2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Competitive Multi-Team Behavior in Dynamic Flight Scenarios

Tim Seyde[1], Mathias Lechner[1], Joshua Rountree[2] and Daniela Rus[1]

***Abstract*— Efficiently learning strategic multi-agent behavior remains a challenge for robotic systems deployed in real-world scenarios, especially when considering underactuated or dynamically unstable systems. Such systems demand an integrated approach that informs long-term strategic planning with constraints imposed by reactive control, and vice versa, to effectively accomplish task objectives in competitive scenarios. In this paper, we introduce a hierarchical control model to address this: a high-level controller synthesizes strategic guidance from aggregated team experiences, while a low-level controller formulates corresponding task-specific continuous controls. We apply this concept to coordination of competitive multi-team behavior in dynamic flight scenarios with F-16 aircraft. This work introduces a hierarchical reinforcement learning approach for multi-agent coordination, leveraging decoupled distributional value representations at the high-level together with goal-conditioned policy learning at the low-level, providing a control structure that integrates long-horizon strategic planning with short-horizon dynamic control. We further provide a parallel simulator for efficient learning with multi-agent F-16 dynamics.**

I. INTRODUCTION

Learning efficient control strategies through interaction presents a challenging task for dynamically unstable systems. Simply maintaining stability is further often insufficient as resulting motions are to be directed towards a task objective. Particularly in multi-agent settings that are prevalent in real-world robotics deployment, this induces complex decision making across multiple levels of abstraction. However, there tends to be a common dichotomy of developing either single-agent controllers under complex dynamics or multi-agent planners under simplified dynamics. Integrating the two provides unique opportunities for unlocking synergies between strategic long-horizon planning and dynamic short-horizon control, while bringing us closer towards scenarios that can become challenging for humans, as exemplified by cognitive overload arising in e.g. real-world flight control [1].

We consider mixed cooperative-competitive multi-agent control tasks for teams of F-16 aircraft. The agents cooperate within and compete across teams in a zero-sum racing task that provides sparse reward feedback based on team rank. The F-16 dynamics provide high maneuverability and require responsive low-level control at high operating speeds, making them an excellent target for automated control. The sparse team-centric task requires agents to not only focus on short-term continuous control but also reason about long-term strategic interaction across multiple team members.

[1]Tim Seyde, Mathias Lechner and Daniela Rus are with the Computer Science and Artificial Intelligence Laboratory (CSAIL), Massachusetts Institute of Technology, USA. [2] Joshua Rountree is with the Department of the Air Force - MIT AI Accelerator, USA {tseyde,mlechner,rountree,rus}@mit.edu

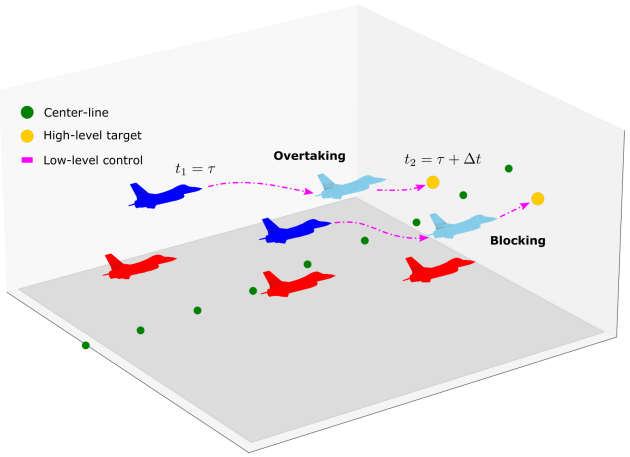

Fig. 1: A competitive multi-agent flight scenario with two teams of F-16 aircraft (red vs. blue) racing along a reference path (green center-line). The task combines long-horizon strategic planning and short-horizon reactivate control, where we leverage hierarchical policies to introduce an abstraction layer between team-centric strategy and ego-centric control.

We approach this problem via hierarchical abstraction spanning multiple levels of control. The high-level controller learns to provide long-term strategic guidance to the low-level controller in the form of goal states by aggregating experience across team members. The low-level controller then generates continuous controls to achieve the high-level goals within the provided time-frame while respecting task constraints. An inner-loop stabilizing controller further translates learned low-level signals to control surface commands. This yields a hierarchical controller that enables coordinated multi-team behavior for complex dynamical systems. Our work leverages the high-fidelity dynamics model provided by the Air Force Research Laboratory in [2]. We further adapt this model to enable data-efficient learning via parallelization across both environments and agents in JAX [3]. In order to fully leverage parallel learning, we further introduce the Decoupled Distributional Expected SARSA agent to coordinate multi-team behavior at the high-level. In summary, our work makes the following key contributions:

- A hierarchical policy structure introducing a layer of abstraction between strategic high-level planning and reactive low-level control for dynamic multi-agent flight.
- A novel reinforcement learning agent for high-level multi-agent control, combining distributional value functions with decoupled control in Expected SARSA.
- A parallel multi-agent F-16 dynamics simulator in JAX.

II. RELATED WORK

A. Multi-agent simulation

A broad spectrum of simulators has been developed to model various aspects of learning multi-agent interaction, ranging from cooperative particle environments [4], [5], [6] over competitive two-player games [7], [8] to mixed cooperative-competitive teams of articulated robots [9], [10]. Recent successes have been demonstrated largely for strategic decision making in competitive games, including one-vs-one board games [7], [8] as well as many-vs-many real-time strategy games [11], [12], where the StarCraft Multi-Agent Challenge emerged as a prevalent benchmark ([13], [14]). While these settings often require long-horizon strategic planning they typically abstract away low-level system dynamics, and there are very few multi-agent environments that consider the interplay between high-level strategy and low-level execution [9], [15], [10], [16]. However, this intricate relationship becomes increasingly important when moving towards deployment in real-world multi-agent scenarios, where underactuated or dynamically unstable systems dictate constraints on what high-level planning may achieve [10], [17]. In this work, we are particularly interested in the intersection of long-horizon decision making and high-speed maneuverability required for multi-agent flight control. There exists a multitude of simulators dedicated to quadrotor dynamics [18], [19] with a subset offering multi-agent learning interfaces [20], [21]. Similarly, multi-agent fixed-wing applications have recently gained prominence in the context of competitive one-vs-one control of F-16 aircraft, with approaches differing in the fidelity of the underlying dynamics model [22], [2], [23]. Efficient learning control of high-fidelity multi-agent interaction requires high data throughput and only a small subset of simulators feature complex dynamics while leveraging vectorization for deployment on hardware accelerators [24], [25], [15], [26], [27]. Here, we study multi-agent flight control in mixed cooperative-competitive scenarios and provide a novel parallelized multi-agent fixed-wing simulation environment in JAX [3].

B. Multi-agent learning

We consider a mixed cooperative-competitive learning scenarios with two teams of agents optimizing for a sparse zero-sum reward [28], [29], [15]. There exists a broad spectrum of multi-agent learning approaches for such tasks, ranging from independent to fully centralized learning. Fully-centralized learners scale poorly with action space dimensionality while independent learners quickly run into coordination challenges, with distributed agents aiming to strike a balance between the two [30], [31], [32], [33], [34], [35]. A common framework for coordinating multi-agent teams is Centralized Training with Decentralized Execution (CTDE), where privileged information is shared during the training phase to align decentralized policies acting under partial observability. In the context of Q-learning based approaches, value decomposition into factored utility functions has been successfully applied across many tasks including decoupled learning of single-agent control [36], [37], [38], [39]. These concepts further extend to the actor-critic setting [40], [41], [42]. Applications in multi-agent fixed-wing flight control typically focus on the competitive one-vs-one scenario and do not require coordination across team members [43], [17], or alternatively solve teaming scenarios via optimization over a selection of predefined low-level behaviors [44]. Recent work has seen an emergence of hierarchical control solutions that aim to provide a layer of abstraction between high-level planning and low-level actions, both in the context of single-agent control [45], [43], [46] as well as multi-agent coordination [44], [47], [15]. Here, we consider a hierarchical policy structure that leverages the CTDE paradigm to coordinate multi-agent team strategy at the high-level via a distributional decoupled variation of Expected SARSA, while optimizing low-level policies that translate high-level guidance into continuous control via Proximal Policy Optimization (PPO).

III. METHOD

In the following, we first introduce the underlying dynamics model in Section III-A and proposed control structure in Section III-B. We then provide technical details regarding the high-level and low-level learning algorithms in Section III-C.

A. F-16 dynamics

We leverage the F-16 aircraft dynamics presented in [2]. The model has 6 degrees of freedom (DoF) with 16 state variables. We follow a notation similar to [2] and represent the aircraft position in a north-east-down coordinate system as (p_n, p_e, p_a). The associated linear velocity is defined via the airspeed v, angle of attack α, and side-slip angle β. The orientation is given by the roll, pitch, and yaw angles (γ, θ, ψ), respectively, with their associated rates $(\dot{\gamma}, \dot{\theta}, \dot{\psi})$. The model further explicitly accounts for the engine power lag Δe. These plant states are augmented by states of an inner-loop controller to yield a "fly-by-wire" system. While throttle (δ_t) is directly commanded by the pilot, the primary control surfaces $(\delta_a, \delta_e, \delta_r)$ are only indirectly controlled via reference commands for the upward acceleration N_z, stability roll rate $\dot{\gamma}_s$, and sum of the side acceleration and yaw rate $N_{y+\dot{\psi}}$. The inner-loop controller then leverages two decoupled Linear-Quadratic Regulators (LQR) along the longitudinal and lateral modes in combination with integral tracking control. Additional details are provided in [2].

We adapt the implementation of the dynamics and inner-loop controller to JAX [3] to enable parallel deployment on hardware accelerators. In particular, we consider both parallelization across n_b environments and n_a agents within environments to yield state tensors with dimensions $[n_b, n_a, ...]$. This facilitates efficient parallel sample generation for downstream learning applications. Our implementation supports both the look-up table model by Stevens [48] as well as the polynomial interpolation model by Morelli [49], where we use the latter dynamics model for learning controllers. We do not adjust the original parameters of the inner-loop controller and consider inner-loop behavior to be part of the dynamics.

B. Control structure

Based on the definitions introduced in the previous section we formulate the underlying state $\boldsymbol{s}$ and action $\boldsymbol{u}$ vectors as

$$\boldsymbol{s} = [v, \alpha, \beta, \gamma, \theta, \psi, \dot{\gamma}, \dot{\theta}, \dot{\psi}, p_n, p_e, p_a, \Delta e, N_z, \dot{\gamma}_s, N_{y+\dot{\psi}}],$$
$$\boldsymbol{u} = [\delta_t, N_{z,ref}, \dot{\gamma}_{s,ref}, N_{y+\dot{\psi},ref}],$$

where the reference side acceleration and yaw rate is set to zero in accordance with [2]. The inner-loop controller is therefore provided with three reference actions from the outer-loop controller, of which the thrust command is fed through to the plant and the reference upward acceleration and stability roll rate are translated to the control surfaces.

We formulate the outer-loop controller as a hierarchical policy. This introduces a layer of abstraction between high-level strategic planning and low-level continuous control [15]. The high-level policy learns to propose multi-step goal states to the low-level policy. At the high-level, we leverage categorical variables in 3D space to encode explicit mode-switching behavior, e.g. *"Agent 1: fall back and roll to the left; ..."*, as an instance of multi-agent option learning [50], [51]. At the low-level, we generate continuous commands to achieve individual goal states within the provided time horizon while satisfying constraints, serving as references for the inner-loop controller as visualized in Figure 2. This controller design effectively decouples team-centric strategic reasoning from ego-centric motion planning to facilitate efficient behavior learning in multi-agent settings, while enabling the low-level reactive control to deviate from high-level guidance if local constraints require so.

C. Learning algorithms

The high-level and low-level policies are updated iteratively within a bilevel optimization. We provide details of the respective learning algorithms in the following, where the underlying observation and action space are defined as

$$\boldsymbol{o}^t_{\text{HL}} = [\overline{\boldsymbol{s}}^t, \boldsymbol{a}^{t-1}_{\text{LL}}, \boldsymbol{o}^t_{\text{ado}}, \boldsymbol{o}^t_{\text{task}}], \quad \boldsymbol{a}^t_{\text{HL}} = [\Delta p_n, \Delta p_e, \Delta p_a],$$
$$\boldsymbol{o}^t_{\text{LL}} = [\overline{\boldsymbol{s}}^t, \boldsymbol{a}^t_{\text{HL}}, \boldsymbol{o}^t_{\text{ado}}, \boldsymbol{o}^t_{\text{task}}], \quad \boldsymbol{a}^t_{\text{LL}} = [\delta_t, N_{z,ref}, \dot{\gamma}_{s,ref}],$$

with $\overline{\boldsymbol{s}}$ as a transform of the original state vector replacing angles γ with $[\sin(\gamma), \cos(\gamma)]$, $\boldsymbol{o}_{\text{ado}}$ providing relative spatial position and velocity readings of other agents, and $\boldsymbol{o}_{\text{task}}$ conferring task-specific information. We note that the low-level policy is conditioned on the current high-level actions. In the following, we denote learnable high-level parameters by subscripts ϕ and low-level parameters by subscripts φ.

a) High-level control: The high-level policy is represented with Categorical variables obtained as an ϵ-greedy evaluation of a state-action value function $Q_\phi(\boldsymbol{o}_{\text{HL}}, \boldsymbol{a}_{\text{HL}})$. We leverage a variation of Decoupled Expected SARSA [39], [15] to enable Centralized Training with Decentralized Execution (CTDE) in cooperative multi-agent settings. This features value decomposition ([52]) of the state-action value function across both action dimension as well as team members. We consider two variations of our approach with either a deterministic or distributional value function. In the deterministic case we have the linear value decomposition

$$Q_\phi(\boldsymbol{o}_{\text{HL}}, \boldsymbol{a}_{\text{HL}}) = \frac{1}{3n_a} \sum_{i=1}^{n_a} \sum_{j=1}^{3} Q^{i,j}_\phi(\boldsymbol{o}^i_{\text{HL}}, \boldsymbol{a}^{i,j}_{\text{HL}}), \tag{1}$$

where $Q^{i,j}_\phi$ denotes a utility function of agent i and high-level action dimension j (target offsets in 3d space $\Delta p_{\{n,e,a\}}$). In the distributional setting, we consider the C51 critic from [53] and proceed with a linear composition in probability space over the value bins ("atoms"), z^k, such that

$$p_\phi(\boldsymbol{o}_{\text{HL}}, \boldsymbol{a}_{\text{HL}}) = \frac{1}{3n_a} \sum_{i=1}^{n_a} \sum_{j=1}^{3} p^{i,j}_\phi(\boldsymbol{o}^i_{\text{HL}}, \boldsymbol{a}^{i,j}_{\text{HL}}), \tag{2}$$

where $p^{i,j}$ now denotes the atom probabilities corresponding to agent i and action dimension j. The set of atoms $\{z^k = v_{\min} + k\frac{v_{\max}-v_{\min}}{K-1}\}$ consists of $K = 51$ potential return values that evenly divide the range between return bounds $\{v_{\min}, v_{\max}\}$. The expected value is recovered in combination with the predicted atom probabilities as

$$Q_\phi(\boldsymbol{o}_{\text{HL}}, \boldsymbol{a}_{\text{HL}}) = \sum_{k=1}^{51} z^k p^k_\phi(\boldsymbol{o}_{\text{HL}}, \boldsymbol{a}_{\text{HL}}). \tag{3}$$

The value function is then optimized on the temporal difference (TD) error between itself and the target y_t. We note that the high-level control loop runs at a lower frequency than the low-level control and we therefore accumulate rewards over the low-level time-horizon Δt. The target value is then $y_t = \sum_{\tau=t}^{t+\Delta t} r(\boldsymbol{s}^\tau, \boldsymbol{a}^\tau_{\text{LL}}) + \gamma Q_\phi(\boldsymbol{o}^{t+\Delta t}_{\text{HL}}, \boldsymbol{a}^{t+\Delta t}_{\text{HL}})$, where we minimize TD errors via the Huber loss, L_δ. In both the deterministic and distributional formulation, the training proceeds centralized across team members to discover multi-agent strategies, while execution proceeds decentralized via ϵ-greedy evaluation of the agent-specific utility functions.

b) Low-level control: The low-level policy receives goal states from the high-level policy and generates continuous actions based on a multi-variate Gaussian distribution. This action-selection is decentralized across each agent and implicitly considers strategic teaming behavior via the provided goal state distribution. We leverage Proximal Policy Optimization (PPO) [54] as an effective on-policy actor-critic algorithm in parallel simulation settings. In particular, we employ the common clipped objective with likelihood ratio $r^t(\varphi) = \frac{\pi_\varphi(\boldsymbol{a}^t_{\text{LL}}|\boldsymbol{o}^t_{\text{LL}})}{\pi_{\varphi'}(\boldsymbol{a}^t_{\text{LL}}|\boldsymbol{o}^t_{\text{LL}})}$ and advantage function $\hat{A}^t$, where $\pi_{\varphi'}$ denotes a previous version of the network parameters φ. We omit the TD value function objective and entropy loss to avoid clutter, and provide the clipped policy objective as

$$L^{\text{CLIP}}_{\text{PPO}} = \min(r^t(\varphi)\hat{A}^t, \text{clip}(r^t(\varphi), 1-\epsilon, 1+\epsilon)\hat{A}^t). \tag{4}$$

c) Bi-level optimization: The overall optimization then alternates between improving single-agent low-level control under fixed high-level target distributions, and optimizing multi-agent high-level targets under fixed low-level controls:

$$\min_\phi \mathbb{E}_{a_{\text{HL}} \sim Q_\phi, \tau_{\text{LL}} \sim \pi_{\hat{\varphi}}} L_\delta(y_t - Q_\phi(o_{\text{HL}}, a_{\text{HL}})) \tag{5}$$
$$\text{s.t. } \hat{\varphi} = \arg\max_\varphi \mathbb{E}_{a_{\text{HL}} \sim Q_\phi, \tau_{\text{LL}} \sim \pi_{\hat{\varphi}}} L^{\text{CLIP}}_{\text{PPO}}.$$

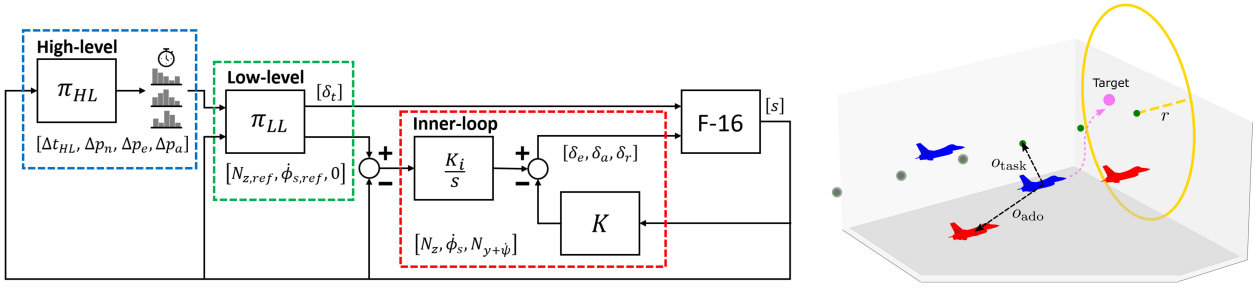

Fig. 2: Left: a schematic of our hierarchical policy design. The high-level module (blue) learns goal state predictions based on environment observations and relays these to the low-level module (green) with a time-horizon for goal completion. The low-level policy is conditioned on high-level goals and predicts thrust commands as well as reference actions that an inner-loop controller (red) translates to control surface commands [2]. This three-layer hierarchy introduces layers of abstraction between long-horizon team-centric planning (HL), medium-horizon ego-centric control (LL), and hardware-level short-horizon system stabilization. Right: scene featuring a blue team competing against a red team. The agents observe center-line waypoints (green) and constraint volumes (yellow) over a preview horizon together with relative spatial information of other agents. Each agent predicts high-level goal states (magenta, sphere) to be achieved via low-level control (magenta, line).

IV. EXPERIMENTS

In the following, we first provide an overview of the multi-agent teaming task together with details on the network architectures and hyperparameters in Sections IV-A and IV-B. We then provide both quantitative as well as qualitative experimental results comparing our proposed approach to baselines ablating on individual components in Section IV-C.

A. Task description

The environment features two teams of four F-16 aircraft competing in a zero-sum racing task. The racetrack is defined by a center-line that interpolates waypoints placed randomly in 3D space under smoothness constraints. Additionally, each waypoint has an associated radius that defines a variable-width tube within which agents need to operate. A visual representation of the environment is provided in Figure 2 (schematic) and Figure 5 (trajectory). Apart from their ego state information ($\overline{s}^t$, Section III-C), agents observe task information o_{task} in the form of center-line waypoints with their associated constraint radius over a limited preview horizon. Lastly, relative position and velocity information of other agents together with their respective team IDs are provide as o_{ado}. This requires only information from standard on-board TCAS/IFF systems [55], and does not assume privileged access to internal opponent states. All position-based information is transformed into spherical coordinates as we found this to significantly accelerate learning. Agent-specific termination conditions are introduced that check multi-agent collisions, force agents to stay within altitude bounds and enforce realistic turn-rates. This is particularly important as we simulate agile multi-agent maneuvers at velocities of over 1000 km/h. The high-level reward function provides sparse zero-sum team-rank reward, where team-rank λ^T represents the rank of the leading team member. It further features a zero-sum regularization term at an order of magnitude lower that encourages improvements in ego-centric rank λ^E. The high-level reward is thus defined as $r_t^{\text{HL}} = (1 - 2\lambda_t^T) + 0.1(\lambda_{t-1}^E - \lambda_t^E)/(1 + \min(\lambda_{t-1}^E, \lambda_t^E))$. The low-level policy receives a proximity reward for being close to the current high-level waypoint together with penalty terms for agent collisions and leaving the track boundaries. We further introduce small regularization terms that encourage smoothness of the controls and angular rates. Our multi-agent simulator extends the single agent system-dynamics of [2], [26] and enables rollouts of 4-vs-4 team races across thousands of parallel environments on hardware accelerators via tensorized representations in JAX [3]. We will release our simulator upon publication to enable broader access to time-efficient learning of dynamic multi-team behaviors.

B. Learning architecture

The high-level critic as well as low-level actor-critic are implemented as two-layer fully-connected networks. We enable permutation invariance with respect to observations of other agents by pre-processing o_{ado} with a multi-headed cross-attention encoder that treats ego information as queries and other agent's observations as key-value pairs [56], [15]. Further hyperparameters are provided in Table I. The high-level policy predicts position offsets to a target point on the center-line at the end of the preview horizon via a 7-bin Categorical along each spatial dimension, based on an ϵ-greedy evaluation of the associated state-action value function as outlined in Section III-C. The high-level decision making proceeds at a lower rate than the low-level control and we consider two time-horizons, updating every 8s or 20s. The low-level policy computes the associated continuous controls for achieving high-level targets at a constant rate of 12.5Hz. We note that the high-level policy provides strategic guidance to the low-level policy, while the latter retains the flexibility to deviate from the proposed target if required.

C. Empirical evaluation

We train multi-agent control policies on mixed cooperative-competitive racing tasks in our parallel simulator via self-play. To this end, we maintain a pool

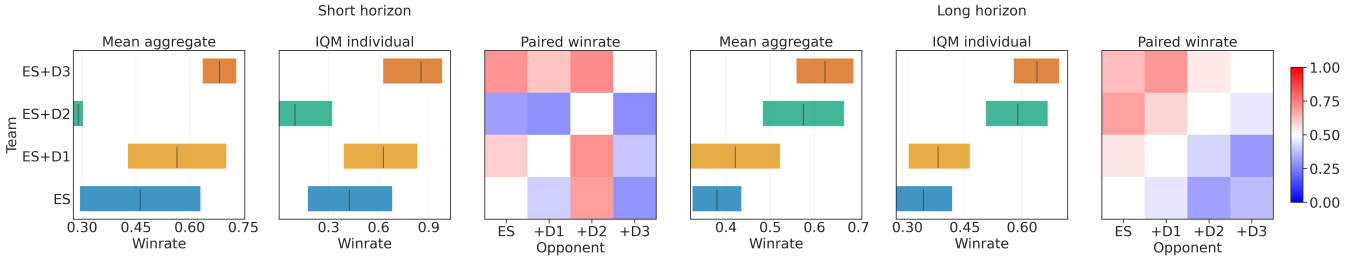

Fig. 3: Evaluation of learned teaming behavior based on round-robin winrates across 5×5 seed pairings yielding 12800 races per agent. We vary the time-scale of the high-level decision making and consider $\Delta t_{\text{HL}} = 8$s (panels 1-3) and $\Delta t_{\text{HL}} = 20s$ (panels 4-6). We compare our proposed ES+D3 agent to ablations including removal of the distributional value function (ES+D2), decoupling across team-members (ES+D1) and no decoupling (ES). We provide mean total winrate per agent with standard deviation (panels 1 and 4), interquartile mean winrate among pairings with 95% confidence intervals (panels 2 and 5), and mean winrate of individual pairings (panels 3 and 6). We observe significant benefit of learning distributionally decoupled value representations (ES+D3) in coordinating multi-team racing behavior, outperforming both ablations as well as the DecSARSA baseline across a total of 38400 evaluation races with randomized track configuration and starting positions.

TABLE I: Training hyperparameters.

Parameter	Value	Parameter	Value
n_{env}	1024	h_{dim}	$[64, 64]$
γ	0.99	$n_{\text{PPO, step}}$	32
$n_{\text{ES, step}}$	10	$n_{\text{PPO, mini}}$	4
$n_{\text{ES, bin}}$	7	$n_{\text{PPO, epoch}}$	4
ϵ_{ES}	0.1	λ_{GAE}	0.95

of past checkpoints and either play against a copy of our current policy or a former checkpoint with equal likelihood. We abbreviate Expected SARSA as ES and denote our factored distributional agent as ES+D3, which subsumes agents that ablate on individual features as follows

- **ES + D3** Distributional factored critic (Eq. 3)
- **ES + D2** Factored critic across agents (Eq. 1)
- **ES + D1** Factored critic across actions
- **ES** Expected SARSA

where the latter two do not aggregate information across team members and can be regarded as egoistic agents.

We train all agents with the same self-play setup for 1 million training steps across 1024 parallel environments. We then compare performance quantitatively between agents in a round-robin tournament, where each agent plays every other agent for 12800 races each. Figure 3 provides empirical results for high-level updating every $\Delta t_{\text{HL}} = 8$ seconds (panels 1-3) and every $\Delta t_{\text{HL}} = 20$ seconds (panels 4-6). We provide mean total winrate with standard deviation (panels 1 and 4), interquartile mean winrate among pairings with 95% confidence intervals (panels 2 and 5) [57], and mean winrate of individual pairings (panels 3 and 6). The evaluations in Figure 3 indicate very strong performance of our ES+D3 agent, which achieves winrates of around 70%. This highlight that learning decoupled value functions across both action dimensions and team members in combination with distributional value representations can yield highly capable multi-agent systems. Furthermore, we observe improved general performance over the non-distributional agent ES+D2 across all evaluations. The latter especially struggles on the short horizon task instance, which could indicate that aggregating learning signals across team members under increased perceived randomness resulting from more fine-grained updates could require distributional approaches to stabilize optimization in expectation. The pairing-specific winrate statistics, comparing ego (row) to opponent (column) teams, in panels 3 and 6 further indicates that the distributional agent outperforms all opponents (top row). In the long-horizon scenario, a clear hierarchy across system ablations emerges that underlines the benefit of individual features.

Lastly, we provide qualitative results that showcase the emergence of multi-team coordination based on sample trajectories from our simulator in Figures 4 and 5. In Figure 4, the blue team (agents 0-3) and red team (agents 4-7) all start with the same forward displacement and randomized lateral displacement as well as altitude. Here, agent 0 and 1 immediately roll to the right (top row) in order to push agent 5 out-of-bounds and block agent 6 and 7, respectively. This enables agent 3 to take the lead for the blue team, while agent 2 moves into the center and limits the space for the red team to attack from below. The interaction trajectories of agents 0 and 5 (left) as well as 1 and 7 (right) are highlighted for visual clarity. The multi-team scenario allows for the emergence of strategic interactions. In Figure 5, the blue team starts in front with agents 6 and 7 (red) attempting to challenge agents 0 and 1 (blue) for the lead in frame 1. Agents 0 and 1 (blue) execute a coordinated blocking maneuver by rolling to the right and cutting off agents 6 and 7 (red) in frame 2. Agent 0 (blue) effectively forces agent 7 (red) to fall back by pushing it outside the constraint perimeter (yellow), while agent 6 (red) evades the block by changing course and rolling to the left in frame 3. While agents 0 and 2 (blue) attempt to close the gap between them, agent 6 (red) squeezes through in frame 4, then immediately rolls right in frame 5 to pass agent 1 (blue) on the outside and take the lead. This qualitative trajectory highlights the emergence of strategic multi-step decision making both from the perspective of cooperation via e.g. coordinated blocking and line adjustment, as well as cross-team competition via agile evasion and aggressive overtaking. While our proposed

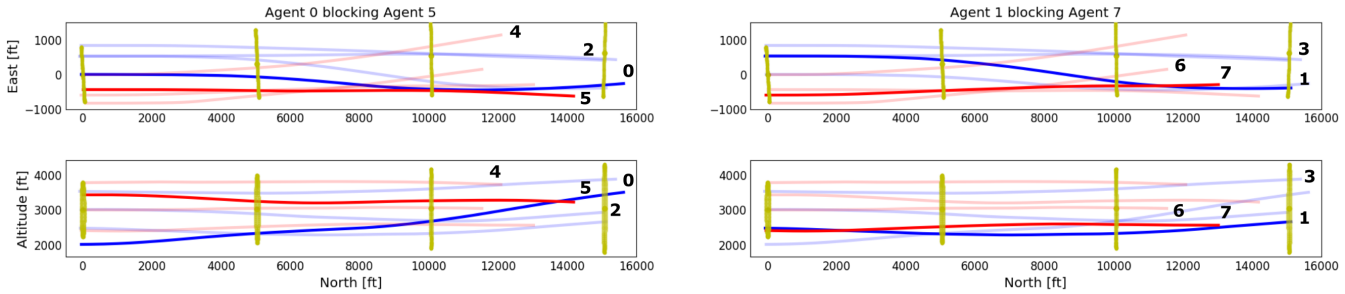

Fig. 4: Sample trajectory highlighting a multi-agent coordinated blocking maneuver executed by the blue team (agents 0-3). Forward displacement is provided on the x-axis with lateral displacement (top) and altitude (bottom) on the y-axis. We highlight two interactions in the same sequence separately for visual clarity. Left: agent 0 (blue) rolls to the right, pushing agent 5 (red) out-of-bounds (yellow). Right: agent 1 (blue) similarly rolls to the right and blocks both agents 6 and 7 (red). Agent 3 takes the lead, while agent 2 moves into the middle reducing the space for overtakes from below.

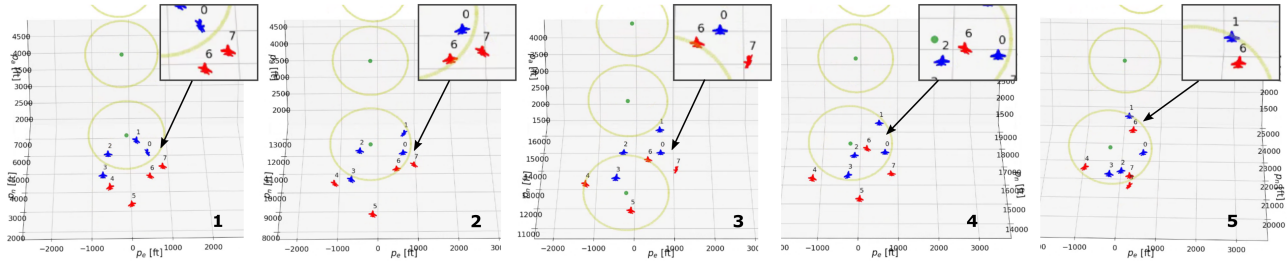

Fig. 5: Sample scenario highlighting the emergence of strategic cooperative and competitive behavior. Frame 1: agents 6 and 7 (red) attempt to challenge agent 1 (blue) for the lead. Frame 2: agents 0 and 1 (blue) roll to the right in a blocking maneuver. Frame 3: agent 0 (blue) forces agent 7 (red) to fall back by pushing it outside the constraint perimeter (yellow), while agent 6 (red) evades the block by rolling left. Frame 4: agents 0 and 2 (blue) attempt to close the gap but agent 6 (red) squeezes through. Frames 5: agent 6 (red) cuts back to the right and eventually overtakes agent 1 (blue) for the lead. Our approach enables the emergence of strategic high-level behavior complementing reactive low-level control, an instance of which is provided by the competitive multi-step interactions between agents 6-7 (red) and 0-2 (blue).

approach is capable of learning behaviors that successfully solve the task with high winrates compared to baseline approaches, it further enables the emergence of strategic high-level behavior complementing reactive low-level control.

V. CONCLUSIONS

Learning competitive multi-agent control in dynamic real-world settings requires close integration of long-horizon strategic planning with short-horizon reactive control. We study efficient coordination and behavior acquisition in mixed cooperative-competitive settings in the context of multi-team flight control with F-16 dynamics. Our approach leverages hierarchical reinforcement learning, introducing a layer of abstraction between team-centric high-level strategy and ego-centric low-level control. At the high-level, we aggregate multi-agent information within a decoupled distributional critic to predict long-horizon goal state distributions for each team member. At the low-level, we train a goal-conditioned policy that aims to achieve the high-level goals within the given time-horizon, while retaining flexibility to negotiate trade-offs between global strategy and local constraints. The hierarchical policy is trained iteratively via bilevel optimization, learning multi-agent high-level behaviors with our proposed Decoupled Distributional Expected SARSA agent. To accelerate multi-agent learning in competitive teaming scenarios, we further introduce a tensorized JAX-based version of established F-16 dynamics that provide parallelization across environments and agents to exploit the efficiency of hardware accelerators. We leverage these dynamics to formulate a competitive multi-team racing task that focuses on emergent multi-agent behavior from sparse rank rewards. Our empirical evaluations underscore the benefits of our hierarchical policy structure and decoupled distributional high-level learning framework, both with respect to raw quantitative winrates, as well as the qualitative emergence of high-level strategies such as coordinated blocking, actively forcing evasive maneuvers, or aggressive overtaking.

There are several promising directions future work, including explicitly comparing the emergent low-level maneuvers arising from end-to-end hierarchies with learned options over hand-crafted expert maneuvers. While we focused on a specific scenario in our study, extensions could further consider a broader set of tasks such as multi-agent surveillance scenarios as well as coordination among heterogeneous agent classes [58]. Our current implementation also considers fixed high-level horizons across all agents for efficient parallelization of high-level training, while observation conditioned timings could broaden the set of emergent capabilities.

ACKNOWLEDGMENT

Research was in part sponsored by the United States Air Force Research Laboratory and the Department of the Air Force Artificial Intelligence Accelerator and was accomplished under Cooperative Agreement Number FA8750-19-2-1000. The views and conclusions 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 Department of the Air Force or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

REFERENCES

[1] T. Deveans and R. Kewley, "Overcoming information overload in the cockpit," *Operations Research Center of Excellence, West Point, NY. ORCEN Technical report: DSE-TR-0904. Available from Defence Technical Information Center website www. dtic. mil with ID: ADA506356*, 2009.

[2] P. Heidlauf, A. Collins, M. Bolender, and S. Bak, "Verification challenges in f-16 ground collision avoidance and other automated maneuvers." in *ARCH@ ADHS*, 2018, pp. 208–217.

[3] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, "JAX: composable transformations of Python+NumPy programs," 2018. [Online]. Available: http://github.com/google/jax

[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] L. Zheng, J. Yang, H. Cai, M. Zhou, W. Zhang, J. Wang, and Y. Yu, "Magent: A many-agent reinforcement learning platform for artificial collective intelligence," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[6] M. Bettini, R. Kortvelesy, J. Blumenkamp, and A. Prorok, "Vmas: a vectorized multi-agent simulator for collective robot learning," *arXiv preprint arXiv:2207.03530*, 2022.

[7] J. Schrittwieser, I. Antonoglou, T. Hubert, K. Simonyan, L. Sifre, S. Schmitt, A. Guez, E. Lockhart, D. Hassabis, T. Graepel, *et al.*, "Mastering atari, go, chess and shogi by planning with a learned model," *Nature*, vol. 588, no. 7839, pp. 604–609, 2020.

[8] J. Perolat, B. De Vylder, D. Hennes, E. Tarassov, F. Strub, V. de Boer, P. Muller, J. T. Connor, N. Burch, T. Anthony, *et al.*, "Mastering the game of stratego with model-free multiagent reinforcement learning," *Science*, vol. 378, no. 6623, pp. 990–996, 2022.

[9] K. Kurach, A. Raichuk, P. Stanczyk, M. Zajkac, O. Bachem, L. Espeholt, C. Riquelme, D. Vincent, M. Michalski, O. Bousquet, *et al.*, "Google research football: A novel reinforcement learning environment," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 4501–4510.

[10] T. Haarnoja, B. Moran, G. Lever, S. H. Huang, D. Tirumala, M. Wulfmeier, J. Humplik, S. Tunyasuvunakool, N. Y. Siegel, R. Hafner, *et al.*, "Learning agile soccer skills for a bipedal robot with deep reinforcement learning," *arXiv preprint arXiv:2304.13653*, 2023.

[11] O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, *et al.*, "Grandmaster level in starcraft ii using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, 2019.

[12] C. Berner, G. Brockman, B. Chan, V. Cheung, P. Debiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, *et al.*, "Dota 2 with large scale deep reinforcement learning," *arXiv preprint arXiv:1912.06680*, 2019.

[13] M. Samvelyan, T. Rashid, C. S. de Witt, G. Farquhar, N. Nardelli, T. G. J. Rudner, C.-M. Hung, P. H. S. Torr, J. Foerster, and S. Whiteson, "The StarCraft Multi-Agent Challenge," *CoRR*, vol. abs/1902.04043, 2019.

[14] B. Ellis, S. Moalla, M. Samvelyan, M. Sun, A. Mahajan, J. N. Foerster, and S. Whiteson, "Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning," 2022. [Online]. Available: https://arxiv.org/abs/2212.07489

[15] P. Werner, T. Seyde, P. Drews, T. M. Balch, I. Gilitschenski, W. Schwarting, G. Rosman, S. Karaman, and D. Rus, "Dynamic multi-team racing: Competitive driving on 1/10-th scale vehicles via learning in simulation," in *7th Annual Conference on Robot Learning*, 2023.

[16] R. J. Torbati, S. Lohiya, S. Singh, M. S. Nigam, and H. Ravichandar, "Marbler: An open platform for standardized evaluation of multi-robot reinforcement learning algorithms," in *2023 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 2023, pp. 57–63.

[17] J. Chai, W. Chen, Y. Zhu, Z.-X. Yao, and D. Zhao, "A hierarchical deep reinforcement learning framework for 6-dof ucav air-to-air combat," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2023.

[18] W. Guerra, E. Tal, V. Murali, G. Ryou, and S. Karaman, "FlightGoggles: Photorealistic sensor simulation for perception-driven robotics using photogrammetry and virtual reality," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, nov 2019.

[19] Y. Song, S. Naji, E. Kaufmann, A. Loquercio, and D. Scaramuzza, "Flightmare: A flexible quadrotor simulator," in *Proceedings of the 2020 Conference on Robot Learning*, 2021, pp. 1147–1157.

[20] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," in *Field and Service Robotics*, 2017. [Online]. Available: https://arxiv.org/abs/1705.05065

[21] J. Panerati, H. Zheng, S. Zhou, J. Xu, A. Prorok, and A. P. Schoellig, "Learning to fly—a gym environment with pybullet physics for reinforcement learning of multi-agent quadcopter control," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021.

[22] K. Weiren, Z. Deyun, K. Zhang, and Y. Zhen, "Air combat autonomous maneuver decision for one-on-one within visual range engagement base on robust multi-agent reinforcement learning," in *2020 IEEE 16th International Conference on Control & Automation (ICCA)*. IEEE, 2020, pp. 506–512.

[23] J. Berndt, "Jsbsim: An open source flight dynamics model in c++," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2004, p. 4923.

[24] O. So, P. Drews, T. Balch, V. Dimitrov, G. Rosman, and E. A. Theodorou, "Mpogames: Efficient multimodal partially observable dynamic games," 2022.

[25] C. Lu, J. Kuba, A. Letcher, L. Metz, C. Schroeder de Witt, and J. Foerster, "Discovered policy optimisation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 16 455–16 468, 2022.

[26] O. So and C. Fan, "Solving stabilize-avoid optimal control via epigraph form and deep reinforcement learning," *arXiv preprint arXiv:2305.14154*, 2023.

[27] A. Rutherford, B. Ellis, M. Gallici, J. Cook, A. Lupu, G. Ingvarsson, T. Willi, A. Khan, C. S. de Witt, A. Souly, *et al.*, "Jaxmarl: Multi-agent rl environments in jax," *arXiv preprint arXiv:2311.10090*, 2023.

[28] L. Busoniu, R. Babuska, and B. De Schutter, "A comprehensive survey of multiagent reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 38, no. 2, pp. 156–172, 2008.

[29] S. Gronauer and K. Diepold, "Multi-agent deep reinforcement learning: a survey," *Artificial Intelligence Review*, pp. 1–49, 2022.

[30] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proceedings of the tenth international conference on machine learning*, 1993, pp. 330–337.

[31] C. Claus and C. Boutilier, "The dynamics of reinforcement learning in cooperative multiagent systems," *AAAI/IAAI*, vol. 1998, p. 2, 1998.

[32] L. Matignon, G. J. Laurent, and N. Le Fort-Piat, "Independent reinforcement learners in cooperative markov games: a survey regarding coordination problems," *The Knowledge Engineering Review*, vol. 27, pp. 1–31, 2012.

[33] J. G. Schneider, W.-K. Wong, A. W. Moore, and M. A. Riedmiller, "Distributed value functions," in *ICML*, 1999.

[34] S. J. Russell and A. Zimdars, "Q-decomposition for reinforcement learning agents," in *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 2003, pp. 656–663.

[35] B. Wang, J. Xie, and N. Atanasov, "Darl1n: Distributed multi-agent reinforcement learning with one-hop neighbors," in *2022 IEEE/RSJ*

International Conference on Intelligent Robots and Systems (IROS). IEEE, 2022, pp. 9003–9010.

[36] T. Rashid, M. Samvelyan, C. Schroeder, G. Farquhar, J. Foerster, and S. Whiteson, "Qmix: Monotonic value function factorisation for deep multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2018, pp. 4295–4304.

[37] K. Son, D. Kim, W. J. Kang, D. E. Hostallero, and Y. Yi, "Qtran: Learning to factorize with transformation for cooperative multi-agent reinforcement learning," in *International Conference on Machine Learning*. PMLR, 2019, pp. 5887–5896.

[38] A. Tavakoli, M. Fatemi, and P. Kormushev, "Learning to represent action values as a hypergraph on the action vertices," *arXiv preprint arXiv:2010.14680*, 2020.

[39] T. Seyde, P. Werner, W. Schwarting, I. Gilitschenski, M. Riedmiller, D. Rus, and M. Wulfmeier, "Solving continuous control via q-learning," in *The Eleventh International Conference on Learning Representations*, 2022.

[40] Y. Wang, B. Han, T. Wang, H. Dong, and C. Zhang, "Dop: Off-policy multi-agent decomposed policy gradients," in *International Conference on Learning Representations*, 2020.

[41] J. Su, S. Adams, and P. A. Beling, "Value-decomposition multi-agent actor-critics," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, 2021, pp. 11 352–11 360.

[42] B. Peng, T. Rashid, C. S. de Witt, P.-A. Kamienny, P. Torr, W. Boehmer, and S. Whiteson, "FACMAC: Factored multi-agent centralised policy gradients," in *Advances in Neural Information Processing Systems*, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., 2021. [Online]. Available: https://openreview.net/forum?id=wZYWwJvkneF

[43] A. P. Pope, J. S. Ide, D. Mićović, H. Diaz, D. Rosenbluth, L. Ritholtz, J. C. Twedt, T. T. Walker, K. Alcedo, and D. Javorsek, "Hierarchical reinforcement learning for air-to-air combat," in *2021 international conference on unmanned aircraft systems (ICUAS)*. IEEE, 2021, pp. 275–284.

[44] H. Shin, J. Lee, H. Kim, and D. H. Shim, "An autonomous aerial combat framework for two-on-two engagements based on basic fighter maneuvers," *Aerospace Science and Technology*, vol. 72, pp. 305–315, 2018.

[45] M. Wulfmeier, A. Abdolmaleki, R. Hafner, J. T. Springenberg, M. Neunert, T. Hertweck, T. Lampe, N. Siegel, N. Heess, and M. Riedmiller, "Compositional transfer in hierarchical reinforcement learning," *arXiv preprint arXiv:1906.11228*, 2019.

[46] R. Reiter, J. Hoffmann, J. Boedecker, and M. Diehl, "A hierarchical approach for strategic motion planning in autonomous racing," *arXiv preprint arXiv:2212.01607*, 2022.

[47] R. S. Thakkar, A. S. Samyal, D. Fridovich-Keil, Z. Xu, and U. Topcu, "Hierarchical control for cooperative teams in competitive autonomous racing," *arXiv preprint arXiv:2204.13070*, 2022.

[48] B. L. Stevens, F. L. Lewis, and E. N. Johnson, *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[49] E. A. Morelli, "Global nonlinear parametric modelling with application to f-16 aerodynamics," in *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, vol. 2. IEEE, 1998, pp. 997–1001.

[50] R. S. Sutton, D. Precup, and S. Singh, "Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning," *Artificial intelligence*, vol. 112, no. 1-2, pp. 181–211, 1999.

[51] J. Chakravorty, N. Ward, J. Roy, M. Chevalier-Boisvert, S. Basu, A. Lupu, and D. Precup, "Option-critic in cooperative multi-agent systems," *arXiv preprint arXiv:1911.12825*, 2019.

[52] P. Sunehag, G. Lever, A. Gruslys, W. M. Czarnecki, V. Zambaldi, M. Jaderberg, M. Lanctot, N. Sonnerat, J. Z. Leibo, K. Tuyls, *et al.*, "Value-decomposition networks for cooperative multi-agent learning based on team reward," in *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, 2018, pp. 2085–2087.

[53] M. G. Bellemare, W. Dabney, and R. Munos, "A distributional perspective on reinforcement learning," in *International conference on machine learning*. PMLR, 2017, pp. 449–458.

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

[55] M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos, "Next-generation airborne collision avoidance system," Massachusetts Institute of Technology-Lincoln Laboratory Lexington United States, Tech. Rep., 2012.

[56] S. Iqbal and F. Sha, "Actor-attention-critic for multi-agent reinforcement learning," in *International conference on machine learning*. PMLR, 2019, pp. 2961–2970.

[57] R. Agarwal, M. Schwarzer, P. S. Castro, A. Courville, and M. G. Bellemare, "Deep reinforcement learning at the edge of the statistical precipice," *Advances in Neural Information Processing Systems*, 2021.

[58] M. Lechner, L. Yin, T. Seyde, T.-H. J. Wang, W. Xiao, R. Hasani, J. Rountree, D. Rus, *et al.*, "Gigastep-one billion steps per second multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, 2024.