IEEE Robotics and Automation Letters (RAL) paper presented at the
2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

Defensive Escort Teams for Navigation in Crowds via Multi-Agent Deep Reinforcement Learning

Yazied A. Hasan[1], Arpit Garg[1], Satomi Sugaya[1], and Lydia Tapia[1]

***Abstract*— Coordinated defensive escorts can aid a navigating payload by positioning themselves strategically in order to maintain the safety of the payload from obstacles. In this paper, we present a novel, end-to-end solution for coordinating an escort team for protecting high-value payloads in a space crowded with interacting obstacles. Our solution employs deep reinforcement learning in order to train a team of escorts to maintain payload safety while navigating alongside the payload. The escorts utilize a trained centralized policy in a distributed fashion (i.e., no explicit communication between the escorts), relying only on range-limited positional information of the environment. Given this observation, escorts automatically prioritize obstacles to intercept and determine where to intercept them, using their repulsive interaction force to actively manipulate the environment. When compared to a payload navigating with a state-of-art algorithm for obstacle avoidance our defensive escort team increased navigation success up to 83% over escorts in static formation, up to 69% over orbiting escorts, and up to 66% compared to an analytic method providing guarantees in crowded environments. We also show that our learned solution is robust to several adaptations in the scenario including: a changing number of escorts in the team, changing obstacle density, unexpected obstacle behavior, changes in payload conformation, and added sensor noise.**

I. Introduction

Successful navigation in crowded scenarios often requires assuming a non-zero collision probability between the agent and stochastic obstacles [1]. This required assumption of risk is potentially frightening given the value of cargo that modern autonomous agents will be transporting, e.g., human life. In many real-world scenarios, humans employ escorts for enhanced safety in crowds during high-consequence navigation, e.g., a parent with a child, presidential security, or military convoys. For example, the US Army employs a tactical convoy to move a payload, personnel and/or cargo, via a group of ground vehicles to or from a given destination. Some of the vehicles in the convoy act as coordinated escorts to prevent traffic from overtaking the convoy, dispersing crowds, and/or establishing a secure perimeter (cordon area) that is essential to the safety of the soldiers. We focus on the protection of a payload, similar to this convoy protection scenario, and provide a deep reinforcement learning (deep RL) solution. Our solution is able to learn pedestrian-like, interacting, environmental dynamics and exhibits emergent cooperative behavior while no explicit communication between the escorts is considered in the system design. Fig. 1 illustrates our problem setup. The navigating *payload* is represented by the orange circle surrounded by three blue dots, *defensive escorts*, protecting the payload. The escorts protect the payload by enforcing the *cordon area* around the payload, encompassed by the orange ring, while the payload navigates to the destination, the black dot labeled 'Goal'. The environment is crowded with *interacting moving obstacles*, e.g., pedestrians, shown by the gray dots. The blue ring centered around each escort represents the lidar sensor range.

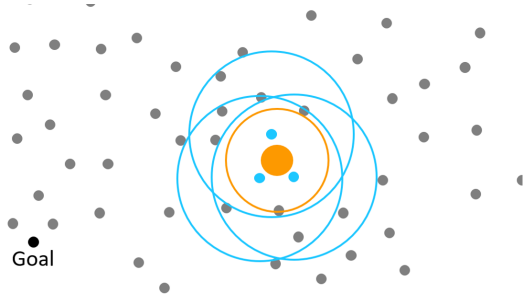


Fig. 1. Our experimental setup demonstrates the payload (large orange dot) navigating to the goal (black dot) with coordinated escorts (blue dots) interacting with obstacles (grey dots). The blue outlined circles indicate the sensor radius of the escorts. The orange outlined circle indicates a cordon safety area around the payload.

Over several decades, diverse variations of related problems have been considered (detailed in Sec. II) including: convoy protection [2]–[5], perimeter surveillance [6], [7], multi-robot coordination [8], multi-player perimeter defense [9], [10], multi-player reach avoid [11], [12], and guarding a territory game [13]–[15]. Some recent work considers variations that are similar to ours, including moving obstacles, payload protection and obstacle interception by escorts [9]–[12], [16]–[19]. However, due to the curse of dimensionality [20], the obstacle dynamics don't consider interaction and are first-order [9]. Solutions based on learning have recently emerged as potential solutions. However, their focus has been on non-interacting obstacles [10], [13], [14], [18], consider only one obstacle [13], [14], or assume some communication between escorts [10], [18].

*This work is partially supported by the National Science Foundation (NSF) under Grant Numbers IIS-1528047 and IIS-1553266. This work is also partially supported by the Air Force Research Laboratory (AFRL) under agreement number FA9453-18-2-0022. Also, partial support was provided by the Army Research Laboratory (ARL) and was accomplished under Cooperative Agreement Number W911NF-19-2-0215. Any opinions, findings, conclusions, recommendations, or views contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the NSF, AFRL, ARL or U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

[1]Department of Computer Science, University of New Mexico, MSC01 11301 University of New Mexico, Albuquerque, NM 87131, USA yhasan@unm.edu, kiralobo@cs.unm.edu, satomi@cs.unm.edu, and (corresponding) tapia@cs.unm.edu

Copyright ©2020 IEEE

We set out to protect the moving payload *in an environment with a high density of non-aggressive but interacting moving obstacles*. Solutions such as ours are critical, as previous work has shown that successful navigation is not possible when aggressive obstacles grossly outnumber the escorts [9]. Additionally, solutions are highly relevant to real-world applications where robotic agents are tasked to navigate through a crowd, e.g., pedestrians, using observations. We consider that the number of obstacles is *orders of magnitude larger* than the number of escorts, e.g., 50 obstacles to 3 escorts. The obstacles interact with one another and with the escort through a social force model [21].

The complexity of the problem dynamics due to numerous interacting moving obstacles motivates an end-to-end learning solution. We present an autonomous solution to the navigation scenario where a team of escorts learn to defend the payload. A trained centralized policy is used in a distributed manner by each escort. Escorts learn to follow along the navigating payload and to disperse obstacles expected to enter the cordon region. Our deep RL solution takes only range-limited observations of the environment, *i.e.*, only positional information of obstacles within a sensor range (not velocity or acceleration), and no other information. The escorts learn to adapt to obstacles with interacting motions and *automatically coordinate* themselves around the payload. Each escort merely proximally observes other agents, *i.e.*, escorts, obstacles, and the payload.

Our deep RL solution provides enhanced safety of the payload along a fixed navigation route as compared to agile maneuvering of the payload using a state-of-the-art obstacle avoidance algorithm [22], formations of escorts [6]–[8], and an analytical solution with guarantees [9]. Our learned solution is scalable to number of obstacles and escorts and is robust to changes in scenarios such as a changing conformation of the payload, unexpected obstacle motion, and sensor noise. A video highlighting results is attached.

II. Related Work

There are several related problems regarding protection where the terminology varies widely. While we provide a perspective on the recent literature, we fix the terminology in Section I to avoid confusion. For example, synonymous terms to our obstacles are: attackers, intruders, evaders, invaders, adversaries, and synonymous to our escorts are: defenders, pursuers, guardians, bodyguards. Our terms, payload and cordon, can be thought of as the convoy, target, or territory. One category of related methods focuses on strategies to guarantee visual surveillance of a static or moving payload. Optimal control [2], [3], moving path following methods [4], and hierarchical system approaches [5] are used to solve for the surveillance strategies in both an idealized model [2], [3] and physical models with constraints on the escorts and the payload motions [4]. While these problems focus on the strategy to provide continuous surveillance of the payload, it is in the absence of moving obstacles.

The most closely related work to our problem focuses on protecting a moving or static region from obstacles using defensive escorts. This is typically handled in two ways by the escorts, by either employing passive or active behavior. In the first category, escorts provide passive protection, taking no actions to actively pursue and intercept obstacles. Using sampling-based motion planning, [6], [7] propose a virtual fence, created by multiple robot escorts circling around the perimeter, to protect the perimeter of a moving cordon area. A combination of physics-based motion planning and an artificial potential field method is used in [8] to coordinate escorts to surround a payload. In these methods, full knowledge of the environment is required including full dynamical information, only static obstacles are considered and escorts are restricted to orbiting motion [6], [7] or to a static formation with planning done off-line [8]. In the second category, active interception of aggressive and strategic obstacles, methods such as two-player [13], [15] or multi-player [9], [11], [12], [14], [16], [17] differential games have been used. Optimal strategies and bounds on escorts' performances in multi-player games are found through Hamilton-Jacobi-Isaacs formulation [11], [12] or by using geometric methods [9]. This centralized optimal solution has been mapped to a decentralized escort policy [10] using supervised learning. Some methods also employ fuzzy logic to compensate for uncertainties in observations [13]–[15], and use Fuzzy RL techniques such as fuzzy Q-learning [23] and fuzzy actor-critic [24] to approximate optimal solutions. In these methods, however, the obstacle dynamics considered are of first-order [9], [14], no obstacle-obstacle interactions arising from the dynamics are considered, and full knowledge of the obstacles within observation are required. In order to address some of the issues of other active escort solutions, a deep RL solution was proposed recently for perimeter defense games [18]. That work considers a robotic escort team protecting a VIP payload from aggressive and non-aggressive obstacles. While the problem set up is similar to ours, that work does not consider interacting obstacles and relies on communication between the escorts.

Deep RL has recently shown great success on highly dynamic navigation tasks [25]. Some methods combine long-range planning with highly adaptable short-range deep RL solutions that continually replan in order to navigate collision free [26], [27]. Some of our prior work presented a deep RL solution for navigating in dynamic environments and compared the learned collision probabilities against a formal and complete method [22]. Other navigation-based problems like Pursuit-Evasion and Waterworld have been previously studied by extending deep RL algorithms to cooperative multi-agent systems [28] that do not use any explicit communication. Although these solutions involve dynamic obstacle avoidance and learning cooperative navigation, the navigation objectives do not involve enhancing safety by escorting a moving payload.

III. Problem Formulation

Rather than finding or approximating an escort's policy by analytic or semi-analytic means or through dynamic programming [29], reinforcement learning approximates the pol-

icy by trial-and-error commonly through policy gradient [20]. The gradient includes the information about the quality of the policy and the change to improve the policy. This quality information, through the value of a state, depends on the reward structure. Value of a state (or state value) is the expected cumulative future reward of the current state. A unique advantage of a RL solution is that it is agnostic to the system dynamics, *i.e.*, agent and environmental dynamics such as properties of the obstacle space and motions. This is because RL is devised to learn the mapping between observation and action given an objective (cumulative reward) and environment.

A. System Dynamics

We consider holonomic escorts protecting a payload that navigates in a straight line with constant speed from a start to a goal. The surrounding environment is *densely* populated with non-aggressive interacting obstacles, *i.e.,* moving with acceleration. The obstacles are assumed to move at most at the speed of the escorts as seen in similar problems [9], [22]. This assumption is realistic given pedestrian-like obstacles.

These obstacles interact both between themselves and with the escorts through a social force model originally designed for interacting pedestrians [21]. In this model, temporal change in the velocity, $\boldsymbol{w}_k$, of an obstacle k is obtained through the force acting on it, $\boldsymbol{F}_k(t)$,

$$\boldsymbol{F}_k(t) := \boldsymbol{F}_{\boldsymbol{k}}^{\boldsymbol{0}} + \sum_{l \neq k} \boldsymbol{F}_{kl} + \sum_{i} \boldsymbol{F}_{ki}, \quad \frac{d\boldsymbol{w}_{\boldsymbol{k}}}{dt} := \boldsymbol{F}_k(t). \quad (1)$$

Note that we assume all the obstacles and escorts to have the same mass, and without loss of generality, set the mass to unity. Hence, the force terms are assumed to be forces per unit mass although we refer to them as forces. The *social force* takes into account the tendency of an obstacle to reach a certain desired velocity with a relaxation time. This effect is given by $\boldsymbol{F}_k^0$, and the *repulsive effects* of other obstacles, l, and escorts, i, are given by $\boldsymbol{F}_{kl}$ and $\boldsymbol{F}_{ki}$, respectively.

The repulsive potential is implemented as in [21], and is assumed to decrease exponentially in the form of an ellipse that is directed into the direction of motion. The repulsive effects are only felt if an obstacles is within the influential radius and inside the directionally dependant vision cone of other obstacles. The escorts can apply social forces on the obstacles. However, escorts don't apply social forces amongst one another and the escorted payload does not apply social forces. The lack of social force on the payload has been commonly used due to the presence of distracted pedestrians whose unawareness could cause collision [30].

B. Multi-agent Partially Observed Markov Decision Process

Formally, the Payload Protection Problem can be formulated as a multi-agent extension of a Partially Observed Markov Decision Process (POMDP) [31], given as a tuple $(S, \mathcal{A}^N, \mathcal{O}, R, \mathcal{T}, \rho, \mathcal{N}, \mathcal{K}, \gamma)$, where agents are the escorts. $\mathcal{N} = \{1, 2, \ldots, i, \ldots, N\}$ and $\mathcal{K} = \{1, 2, \ldots, k, \ldots, K\}$ are the sets of homogeneous escorts and homogeneous obstacles in the system, respectively. At a given time, $s_i \in S_i$, $s_k \in S_k$, and $s_p \in S_p$ are the states of the i-th escort, k-th obstacle, and the payload. The state space S of the system is given by $S \equiv \{S_i\}_{i\in\mathcal{N}} \times \{S_k\}_{k\in\mathcal{K}} \times S_p$.

At each step, for a given state $s \in S$, the escort $i \in \mathcal{N}$, receives a range-limited observation $o_i \in \mathcal{O}_i$, determined by the conditional observation probability $\rho(s, o_i) = P(o_i|s)$ and takes an action $a_i \in \mathcal{A}$ given by a parametrized policy, $\pi_\theta(o_i, a_i)$, where θ represents the set of parameters. Given actions from all the escorts, a joint action $\{a_i\}_{i\in\mathcal{N}} = a \in \mathcal{A}^N$ is formed which induces transition in the environment according to the state transition function $\mathcal{T}(s, a, s') = P(s'|s, a)$.

The observation of i-th escort is given by $o_i = (\{o_{i,k}\}_{k\in\mathcal{K}'}, \{o_{i,j}\}_{j\neq i\in\mathcal{N}'}, o_{i,p})$ where $\mathcal{K}'$, $\mathcal{N}'$, $\{o_{i,k}\}_{k\in\mathcal{K}'} \in O_{i,\mathcal{K}}$, $\{o_{i,j}\}_{j\neq i\in\mathcal{N}'} \in O_{i,\mathcal{N}}$, and $o_{i,p} \in O_p$ respectively, are the subset of obstacles, escorts, and the payload within the sensor range of i-th escort. This observation is assumed to be made by an arbitrary sensor which we chose to be a 1-d lidar with α rays equally distributed radially. Specifically, $\{o_{i,k}\}_{k\in\mathcal{K}'}$ is a vector of α elements where each element is the minimum of distance to the nearest obstacle and the lidar range, along each ray. $\{o_{i,j}\}_{j\neq i, j\in\mathcal{N}'}$ and $o_{i,p}$ are defined similarly. Range-limited observation is due to the finite sensor range and one assumption on the lidar: while agents of the same kind, *i.e.*, escort-escort and obstacle-obstacle, occlude each other from the lidar rays, agents of different kinds do not occlude.

For the action a_i in state s, the escort i receives a global reward $R(s, a_i)$. Each escort individually tries to maximize their expected cumulative reward, $\mathop{\mathbb{E}}_{\tau\sim\pi}[R(\tau)]$, discounted by γ, where τ represents a sequence of states and actions of the escorts following the policy π.

The payload protection task aims to find an escort policy π_θ that maps observations to robot action while maximizing payload safety. Safety is enforced by minimal probability of collision while navigating to the goal. Collision events, $\mathbb{C}$, are impacts that involve the payload and/or an escort. Additionally, it is often critical to defend a zone around the payload by minimizing the probability of any obstacles entering a cordon area, $\mathbb{B}$. This translates to $\pi_\theta = \arg\min_\theta \mathbb{P}(\mathbb{B} \cup \mathbb{C})$, where $\mathbb{P}(\mathbb{B} \cup \mathbb{C})$ is the joint probability of the collision and cordon breach events.

IV. Method

A. Reward

To train the escorts we design a reward function that acts as a signal to reinforce desired behavior. For a given state, s, the reward function $R(s)$ is defined to assign a reward $r_{goal} = 1$ when the payload reaches the goal, $r_{collision} = -1$ when a collision occurs, $r_{step} = 0.01$ at every timestep, and r_{cordon} when the cordon area is breached. Since our escorts are homogeneous in terms of their goals and capabilities, we reward all the escorts by the same global reward. The penalty for cordon breach, r_{cordon}, assigns negative reward for every obstacle that penetrates the cordon area proportional to their

proximity to the payload and is defined as

$$r_{cordon} = -c \sum_{\{o_i | d(\boldsymbol{x}_p, \boldsymbol{x}_k) < S_{cordon}\}} \left(1 - \frac{d(\boldsymbol{x}_p, \boldsymbol{x}_k)}{S_{cordon}}\right), \quad (2)$$

where $d(\boldsymbol{x}_p, \boldsymbol{x}_k)$ is the distance between the payload, $\boldsymbol{x}_p$, and obstacle, $\boldsymbol{x}_k$, positions, S_{cordon} is the radius of the cordon area. The parameter c is a constant that tunes the penalty an escort receives per an obstacle entering the cordon region. A large c-value results in high penalty, which empirically resulted in the escorts colliding with the payload to end episodes. A low value reduces the significance of the cordon region. In all results shown, we used a value of 0.5.

B. Escort Policy Training

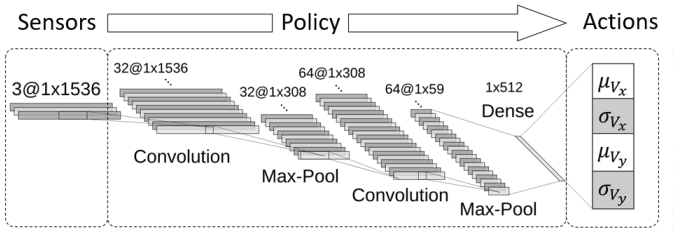


Fig. 2. Neural network architecture. The network takes in the sensor information from each type of sensed object: payload, obstacles, and other escorts, and outputs a diagonal Gaussian distribution from which continuous actions are sampled. The network consists of 2 sets of alternating convolutional and max-Pooling layers followed by a flattened dense layer.

The large continuous state space of the escorts motivates a deep RL approximate solution for the Payload Protection Task. While there exist many deep RL solutions, a class of policy gradient algorithms [32], actor-critic methods [33], have been widely used in the RL scheme that train a value function, *i.e.*, critic, using Bellman's equation to estimate the gradient of the performance. The gradient is then followed to update the policy, *i.e.*, the actor. This reduces the variance thus stabilizing the training. Generalized Advantage Estimation (GAE) [34] is an actor-critic method that improves sample efficiency and further stabilizes the learning by using an exponentially-weighted estimator of the advantage function as a baseline function and by using trust region optimization [35] for both the actor and the critic.

We train multiple escorts that share a single GAE stochastic policy, an approach that is similar to independent actor-critic with shared parameters [28], [36], using RLlib [37]. The actor and critic are represented by two separate networks having the same architecture. Our sensor provides information about the payload's, obstacles' and escorts' shape, and location. In order to obtain this information, we used simulated 1D lidar (512 uniformly spaced radial rays) from each escort. Object classification is implemented by concatenating three lidar distance measurements, one each to detect objects of a single type, *i.e.*, escorts, payload and obstacles. To enable some inference of velocities and accelerations, readings from the last three time steps are passed as an observation. This method, known as frame-stacking, is suitable for parallelization and is easy to converge stably. This forms an array of size 3x1x1,536 (as shown in Fig. 2).

The output of the network is a set of actions for each escort that enables interception of obstacle threats. This was implemented in the network by outputting a diagonal Gaussian distribution, $N([\mu_{V_x}; \mu_{V_y}], [\sigma_{V_x}; \sigma_{V_y}])$, where μ_{V_x} and μ_{V_y}, and σ_{V_x} and σ_{V_y} are the means and standard deviations of the escorts' horizontal and vertical speeds, respectively, from which continuous actions can be sampled.

The full network (Fig. 2) encodes a policy that maps input sensor information to output actions. We implemented this mapping through convolution layers (32 and 64 filters of size 1x10 and stride 1 with ReLU activation) each followed by a max pool layer (size 1x5 and stride 5). The output of the convolutional neural network (NN) is flattened and fed to a fully connected layer (size 512 with ReLU activation). Regarding our network architecture design, we make a note that a convolutional NN was utilised since a simple fully connected network did not represent the observation well. A convolutional NN was favored for its well known ability to recognise shapes and geometry, and reduced number of parameters compared to a fully connected counterpart [38].

We define collision to take place only between objects of any two different types, *i.e.*, *payload-obstacle*, *payload-escort* and *escort-obstacle*. (Collisions among homogeneous agents are not terminal for our problem formulation since obstacle-obstacle interactions are dictated by the dynamics simulator and escort-escort collision is preventable given our full control of escort motion.) We terminate the episode if there is a collision or if the goal is reached.

We use a single GAE policy that is shared between all the escorts. To train this policy we collect experience samples in parallel on 100 cores of Intel Xeon E-2146G @ 3.50 GHz. We train the policy every time a training batch of size 524,288 samples is collected by performing stochastic gradient descents of mini-batch size of 65,536 samples on 4 NVIDIA Tesla V100 GPUs in parallel. We use mean of the rewards of all the samples in a training batch as a metric for convergence that typically occurs in 100M samples and takes about 24hrs and was performed once for all experiments run. For escort adaptability, we used challenging training scenarios. We train the escorts at a high obstacle density (90 obstacles), as we empirically observed that this allowed the escorts to handle lower densities without retraining. Additionally, we train the escorts with each episode employing from one to six escorts, as we empirically observed that training with varying escort numbers produces a more flexible model capable of post-training adjustments.

C. Experimental Setup

For all experiments the following specifications of the environment were used, unless specified otherwise. All objects are circular moving rigid bodies: a payload (radius 1.5m, maximum speed 0.25m/s), obstacles (radius 0.5m, maximum speed 1m/s), and holonomic escorts (radius 0.5m, maximum speed 1m/s). The environment is 50m by 50m. When the objects reach the boundary of the environment, they teleport

and reappear at the opposite boundary. The radius of the cordon area is $S_{cordon} = 5\,\mathrm{m}$. The simulated lidar has a maximum vision distance of 8m, and escorts employ 512 beams at uniform intervals. At the beginning of each episode, obstacles are randomly assigned a position and a desired moving direction. Heuristics help facilitate setup by reducing states in collision and assisting the escorts to initially find the payload: escorts are spawned within 2.5m to 3.5m from the payload, obstacles are at least 4.6m away from the payload, and the goal is exactly 20m away from the payload. For the social force model, the repulsive potential has the maximum amplitude of $7.9\,\mathrm{m}^2/\mathrm{s}^2$, influential radius of 5m, and the vision cone of 200° (in the direction of motion).

D. Assumptions and Limitations

Our method is reliant on both the POMDP formulation and the estimation of the solution through deep RL. Due to both the formulation and estimation, there are some practical limitations that should be noted. The lack of communication and coordination between escorts is provided by a homogeneity assumption of escort behavior and decision making. This enables escorts to interpret the actions of their teammates in order to determine their own best next action. However, other parameters of the method are not as constrained; the dynamics are automatically estimated by the learning, and estimated over multiple observations. This has been shown to be effective for estimating non-linear dynamics for moving obstacle avoidance [22]. Additionally, while the learning structure demonstrated uses sensor input, it is only assuming that positional information of the interacting agents is provided. While the assumption that the multi-channel non-occluded lidar input is currently not realized in hardware, this parameterization demonstrates the flexibility of the learning to learn the mapping of high-dimensional sensor inputs to the complex action space. Also, the reliance on learning comes with limitations. First, as a learning approach, there are no guarantees provided. Second, as static obstacles would not be subject to interacting forces, they were not implemented in the learning. However, robust learning has been shown for navigating with static and dynamic obstacles and in environments with noise [26].

V. Results

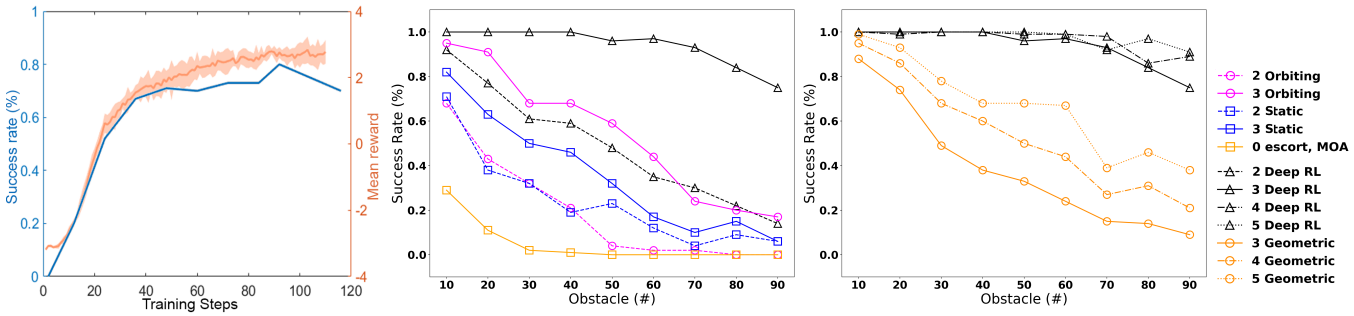
The Payload Protection Task requires both collision free navigation and protection of the cordon area. *Success rate* directly measures our policy's ability to navigate without collision. This metric is computed as the ratio of collision-free runs to all runs. Another metric, *cordon area breach time*, quantifies the obstacles' cordon area breach duration. This time is computed as percentage of the time the cordon area is breached over the total duration of an episode. All experiments are averaged over 100 iterations with different initial conditions unless otherwise specified. In cases where comparisons are being made, the same random seed is used between comparisons to generate the initial conditions for the episodes to ensure they have the same starting configurations. Otherwise, randomized initial conditions were used. We employed GNU Parallel [39] to evaluate experiments in parallel. Stable convergence of our policy is demonstrated in Fig. 3(a) where mean rewards and success rate are shown as functions of training steps.

A. Escort Efficacy and Scalability

To demonstrate the scalability of our solution, we evaluated our escorts in an environment with increasing number of interacting obstacles (Fig.3). This scenario mirrors our problem: parents with a child or military convoy navigating in dense pedestrian crowd. Our escorts show a success rate above 70%, up to 90 obstacles (see (b)). Additionally, our method (due to homogeneity assumptions of the escorts) supports varying numbers of escorts without retraining.

To demonstrate the efficacy of autonomous escorts, we evaluated trained escorts against a state-of-the-art moving obstacle avoidance (MOA) policy [22], a fixed-escort formation strategy inspired by a work in multi-robot coordination [8], an orbiting escorts strategy similar to [6], [7], and an analytical method providing guarantees solving a perimeter defense problem [9] adapted to protect a moving payload. In the MOA approach, the payload itself was trained to avoid obstacles en-route to its destination. The fixed-formation escorts, Static and Orbiting, are prescribed to surround the payload uniformly in a fully static position or orbiting at maximum speed, respectively, and to repulse obstacles with social forces while the payload navigates. To compare with an analytical method, we implement the escort as described in the analytical solution [9] (Geometric), matching escorts to obstacles by geometrically determining interception. The escorts then move towards their matched obstacle at maximum speed. Assignments are frequently reassessed given the crowding in the environment in order to respond to changes in obstacles within visible range.

First, we compare performance against the set of intuitive strategies (Fig. 3(b)): Static, Orbiting, and MOA. The MOA payload performs poorly as it moves slower than the obstacles, as seen in the red-square line where the success rate plummets to 2% when the number of obstacles is above 20. Adding static escorts to the payload, as the payload navigates in a straight line to the goal, improved the success rate by up to 42% (using two escorts), so does increasing the number of escorts (blue-square lines). This was further improved, by up to 54% (Orbiting - pink circle lines). However, the escorts tend to collide with obstacles (67% of collisions were escort to obstacle at the 90 obstacle density) and have difficulty maintaining formation due to the movement of the payload and maximum speed rotation. Our deep RL escorts (black-triangle lines) improve the success rate by up to 99% versus MOA, up to 83% versus Static and up to 69% over Orbiting. The two-escort team finds a solution 14% of the time while the three-escort team finds a solution 75% of the time in the most challenging scenario with 90 obstacles. Next, we compare against the Geometric escorts. We can observe in Fig.3(c) that despite the advantages of Geometric escorts (orange-circle lines), *i.e.*, full observation and optimality



(a) (b) (c)

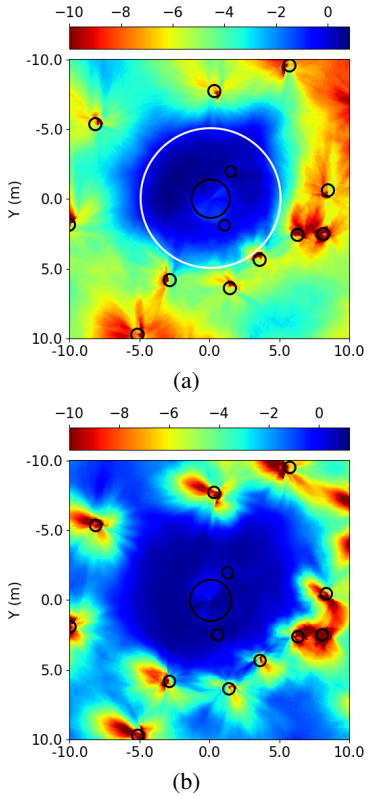
Fig. 3. Demonstration of escort efficacy. (a) Mean reward (right y-axis) averaged over 4 random network initializations (standard deviation shown in shade) and navigation success rate (left y-axis) evaluated every 12 steps during Deep RL training with 3 escorts and 90 obstacles. (b-c) Post-training performance in environments with increasing numbers of obstacles. (b) Comparison with an approach where the payload uses a learned moving obstacle avoidance policy (0 escort, MOA), fixed-formation escorts uniformly surrounding the payload (static) applying social forces to the obstacles, escorts orbiting the payload at max speed (orbiting). (c) Comparison to an analytical solution to a perimeter defense problem (Geometric). The number in the legend indicates the number of escorts.

for directly approaching obstacles, our deep RL method outperforms by up to 66%. The dynamic nature of the problem, with obstacles entering and leaving the visible range causes frequent reassignment, unlike the aggressive obstacle scenario considered in the original paper [9]. It should be noted, trained deep RL escorts are able to enhance protection of the cordon area (quantitative result shown in attached video). For example, the total cordon area breach time is as much as 32 seconds less for deep RL escorts than Geometric escorts at 90 obstacles.

B. Learned Cooperative Behavior and the State Value

In addition to abiding by the explicit constraints in the reward structure, *i.e.*, avoiding collision with the payload, keeping the payload alive, etc, the escorts also display behavior that is *not explicitly* addressed in the reward structure. The escorts learn to stay close to the payload despite there not being a reward for them to do so. They also learn to follow the moving payload and keep pace with it, maintaining an efficient formation as they move. This behavior is guided by the learned value function which can be visualized using a heatmap (Fig. 4(a)). The heatmap is generated by placing an escort at each pixel location of the output plot and sampling the value function. The result shows that the escorts learned the cordon area, as indicated by the larger state value (blue circular region around the payload). This is despite the fact the escorts are never given explicit information about the cordon, and only penalized when it is breached. This is further emphasized when we retrain the escorts with no regard to cordon breaches (cordon breach reward is 0) (Fig. 4(b)). The escorts no longer have as strong of a preference to staying within that area; the higher state value region is not as confined to the cordon area.

The escorts also learn to cooperate with each other, exhibited in their tendency to spread evenly around the payload to offer the most coverage. Recall that an escort has positional information of other escorts who are within the lidar range. Besides this, there is no explicit communication between the escorts, *i.e.*, cooperation or collision with each other are not rewarded and no extra message passing between the escorts



(a) (b)

Fig. 4. Value functions learned by an escort for the Payload Protection Task (a) with a penalty for obstacles entering the cordon area and (b) without penalty. Black circles represent payload, escorts, and obstacles. White circle in (a) represents cordon area.

is considered. To demonstrate this, we test the impact of the loss or addition of an escort while the payload is navigating. The experiment starts with 4 escorts protecting the payload in an environment with 90 obstacles. 25 seconds into the episode, one of the escorts is taken out of the environment, and 25 seconds later it is placed back (within observation range of the payload). The results are shown via the change in cordon breach reward (averaged over 1000 episodes), Fig. 5. We can observe that when one escort is lost, the remaining escorts perform as well as possible with three escorts (as seen by a loss in reward to the level that is roughly

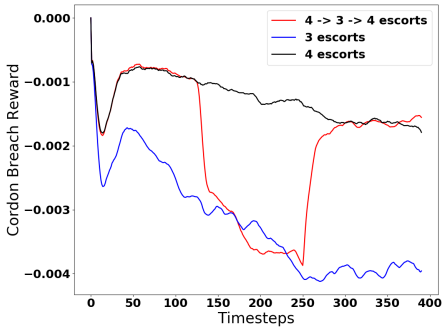


Fig. 5. Average cordon breach reward, r_{cordon}, over the course of successful payload navigation runs in an environment with 90 obstacles. Lines are from runs with 3 deep RL escorts (blue), 4 deep RL escorts (black), and four escorts at the start, a loss to three escorts (at timestep 125), and an increase to four (at timestep 250).

equivalent to the values from a run with three escorts). The converse is true when an additional escort is gained (from three to four escorts), as seen in the increase in reward. This automatic reconfiguration occurs even though there is no explicit communication between escorts.

C. Robustness to Noise

Despite our method being susceptible to the limitations of RL methods as a whole, *i.e.*, learned capabilities corresponding with the training environments, it is still robust to some unseen scenarios. We explore how trained escort teams adapt to scenarios not seen during the training by introducing a disruption or disturbance. We look at three practical disruptions: an unexpected change in obstacle motion dynamics, a change in payload size that represents a reconfiguration of the payload, and sensor noise. The results shown are produced with no retraining.

Fig. 6(a) shows the impact of unexpected disruptions in the social force model, creating unexpected obstacle motion, compared to the baseline case without disruptions, in an environment with 3 deep RL escorts and 50 obstacles. For this scenario, the obstacles experience the change in their velocity in the form $d\boldsymbol{w_k}/dt = \boldsymbol{F}_k(t) + fluctuation$, where $\boldsymbol{F}_k(t)$ is as given in Eq.(1). The $fluctuation$ term is sampled from a normal distribution $\mathcal{N}(\mu = 0, \sigma)$, where σ is varied from $1\,\mathrm{m/s^2}$, $2\,\mathrm{m/s^2}$, $3\,\mathrm{m/s^2}$, to $4\,\mathrm{m/s^2}$. This variation makes the obstacle dynamics increasingly unpredictable. Note that on average, $\boldsymbol{F}_k(t)$ is on the order of $0.75\,\mathrm{m/s^2}$ over an episode. Thus, the social force fluctuation we introduce is large. Furthermore, since the obstacle speed is clipped at the maximum speed of escorts, the fluctuation manifests largely in the stochastically changing *direction* of the obstacle motion. As expected, collision-free payload navigation success drops as the social force fluctuations increase. However, even under moderate noise with $\sigma = 2.0\,\mathrm{m/s^2}$, the escorts successfully defended the payload with 93% of the success rate of the baseline (with no noise).

In Fig. 6(b) we explore the adaptability of the learning policy to a change in the payload. In practice, this change could be a reconfiguration of a multi-body payload to a new conformation, *e.g.*, as mentioned in Sec. I, military

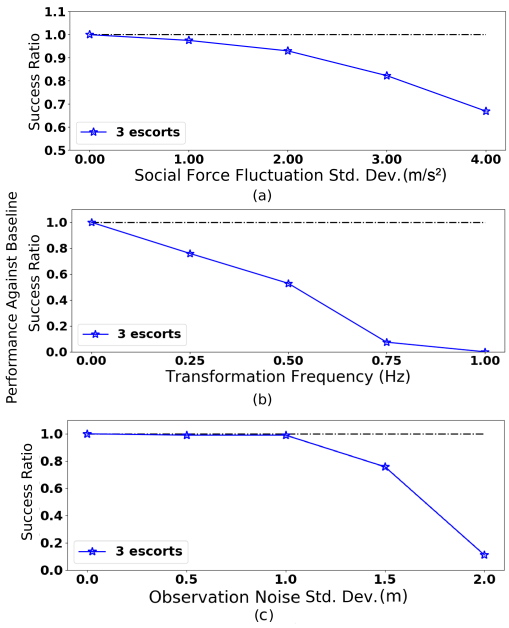


Fig. 6. Success rate expressed as a ratio to a baseline problem of three deep RL escorts and 50 obstacles after the disruptions of: (a) increased standard deviation of $fluctuations$ in the social forces and (b) increased rate of transformation of the payload size and (c) increased observation noise (standard deviation).

convoy re-formation. In order to test this, we implemented a payload of expanding and shrinking radius between 1.5 m and 2.5 m. Frequency (horizontal axis) represents a change in the payload size over time, *i.e.*, a transformation frequency of 0.5 represents a change to a radius of 2.5 m from the original 1.5m in 2 seconds. Results with transformations were tested in an environment with 50 obstacles and three deep RL escorts. While the disruptions result in a loss in success rate, as compared to the same setup without the transforming payload, the escorts show great efficacy still defending the payload while it is navigating. At a frequency as high as 1 Hz, the escorts are unable to complete the task due to the expanding payload overtaking the escorts (98% of runs). This makes sense as as the payload expands to a radius of 2.5 m in one second, which is equivalent to the linear speed of 1 m/s, the maximum speed of the escorts.

In Fig. 6(c), we show the performance of the learning policy in the presence of observation noise, drawn from a Gaussian distribution. Performance holds above 96% of baseline at over 10% noise of the observation range, and diminishes as the noise grows larger.

VI. Conclusion

Defensive escorts help provide critical safety for high-value payloads. Escorts work by coordinating their actions in order to protect the payload and can also be trained to provide a safe cordon area around the navigating payload. Deep RL escorts enhance safety over current solutions in crowded environments, can be robust to several changes including disruptions in the system, changes in payload size, gain and loss of escorts, and provide an end-to-end solution for escort coordination. With only range-limited observations

of the environment and no other explicit information, the escorts learn to automatically coordinate their positions.

VII. Acknowledgements

We would like to thank Lewis Chiang and Adam Yanez for their early help and insights on the safe navigation problem and Evan Carter of the Army Research Lab for helpful suggestions and discussions.

References

[1] H.-T. L. Chiang, B. HomChaudhuri, L. Smith, and L. Tapia, "Safety, challenges, and performance of motion planners in dynamic environments," in *Proc. IEEE Int. Symp. Robot. Res. (ISRR)*, Dec. 2017, pp. 1–16.

[2] X. C. Ding, A. Rahmani, and M. Egerstedt, "Optimal multi-uav convoy protection," in *2009 Second Int. Conf. on Robot Communication and Coordination*, Mar. 2009, pp. 1–6.

[3] X. C. Ding, A. R. Rahmani, and M. Egerstedt, "Multi-uav convoy protection: An optimal approach to path planning and coordination," *IEEE Transactions on Robotics*, vol. 26, no. 2, pp. 256–268, Apr. 2010.

[4] T. Oliveira, A. P. Aguiar, and P. Encarnação, "A convoy protection strategy using the moving path following method," in *2016 Int. Conf. on Unmanned Aircraft Systems (ICUAS)*, Jun. 2016, pp. 521–530.

[5] S. C. Spry, A. R. Girard, and J. K. Hedrick, "Convoy protection using multiple unmanned aerial vehicles: organization and coordination," in *Proc. of the 2005, American Control Conf., 2005.*, vol. 5, Jun. 2005, pp. 3524–3529.

[6] A. Jahn, R. J. Alitappeh, D. Saldaña, L. C. Pimenta, A. G. Santos, and M. F. Campos, "Distributed multi-robot coordination for dynamic perimeter surveillance in uncertain environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 273–278.

[7] D. Saldaña, R. J. Alitappeh, L. C. Pimenta, R. Assunçao, and M. F. Campos, "Dynamic perimeter surveillance with a team of robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 5289–5294.

[8] R. Gayle, W. Moss, M. C. Lin, and D. Manocha, "Multi-robot coordination using generalized social potential fields," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2009, pp. 106–113.

[9] D. Shishika and V. Kumar, "Local-game decomposition for multiplayer perimeter-defense problem," in *57th IEEE Conf. on Decision and Control (CDC)*, Dec. 2018, pp. 2093–2100.

[10] J. Paulos, S. W. Chen, D. Shishika, and V. Kumar, "Decentralization of multiagent policies by learning what to communicate," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2019, pp. 7990–7996.

[11] M. Chen, Z. Zhou, and C. J. Tomlin, "A path defense approach to the multiplayer reach-avoid game," in *53rd IEEE Conf. on Decision and Control (CDC)*, Dec. 2014, pp. 2420–2426.

[12] M. Chen, Z. Zhou, and C. J. Tomlin, "Multiplayer reach-avoid games via low dimensional solutions and maximum matching," in *2014 American Control Conf.*, Jun. 2014, pp. 1444–1449.

[13] H. Raslan, H. Schwartz, and S. Givigi, "A learning invader for the "guarding a territory" game," *J. Intelligent and Robotic Sys.*, vol. 83, no. 1, pp. 55–70, Jul. 2016.

[14] C. Analikwu and H. Schwartz, "Multi-agent learning in the game of guarding a territory," *Int. J. of Innovative Computing, Information and Control*, vol. 13, pp. 1855–1872, 2017.

[15] K.-H. Hsia and J.-G. Hsieh, "A first approach to fuzzy differential game problem: Guarding a territory," *Fuzzy Sets and Systems*, vol. 55, no. 2, pp. 157–167, Apr. 1993.

[16] H. Huang, J. Ding, W. Zhang, and C. J. Tomlin, "A differential game approach to planning in adversarial scenarios: A case study on capture-the-flag," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 1451–1456.

[17] L. Liang, F. Deng, Z. Peng, X. Li, and W. Zha, "A differential game for cooperative target defense," *Automatica*, vol. 102, pp. 58–71, 2019.

[18] H. U. Sheikh, M. Razghandi, and L. Boloni, "Learning distributed cooperative policies for security games via deep reinforcement learning," in *2019 IEEE 43rd Annual Computer Software and Applications Conf. (COMPSAC)*, vol. 1. IEEE, Jul. 2019, pp. 489–494.

[19] F. Fang, A. X. Jiang, and M. Tambe, "Optimal patrol strategy for protecting moving targets with multiple mobile resources," in *Proc. of Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2013, pp. 957–964.

[20] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: The MIT Press, 1998.

[21] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, pp. 4282–4286, 1995.

[22] A. Garg, H. L. Chiang, S. Sugaya, A. Faust, and L. Tapia, "Comparison of deep reinforcement learning policies to formal methods for moving obstacle avoidance," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 3534–3541.

[23] P. Y. Glorennec and L. Jouffe, "Fuzzy q-learning," in *Proc. of Int. Fuzzy Systems Conf.*, Jul. 1997, pp. 659–662.

[24] X.-S. Wang, Y.-H. Cheng, and J.-Q. Yi, "A fuzzy actor-critic reinforcement learning network," *Information Sciences*, vol. 177, no. 18, pp. 3764–3781, 2007.

[25] P. Hernandez-Leal, B. Kartal, and M. E. Taylor, "A survey and critique of multiagent deep reinforcement learning," *Autonomous Agents and Multi-Agent Systems*, vol. 33, no. 6, pp. 750–797, 2019.

[26] A. Faust, K. Oslund, O. Ramirez, A. Francis, L. Tapia, M. Fiser, and J. Davidson, "PRM-RL: Long-range robotic navigation tasks by combining reinforcement learning and sampling-based planning," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 5113–5120.

[27] H.-T. L. Chiang, J. Hsu, M. Fiser, L. Tapia, and A. Faust, "RL-RRT: Kinodynamic motion planning via learning reachability estimators from RL policies," *Robot. and Automat. Lett.*, vol. 4, pp. 4298–4305, 2019.

[28] J. K. Gupta, M. Egorov, and M. Kochenderfer, "Cooperative multi-agent control using deep reinforcement learning," in *Proc. of Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS)*, May 2017, pp. 66–83.

[29] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Nashua, NH, USA: Athena Sci., 2005.

[30] S. Kayukawa, K. Higuchi, J. a. Guerreiro, S. Morishima, Y. Sato, K. Kitani, and C. Asakawa, "BBeep: A sonic collision avoidance system for blind travellers and nearby pedestrians," in *Proc. of the Computer Human Int. Conf. on Human Factors in Computing Systems*, May 2019, pp. 52:1–52:12.

[31] C. Boutilier, "Planning, learning and coordination in multiagent decision processes," in *Proc. of the 6th Conf. on Theoretical Aspects of Rationality and Knowledge*, ser. TARK '96. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1996, pp. 195–210.

[32] R. S. Sutton, D. McAllester, S. Singh, and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation," in *Proc. of Neural Information Processing Systems*, Nov. 1999, pp. 1057–1063.

[33] V. R. Konda, Tsitsiklis, and J. N., "On actor-critic algorithms," *SIAM J. Control Optim.*, vol. 42, no. 4, pp. 1143–1166, 2003.

[34] J. Schulman, P. Moritz, S. Levine, M. I. Jordan, and P. Abbeel, "High-dimensional continuous control using generalized advantage estimation," in *Proc. of Int. Conf. on Learning Representations*, Apr. 2017, pp. 1–14.

[35] J. Schulman, S. Levine, P. Moritz, M. I. Jordan, and P. Abbeel, "Trust region policy optimization," in *In Int. Conf. on Machine Learning (ICML)*, Jul. 2015, pp. 1–9.

[36] J. N. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in *AAAI Conf. on Artificial Intelligence*, Feb. 2017, pp. 2974–2982.

[37] E. Liang, R. Liaw, R. Nishihara, P. Moritz, R. Fox, K. Goldberg, J. Gonzalez, M. Jordan, and I. Stoica, "RLlib: Abstractions for distributed reinforcement learning," in *Proc. of Int. Conf. on Machine Learning*, Jul. 2018, pp. 3053–3062.

[38] Y. A. LeCun, P. G. Haffner, L. Bottou, and Y. Bengio, *Object Recognition with Gradient-Based Learning*. Heidelberg, Berlin, Germany: Springer, 1999.

[39] O. Tange, "GNU parallel - the command-line power tool," *;login: The USENIX Magazine*, vol. 36, no. 1, pp. 42–47, Feb. 2011.