-) - \\ &- \arctan \frac{-}{-} - (+) \end{aligned} \tag{13}$$

The safety constraints above are formulated to ensure that the Euclidean distance between the robot and other robots (or obstacles) exceeds a predefined safety margin to ensure safety. The resulting pairwise CBF constraints are



(a) (b) (c) (d)

Fig. 3: Multi-robot safe navigation with GNN-based class- functions using unicycle dynamics. Leveraging our GNN-based class- function, robots successfully navigate to their goal positions while avoiding collisions across different environments with increasing density of robots and type of obstacles. Environment **(a)** depicts 4 robots with circular and square obstacles, **(b)** shows 8 robots with rectangular obstacles, and **(c)** displays 8 robots with a mix of circular and square obstacles. Fig. **(d)** illustrates a performance comparison, showcasing a higher weighted success rate of our approach compared to fixed class- function baselines across 100 different environment configurations with varying number of robots.

$$\dot{h}_{i,j}(x_i) + \alpha^{\theta_i}(h_{i,j,a}(x_i)) \quad 0 \tag{14}$$

$$\dot{h}_{i,l}(x_i) + \alpha^{\theta_i}(h_{i,l,o}(x_i)) \quad 0, \tag{15}$$

where $\alpha^{\theta_i}(\ ) := \eta_i(\ )^{\xi_i}$ and η_i, ξ_i are the parameters of dCBF-QP. For simplicity, we assume that, both the robot-to-robot and robot-to-obstacle CBF constraints share the same set of CBF parameters η_i, ξ_i for each robot A_i.

Results: Generalizability. Incorporating environment-specific information into the class- function and parameterizing it through GNN facilitates the fine-tuning of the function for generating safe control actions that can adapt to novel scenarios. In Fig. 1b, robots 2 and 4 initially struggled to reach their goals safely with a fixed class- function. However, they successfully achieved this task with our GNN-based class- function capable of adapting to changing environments as shown in Fig. 3a.

Using local neighborhood information presented as a time-varying graph, the class- function dynamically adjusts its value, as demonstrated in Fig. 3a, to produce feasible and safe control actions. Notably, robot 4 (depicted in blue) undergoes a mid-way switch in the class- function. This switch is strategically employed to navigate between obstacles and reach its goal, indicating a transition from a hard constraint to a soft constraint within the CBF framework. This switching allows the robot to safely maneuver through obstacles. Our goal is to achieve this precise switching between conservative maneuvers with hard constraints and more aggressive maneuvers with soft constraints in different environments, accomplished through a parameterized class- function that implicitly depends on the environment.

Scalable and Decentralized Execution. Using GNN based parameterization of the class- function and CBF conditions as outlined in (8) enables the decentralized execution of each controller by leveraging local-neighborhood information. GNN exhibits invariance to the order of robots and possesses the ability to handle graphs with dynamically changing nodes and edges, offering scalability to accommodate an arbitrary number of robots.

To evaluate the scalability and generalizability of our framework, we first train the class- function for four robots. Subsequently, we investigate whether the learned interactions captured within the α-Net for four robots can be transferred to an additional group of four robots with randomly assigned initial and goal positions. To achieve this, we replicate the parameters of the α-Net between robots, i.e., $\theta_i \;{}_{i=1}^{4} \xrightarrow{copy} \theta_i \;{}_{i=4}^{8}$, and simulate a safe-navigation scenario for $N = 8$ robots, as illustrated in Fig. 3b and 3c. The supplementary robots successfully leverage the learned interactions, originally learned for the four-robot MRS, to navigate safely to their goals. This shows that learned class- functions can adapt to arbitrary graph topologies without the need for retraining. The trajectories of these additional robots are denoted in black in Figs. 3b and 3c.

For a quantitative assessment of our work with other baselines, we employ the Success Weighted by Path Length (SPL) metric [32] which combines success rate and path length towards the goal. In particular, we compare our work with heuristic-based methods that pre-tune CBF class- through an exhaustive grid search in a fixed environment, as illustrated in Fig. 3a. To evaluate the generalizability of both approaches, we systematically alter the environment by changing obstacle shapes (circular, rectangular, square) and gradually modifying their positions. We conduct tests across 100 different environmental configurations, varying the number of robots. The results, presented in Fig. 3d, indicate that our framework demonstrates a higher (normalized) success rate with lower standard deviations compared to the heuristic-based fixed-CBF parameter baseline.

VI. Conclusion

We introduce a novel parameterization of class- functions for enhancing the generalizability of control barrier functions for safe multi-robot navigation to novel environments. Our approach involves embedding environment-specific information into class-$\mathcal{K}$ functions parameterized by GNN implicitly and employing differentiable optimization layers to optimize it jointly with the control policy in an end-to-end manner using local neighborhood data. The GNN-based class-$\mathcal{K}$ function dynamically adjusts its values in novel environments and this adaptive approach addresses the inherent challenges posed by fixed class-$\mathcal{K}$ functions, aiming to achieve a suitable balance between performance and safety in novel environments. Future work involves optimizing class-$\mathcal{K}$ functions directly in high-dimensional observation spaces.

VII. ACKNOWLEDGMENTS

This work is financially supported by Air Force Office of Scientific Research (AFOSR) Grant #FA9550-21-1-0411, National Aeronautics and Space Administration (NASA) University Leadership Initiative (ULI) Grant #80NSSC17M0051, National Science Foundation's Civil, Mechanical, and Manufacturing Innovation (CMMI,) Information and Intelligent Systems (IIS), and Electrical, Communications and Cyber-Physical Systems Grants #2135925, #2423130, #2211542, #2331878, and #2145134.

REFERENCES

[1] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-Agent Systems: A Survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.

[2] D. Zhang, G. Feng, Y. Shi, and D. Srinivasan, "Physical Safety and Cyber Security Analysis of Multi-Agent Systems: A Survey of Recent Advances," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 2, pp. 319–333, 2021.

[3] 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.

[4] K. Galloway, K. Sreenath, A. D. Ames, and J. W. Grizzle, "Torque Saturation in Bipedal Robotic Walking Through Control Lyapunov Function-Based Quadratic Programs," *IEEE Access*, vol. 3, pp. 323–332, 2015.

[5] 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.

[6] H. Ma, B. Zhang, M. Tomizuka, and K. Sreenath, "Learning differentiable safety-critical control using control barrier functions for generalization to novel environments," in *2022 European Control Conference (ECC)*. IEEE, 2022, pp. 1301–1308.

[7] K. Garg, S. Zhang, O. So, C. Dawson, and C. Fan, "Learning Safe Control for Multi-Robot Systems: Methods, Verification, and Open Challenges," 2023.

[8] C. Dawson, S. Gao, and C. Fan, "Safe Control With Learned Certificates: A Survey of Neural Lyapunov, Barrier, and Contraction Methods for Robotics and Control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 1749–1767, 2023.

[9] M. Saveriano and D. Lee, "Learning barrier functions for constrained motion planning with dynamical systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 112–119.

[10] M. Srinivasan, A. Dabholkar, S. Coogan, and P. A. Vela, "Synthesis of control barrier functions using a supervised machine learning approach," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 7139–7145.

[11] C. Dawson, B. Lowenkamp, D. Goff, and C. Fan, "Learning safe, generalizable perception-based hybrid control with certificates," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1904–1911, 2022.

[12] A. J. Taylor, V. D. Dorobantu, H. M. Le, Y. Yue, and A. D. Ames, "Episodic learning with control lyapunov functions for uncertain robotic systems," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 6878–6884.

[13] A. Taylor, A. Singletary, Y. Yue, and A. Ames, "Learning for safety-critical control with control barrier functions," in *Learning for Dynamics and Control*. PMLR, 2020, pp. 708–717.

[14] Z. Qin, D. Sun, and C. Fan, "Sablas: Learning safe control for black-box dynamical systems," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1928–1935, 2022.

[15] W. Xiao, T.-H. Wang, R. Hasani, M. Chahine, A. Amini, X. Li, and D. Rus, "Barriernet: Differentiable Control Barrier Functions for Learning of Safe Robot Control," *IEEE Transactions on Robotics*, vol. 39, no. 3, pp. 2289–2307, 2023.

[16] Z. Qin, K. Zhang, Y. Chen, J. Chen, and C. Fan, "Learning Safe Multi-Agent Control with Decentralized Neural Barrier Certificates," 2021.

[17] S. Zhang, K. Garg, and C. Fan, "Neural Graph Control Barrier Functions Guided Distributed Collision-avoidance Multi-agent Control," in *7th Annual Conference on Robot Learning*, 2023.

[18] Z. Gao, G. Yang, and A. Prorok, "Online Control Barrier Functions for Decentralized Multi-Agent Navigation," in *2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, 2023, pp. 107–113.

[19] Q. Li, F. Gama, A. Ribeiro, and A. Prorok, "Graph neural networks for decentralized multi-robot path planning," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 11 785–11 792.

[20] B. Hudson, Q. Li, M. Malencia, and A. Prorok, "Graph neural network guided local search for the traveling salesperson problem," *arXiv preprint arXiv:2110.05291*, 2021.

[21] J. Blumenkamp, S. Morad, J. Gielis, Q. Li, and A. Prorok, "A framework for real-world multi-robot systems running decentralized gnn-based policies," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 8772–8778.

[22] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[23] ——, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[24] H. Khalil, *Nonlinear Systems*, ser. Pearson Education. Prentice Hall, 2002.

[25] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and J. Z. Kolter, "Differentiable convex optimization layers," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019.

[26] L. Wu, P. Cui, J. Pei, L. Zhao, and X. Guo, "Graph Neural Networks: Foundation, Frontiers and Applications," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, ser. KDD '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 5831–5832.

[27] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.

[28] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[29] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks," 2018.

[30] L. Wang, A. D. Ames, and M. Egerstedt, "Safety Barrier Certificates for Collisions-Free Multirobot Systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[31] J. Singh and R. Banerjee, "A study on single and multi-layer perceptron neural network," in *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)*, 2019, pp. 35–40.

[32] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva *et al.*, "On evaluation of embodied navigation agents," *arXiv preprint arXiv:1807.06757*, 2018.