2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

CrazyMARL: Decentralized Direct Motor Control Policies for Cooperative Aerial Transport of Cable-Suspended Payloads

Viktor Lorentz[1], Khaled Wahba[1], Sayantan Auddy[1], Marc Toussaint[1,2], and Wolfgang Hönig[1,2]

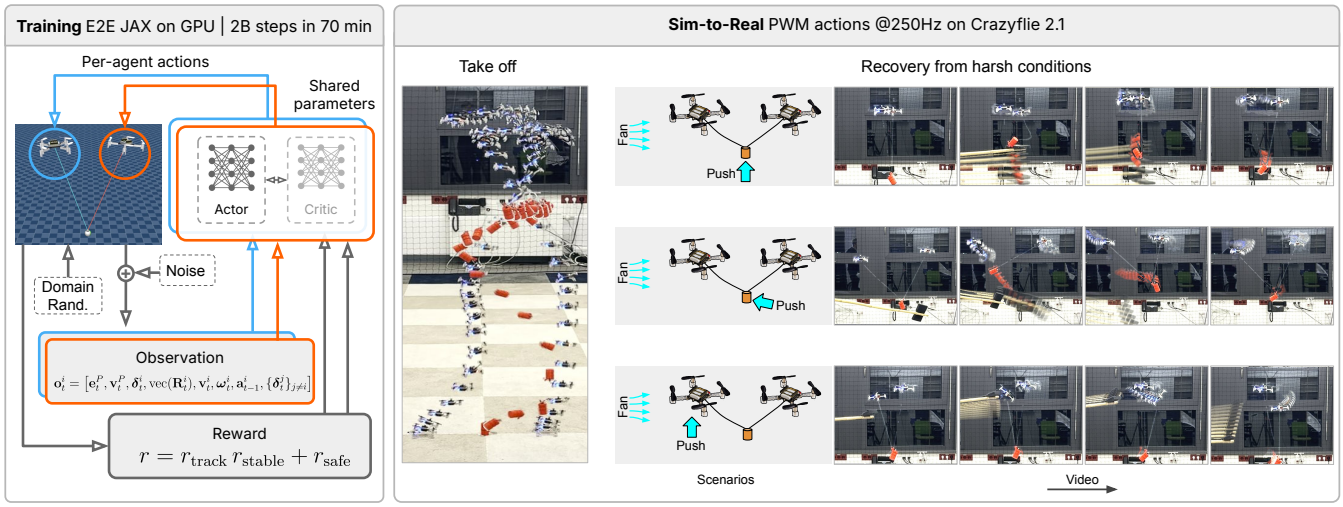

Fig. 1. Overview of our approach. (Left panel) Training in Mujoco with end-to-end JAX and domain randomization. IPPO with shared parameters maps each per agent observation $o_t^i = [\, e_t^P, v_t^P, \delta_t^i, \mathrm{vec}(R_t^i), v_t^i, \omega_t^i, a_{t-1}^i, \{\delta_t^j\}_{j\neq i}\,]$ to motor commands under a reward that combines tracking, stability, and safety. The local observation packs payload error and velocity, the quadrotors' relative position, orientation, and velocities, its last action, and for coordination, the relative positions of teammates. (Right panel) Sim-to-Real on the Crazyflie 2.1 robot, where the same decentralized policy runs on two quadrotors at 250 Hz and outputs direct PWM. The sequences show autonomous takeoff and recovery scenarios of a cable-suspended payload under strong pushes and wind (shown with cyan-colored arrows) with an average speed of 3.5 m/s. Video frames progress from left to right.

***Abstract*— Collaborative transportation of cable-suspended payloads by teams of Unmanned Aerial Vehicles (UAVs) has the potential to enhance payload capacity, adapt to different payload shapes, and provide built-in compliance, making it attractive for applications ranging from disaster relief to precision logistics. However, multi-UAV coordination under disturbances, nonlinear payload dynamics, and slack–taut cable modes remains a challenging control problem. To our knowledge, no prior work has addressed these cable mode transitions in the multi-UAV context, instead relying on simplifying rigid-link assumptions. We propose *CrazyMARL*, a decentralized Reinforcement Learning (RL) framework for multi-UAV cable-suspended payload transport. Simulation results demonstrate that the learned policies can outperform classical decentralized controllers in terms of disturbance rejection and tracking precision, achieving an 80% recovery rate from harsh conditions compared to 44% for the baseline method. We also achieve successful zero-shot sim-to-real transfer and demonstrate that our policies are highly robust under harsh conditions, including wind, random external disturbances, and transitions between slack and taut cable dynamics. This work paves the way for autonomous, resilient UAV teams capable of executing complex payload missions in unstructured environments. Code and videos can be found on the website: `https://imrclab.github.io/CrazyMARL`.**

Corresponding author: `viktor.lorentz@hhi.fraunhofer.de`
[1]Technische Universität Berlin, Berlin, Germany.
[2]Robotics Institute Germany (RIG), Germany.
This work was supported by the German Federal Ministry of Research, Technology and Space (BMFTR) under the Robotics Institute Germany (RIG), and the Deutsche Forschungsgemeinschaft (DFG) - 448549715.

I. Introduction

Unmanned Aerial Vehicles (UAVs) have moved from research prototypes to widely deployed tools in civil and industrial settings, including disaster response, agriculture, logistics, and inspection [1], [2]. Among emerging applications, collaborative cable-suspended payload transport is compelling: cables are lightweight, adaptable to irregular loads, and introduce compliance that attenuates vibration. However, multi-UAV transport of a shared suspended payload is challenging due to pendulum dynamics, tension coupling, contact events, and disturbances.

There are two well-established paradigms to tackle control in robotics: model-based and learning-based methods. Classical model-based controllers offer stability guarantees and well-founded design principles, but struggle with modeling errors and scalability. Centralized coordination can provide optimal solutions but incurs bottlenecks and single points of failure, while decentralized schemes can tackle scalability but may lack performance guarantees [3], [4]. Reinforcement Learning (RL), in contrast, has emerged as a complementary paradigm that can provide adaptivity in scenarios with complex dynamics or incomplete models. It can directly learn control from interaction and improve robustness to unmodeled effects and disturbances. Recent results demonstrate strong performance for single and multi UAV tasks such as agile

flights under challenging conditions [5], [6].

Building on this progress, we leverage RL to address the decentralized control problem of multi-UAV payload transport with hybrid cable dynamics that capture taut–slack transitions. To the best of our knowledge, this is the first use of RL for controlling multiple robots with constrained onboard microcontrollers operating near their actuation limits. Our objectives are to stabilize the suspended payload under external disturbances, manage cable mode switching, and safely distribute forces among the robots without collisions. Fig. 1 illustrates the training pipeline, deployed system, and shows hardware demonstrations with two robots transporting a payload, including robust recovery from harsh conditions and strong disturbance rejection.

We implement our method on the Crazyflie 2.1 research platform, which is widely used for cooperative transport and UAV control studies [7], [8]. The target platform has a low thrust-to-weight ratio of 1.4, which makes operation close to the motor limits much more likely than on other platforms such as high-end racing multirotors. Our method allows each robot to execute its own policy onboard at 250 Hz and relies on local state estimation and relative pose information. The policy output maps directly to motor Pulse Width Modulation (PWM) at high frequency, without relying on cascaded low-level controllers. This enables us to operate close to the actuation limits which is particularly important for agile flights and disturbance rejection.

On the learning side, we leverage high throughput training with large scale parallelized simulation and extensive domain randomization to improve robustness and shorten iteration time. The resulting controller remains fully decentralized with a small computational footprint and no need for inter-robot communication, which simplifies deployment and improves real world performance.

In summary, our main contributions are:

- **End-to-end decentralized Multi-Agent Reinforcement Learning (MARL) for multiple quadrotors carrying a cable-suspended payload.** We train a fully decentralized policy with direct motor PWM commands (no low-level cascades) for cable-suspended payload transport.
- **Empirical evaluation in simulation and hardware.** We validate on Crazyflie 2.1 platform. In wind trials with a mean wind speed of 3.5 m/s the policy maintains stable formations and recovers from large external disturbances.
- **High-throughput simulation.** A GPU-parallelized JAX/MJX pipeline captures cable–payload–robot interactions in contact-rich scenarios, providing a foundation for future aerial manipulation and related tasks.

II. Related Work

We review control strategies for multirotor UAVs transporting cable-suspended payloads, covering classical model-based methods as well as learning-based approaches for both single- and multi-agent coordination. For a comprehensive survey of aerial cable transport, see [4].

A. Traditional Model-based Approaches

Model-based control approaches for aerial payload transport include centralized cascaded geometric controllers that provide stability guarantees for payload and manipulation control [9], as well as decentralized controllers that exploit internal cable tension for quasi-static attitude stabilization [10]. However, these methods rely on noisy payload acceleration as a feedback signal and model cables as rigid rods, which limit the applicability in agile scenarios.

Centralized and decentralized nonlinear model predictive control (NMPC) have advanced multi-UAV payload manipulation without acceleration feedback and can incorporate high-level objectives (e.g., obstacle collision avoidance) [11], [12]. However, centralized NMPC is computationally expensive and suffers when scaled to large teams, while decentralized NMPC depends on iterative inter-robot communication and can suffer from deadlocks. Both approaches still adopt the rigid-rod cable assumption inherited from reactive controllers.

The controllers devised by the previous methods adopt simplified models which restrict control performance to quasi-static regimes. Thus, other works have employed the full system dynamics in offline motion planning through optimization-based planners to account for more accurate models and enable agile maneuver planning [7], [13]–[15]. Nevertheless, despite considering the full dynamics, these approaches still rely on the rigid-rod cable assumption.

To overcome the limitations of the rigid-rod assumption, more accurate cable models have been incorporated by switching the dynamics that explicitly capture the taut–slack transitions [16], [17]. NMPC formulations with such hybrid models enable motions unattainable under rigid-rod assumptions but have so far been limited to single-UAV systems. Extending them to cooperative multi-robot manipulation remains challenging due to the added complexity of modeling, optimization, and coordination.

In contrast, our work explores reinforcement learning (RL) as a promising direction for achieving agile maneuvers and high robustness in multi-UAV payload transport while accounting for realistic cable dynamics.

B. Reinforcement Learning-based Approaches

Early application of RL to UAV control focused on single-agent scenarios, and showed that RL agents trained with model-free RL algorithms perform on par with or better than classical controllers [18]. Later works have demonstrated the effectiveness of RL-trained UAVs in handling harsh initial conditions, executing aggressive maneuvers, and operating near the limits of their dynamic capabilities [19], [20]. Other approaches have extended single-UAV control to payload transport and aerial manipulation, where RL-based controllers have also proven effective in adapting to unknown payload dynamics, maintaining stability and rejecting payload disturbances [21]. Notably, [22] is capable of mode-switching and handling flexible cables for single UAVs. However, compared to the single-UAV-payload-transportation problem, our current work on multi-UAV collaborative payload transport is considerably more challenging due to the coupling between

vehicles, the need for precise coordination to regulate cable tensions, and the heightened risk of collisions.

Multi-Agent Reinforcement Learning (MARL) has emerged as a powerful framework for cooperative tasks. Some MARL approaches employ Centralized Training with Decentralized Execution (CTDE) by using a shared critic, such as [23], [24]. Others, such as Independent Proximal Policy Optimization (IPPO) [25], follow a completely decentralized regime. A decentralized scheme such as IPPO avoids scalability and communication overheads and is thus utilized in our work. Many works have successfully used MARL, particularly methods such as Multi-Agent Proximal Policy Optimization (MAPPO) and IPPO, in real-world, multi-robot collaborative tasks [26], [27]. MARL strategies have also achieved success in swarm coordination, collaborative pursuit and evasion, and obstacle avoidance [8], [28].

In the area of payload transportation, the adaptability and disturbance rejection capabilities of single-UAV methods have naturally led to multi-UAV extensions such as [29]–[31]. However, unlike the centralized approach presented in [29], we adopt a decentralized approach, thereby avoiding communication and scalability overheads. While [31] considers only rigid-link payloads and [30] assumes the payload cables are always taut, we make no such assumptions and model payload links with realistic flexible cables. Neither of these works reports transfer to the real world, whereas we achieve successful zero-shot sim2real transfer and demonstrate the robustness and agility of our robots under harsh real-world conditions. The closest to our current work is the approach from [32], which also uses a decentralized training scheme and showcases robustness under harsh real-world settings. However, unlike our fully decentralized IPPO-based approach, this work uses the CTDE-based MAPPO algorithm. It relies on a low-level controller, whereas we do not, and thanks to our highly parallelized training setup, our training speed is about ten times faster. In addition, [32] still retains the rigid-rod assumption, where our work focuses on exploiting the hybrid cable model to achieve agile maneuvers and recovery from harsh configurations.

III. Method: MARL for Cooperative Payload Transport

We train decentralized MARL policies that enable Q Crazyflie quadrotors to transport a cable suspended payload. The task is modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) $(\mathcal{Q}, \mathcal{S}, \{\mathcal{A}^i\}, P, r, \{\Omega^i\}, O, \rho_0, \gamma)$, where $\mathcal{Q} = \{1, \dots, Q\}$ is the agent index set, $\mathcal{S}$ is the state space, $\mathcal{A}^i = [-1, 1]^4$ is the action space of agent i, P is the transition kernel, r is the shared reward, Ω^i is the local observation space of agent i, O is the observation function, ρ_0 is the reset distribution, and $\gamma \in [0, 1)$ is the discount.

The global state at time t is

$$\mathbf{s}_t = \Big(\{\mathbf{p}_t^i, \mathbf{v}_t^i, \mathbf{R}_t^i, \boldsymbol{\omega}_t^i\}_{i=1}^{Q},\ \mathbf{p}_t^P, \mathbf{v}_t^P \Big), \tag{1}$$

with $\mathbf{p}_t^i \in \mathbb{R}^3$ and $\mathbf{v}_t^i \in \mathbb{R}^3$ the position and linear velocity of agent i, $\mathbf{R}_t^i \in \mathrm{SO}(3)$ its attitude, $\boldsymbol{\omega}_t^i \in \mathbb{R}^3$ its body rates, and $\mathbf{p}_t^P, \mathbf{v}_t^P \in \mathbb{R}^3$ are the payload position and velocity.

Each agent produces a normalized motor command $\mathbf{a}_t^i \in [-1, 1]^4$. We write $\mathbf{a}_t$ for the joint action, the concatenation of $\{\mathbf{a}_t^i\}_{i=1}^{Q}$. The team receives a shared reward $r(\mathbf{s}_t, \mathbf{a}_t)$. The observation spaces are $\{\Omega^i\}$ and the observation function O yields local observations $\mathbf{o}_t^i$ from $\mathbf{s}_t$ as defined below. The initial state is sampled from ρ_0 through the randomized reset distribution. The objective is

$$\max_{\theta}\ J(\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t\, r(\mathbf{s}_t, \mathbf{a}_t) \right], \quad \mathbf{a}_t^i \sim \pi_\theta(\cdot \mid \mathbf{o}_t^i, \mathbf{a}_{t-1}^i). \tag{2}$$

We use IPPO with parameter sharing. A single actor and critic are trained on data from all agents and the critic conditions only on $\mathbf{o}_t^i$ and $\mathbf{a}_{t-1}^i$. Training is centralized by shared parameters and execution is decentralized, following a CTDE paradigm without privileged information. To mitigate partial observability, improve stability, and ease sim-to-real transfer, we augment each local observation with the previous action and encourage smooth commands during training.

A. CrazyMARL Framework

We propose CrazyMARL, an end-to-end JAX-based pipeline that couples MuJoCo MJX with JaxMARL algorithms [33], [34]. It is designed for training coordinated behaviors with multiple quadrotors. We specialize it for use with the Crazyflie research platform carrying a cable-suspended payload, but it can be easily reparameterized for other multirotors and scenarios. MJX allows us to run thousands of environments in parallel on GPU. Our tasks cover single-robot hover/tracking and multi-robot cable-suspended transport with configurable payloads and cables (MuJoCo tendons). We focus on disturbance rejection and hovering under harsh conditions.

B. Observations

We form a global observation

$$\mathbf{o}_t = \big[\mathbf{e}_t^P, \mathbf{v}_t^P, \{\boldsymbol{\delta}_t^i, \mathrm{vec}(\mathbf{R}_t^i), \mathbf{v}_t^i, \boldsymbol{\omega}_t^i, \mathbf{a}_{t-1}^i\}_{i=1}^{Q} \big], \tag{3}$$

where $\mathbf{e}_t^P = \mathbf{p}_{\mathrm{des},t}^P - \mathbf{p}_t^P$, $\boldsymbol{\delta}_t^i = \mathbf{p}_t^i - \mathbf{p}_t^P$, and $\mathrm{vec}(\mathbf{R}_t^i)$ denotes the column vector obtained by stacking the columns of $\mathbf{R}_t^i$. For decentralized execution, agent i observes its local state and the agents' positions relative to the payload.

$$\mathbf{o}_t^i = \big[\mathbf{e}_t^P, \mathbf{v}_t^P, \boldsymbol{\delta}_t^i, \mathrm{vec}(\mathbf{R}_t^i), \mathbf{v}_t^i, \boldsymbol{\omega}_t^i, \mathbf{a}_{t-1}^i, \{\boldsymbol{\delta}_t^j\}_{j \neq i} \big]. \tag{4}$$

During training, we optionally inject scaled Gaussian noise into $\mathbf{o}_t$.

C. Actions

Each agent outputs $\mathbf{a}_t^i \in [-1, 1]^4$. We map to desired thrusts by

$$\mathbf{u}_t^i = \tfrac{\mathbf{a}_t^i + \mathbf{1}}{2} \in [0, 1]^4, \qquad \mathbf{f}_t^{i,\mathrm{cmd}} = \mathbf{u}_t^i\, f_{\max}^i. \tag{5}$$

A first-order lag on a rotor speed proxy approximates non-ideal actuation [35]. Since thrust grows approximately with the square of rotor speed, we define

$$\boldsymbol{\nu}_t^i = \sqrt{\mathbf{f}_t^{i,\mathrm{cmd}}}, \quad \tilde{\boldsymbol{\nu}}_{t+1}^i = \tilde{\boldsymbol{\nu}}_t^i + \alpha^i\big(\boldsymbol{\nu}_t^i - \tilde{\boldsymbol{\nu}}_t^i\big), \quad \alpha^i = \tfrac{\Delta t}{\tau^i}, \tag{6}$$

and apply thrust as

$$\mathbf{f}_t^i = \mathrm{clip}\big((\tilde{\boldsymbol{\nu}}_{t+1}^i)^2,\, 0,\, f_{\max}^i\big). \tag{7}$$

Working in the rotor speed domain aligns the lag more closely with motor and propeller time constants. The filtered speed proxy is initialized near hover and set to zero for grounded starts. In Mujoco the thrust of motor j on vehicle i is applied as an upward force along the body z axis at the motor position and a reaction torque proportional to thrust is added with rotor spin sign and fixed thrust to torque coefficient $k_\tau = 0.006$. All control runs at $250\,\mathrm{Hz}$. On the real hardware, the policy output controls PWM directly without battery voltage compensation. We scale $\mathbf{u}_t^i$ to PWM duty cycle. This is justified because in the operating range of the micro brushed motors, thrust is approximately proportional to duty cycle and in the model, the commanded thrust is proportional to $\mathbf{u}_t^i$. The same normalized action $\mathbf{u}_t^i$ can therefore be interpreted as a normalized thrust command in simulation and as a normalized PWM command on hardware. Direct PWM output avoids the typical thrust mixing step that can lead to motor saturation and therefore improves robustness, especially when operating close to the actuation limits of small lightweight quadrotors with low thrust-to-weight ratio.

D. Simulation and Transition Model

We simulate rigid body dynamics with Mujoco at $250\,\mathrm{Hz}$ with step $\Delta t = 0.004\,\mathrm{s}$. At time t the environment samples the next state from the transition function $s_{t+1} \sim P(\cdot \mid s_t, a_t)$ induced by one Mujoco physics step with our actuation model. It aggregates actuator forces and torques, tendon tension when the cable is taut, contact, and gravity. Each quadrotor is modeled as a freejoint rigid body and connects to the payload body through a Mujoco tendon of length L that exerts tension along the line from payload to robot only when taut, and zero otherwise. Domain randomization parameters and external disturbances enter P at every step. Contacts and friction are solved by the physics engine. This model captures multi-body coupling and the hybrid slack and taut cable modes while remaining fast enough for large scale parallel training.

E. Domain Randomization for Sim-to-Real Transfer

To narrow the reality gap between simulation and reality, we rely on Domain Randomization (DR) techniques. We introduce randomization into several aspects of our training pipeline, including randomized initial states, actuator dynamics parameters, observations, targets, and stochastic disturbances. Our JAX and GPU-based training setup enables training with DR on a large number of environments, resulting in effective zero-shot transfer and robust performance in the real world. Our DR strategy includes the following:

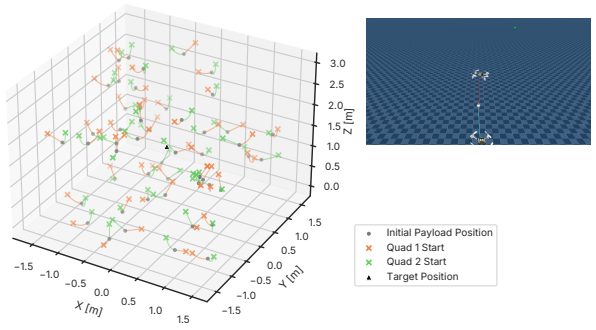

Fig. 2. 50 randomized initial states for $s_0 \sim \rho_0$; harsh cases (slack cables, ground starts) are included. The target is at the center. the top-right shows one state in MuJoCo.

Initial states: The payload is sampled around a nominal target, and quadrotors are placed on a spherical shell clipped by cable length, with randomized attitudes and linear and angular velocities. Challenging cases, such as ground starts and slack cables, are included, as shown in Fig. 2.

Dynamics and actuator parameters We randomize the per motor thrust cap around a quad level base to remain robust under battery discharge and motor aging. For each quad we draw a base thrust from $\mathcal{U}(0.105, 0.15)\,\mathrm{N}$ and add an independent motor offset from $\mathcal{N}(0, 0.008^2)\,\mathrm{N}$, then clip to $[0.095, 0.16]\,\mathrm{N}$. The actuation lag time constants τ^i are sampled in $\mathcal{U}(0.004, 0.05)\,\mathrm{s}$. The filtered thrust state receives a small perturbation at every reset and we inject occasional bounded steps in the filtered RPM proxy. Together, these randomizations support zero-shot sim-to-real transfer despite large variation of motor characteristics in micro brushed motors.

Observations: We inject Gaussian noise into the global observation vector at each timestep. Concretely, given an observation $\mathbf{o}$, we sample a standard normal vector $\boldsymbol{\eta} \sim \mathcal{N}(0, \mathbf{I})$ of the same dimension, and compute the noisy observation using $\mathbf{o}' = \mathbf{o} + \sigma_{\mathrm{obs}}\Lambda\boldsymbol{\eta}$, where σ_{obs} is a tunable noise amplitude and Λ is a diagonal scaling term to tune the noise for each observation component.

Stochastic disturbances and target randomization. At each step, we randomly apply bounded external wrenches. A random UAV receives a small force $f \sim \mathcal{U}(0, 0.05)\mathrm{N}$ and a torque $\tau \sim \mathcal{U}(0, 0.03)\mathrm{Nm}$ in a direction biased toward its body z-axis. The payload receives a small impulse force $f_p \sim \mathcal{U}(0, 5)\mathrm{N}$. We also add occasional jumps in the filtered RPM proxy, and when enabled, bounded random target updates around the goal to improve trajectory following.

F. Reward Design

We propose a modular reward that applies to one or many quadrotors, with or without a cable-suspended payload. It combines tracking, stability, and safety, and uses a bounded exponential envelope to keep gradients smooth [6], [8], [35], [36]. At a low level, the terms promote fast recovery with low swing and tilt, taut cables, safe spacing, and smooth thrust

for sim-to-real transfer. We design reward components to lie in the range $[0, 1]$ wherever possible. This limits trade-off tuning, avoids learning to terminate behavior, and removes the need for a curriculum reward. Our composite reward is defined as

$$r = r_{\text{track}}\, r_{\text{stable}} + r_{\text{safe}}, \tag{8}$$

where r_{track} rewards small payload error and aligns payload velocity with the target direction, and r_{stable} combines terms that cap velocity smoothly, limit payload swing, keep the body-z axis near vertical, and maintain taut cables. The safety component r_{safe} promotes temporal smoothness in actions, balanced motor thrust distribution, and discourages action saturation. It also encourages collision avoidance and distance keeping. The coupling of tracking and stability in (8) encourages high speeds only when the payload is stable, while the additive safety reward applies strict incentives independent of tracking. The same reward structure scales with the number of robots Q, and with minor adjustments, also applies in payload-free scenarios. Full details of the sub-components of the individual rewards in (8) are provided in the Appendix.

G. Training and Policy Architecture

We train decentralized policies with IPPO [25] extending the JaxMARL implementation [34]. All agents share parameters and act independently from local observations. Synchronous vectorized actors collect $N \times T$ transitions per update and we optimize the Proximal Policy Optimization (PPO) objective with generalized advantage estimation, value loss, and an entropy bonus.

The actor is a Multilayer Perceptron (MLP) with three hidden layers $[64, 64, 64]$ with `tanh` activations and a linear mean head. The action distribution is a diagonal Gaussian with learned log standard deviation. The critic is an MLP $[128, 128, 128]$ that maps to a scalar value. We use orthogonal weight initialization with zero bias, sample actions during training, and use the mean at evaluation. In our experiments IPPO is sufficient for up to three quadrotors. For larger teams or stronger partial observability a centralized training variant with a joint critic such as MAPPO may be beneficial.

Hyperparameters were selected with Bayesian optimization over fixed compute budgets. While the pipeline can produce a usable policy in minutes when trained with a small number of environments, we find that with full DR robust performance requires significant parallelization. We therefore use N=16,384 environments, which reduces gradient variance, yields smooth learning curves, and shows little sensitivity to the random seed, while delivering very robust performance under all randomized conditions. Key settings are summarized in Table I. Unless stated otherwise, all reported results use these defaults on a single Nvidia Ada RTX 4000 GPU with 20 GB memory and JAX and MJX parallelization.

TABLE I

KEY TRAINING AND MODEL HYPERPARAMETERS.

Parameter	Setting
Algorithm	IPPO [25]
Actor network	MLP $[64, 64, 64]$, `tanh`
Critic network	MLP $[128, 128, 128]$, `tanh`
Initialization	Orthogonal weights, zero bias
Action distribution	Diagonal Gaussian, learned log std
Rollout	N=16,384 envs, T=128 steps
Optimization	256 minibatches, 8 epochs per update
Learning rate	4×10^{-4}
Entropy coefficient	0.01
Value loss coefficient	0.5
Clip range	0.2
Grad norm clip	0.5
Discount γ	0.997
GAE λ	0.95
Episode length	3072 steps $\approx$ 12.3 s at 250 Hz
Control rate	250 Hz, one sim step per action
Observation noise std	1.0
Action noise std	0.0
Total environment steps	2×10^9
Seed	0
Tuning	Bayesian optimization
Hardware	Single Nvidia Ada RTX 4000, 20 GB

IV. EXPERIMENTS AND RESULTS

This chapter describes the evaluation methodology and outlines the metrics and figures that we used to quantify the performance of our learned policies on both single-quadrotor and multi-quadrotor cable-suspended payload transport tasks.

A. Baseline Comparison

We benchmark the decentralized RL policy against the baseline of [7], which models each cable as a rigid rod and relies on centralized trajectory optimization with an online tracker. This limits responsiveness to cable swing, mode changes, disturbances, and payload variations. In the two quadrotors scenario, the payload begins at $(0, 0, 1.5)$ m and vehicles randomized around it, both pursuing the same goal. We evaluate in simulation for statistical analysis and tightly controlled initial conditions. Each method runs $N = 1000$ trials with randomized payload states and vehicle poses. To favor the baseline we precompute a polynomial trajectory from start to target and have the baseline track it.

Our learned policy achieves 797 out of 1000 successful recoveries $(79.7\,\%)$ with a mean speed of 0.58 m/s. The baseline achieves 435 successful recoveries $(43.5\,\%)$ with a mean speed of 0.27 m/s. Given identical start and goal, the higher mean speed and the higher recovery rate indicates shorter and more reliable recoveries for our method. Qualitatively our policy damps swing quickly and follows a nearly straight approach, whereas the baseline often dips and then spirals, which is consistent with its rigid rod cable model.

B. Generalization

We assess generalization in the two quadrotor payload task by sweeping cable length (L), payload mass, observation noise scale (σ_{obs}), and initialization seed. Success is the fraction of runs that recover within 10 s. As shown in Fig. 4: (i) Cable length: shorter cables sharply reduce success (increased collision risk). Moderate increases above 0.3 m improve success, while very long cables ($>$ 1 m) degrade performance as the payload becomes harder to stabilize. (ii) Payload mass: performance is robust across a broad range; lighter payloads show a slight drop, and heavier payloads reduce success more noticeably. (iii) Observation noise: the policy is tolerant up to a scale of about 1, with a steep decline beyond that. (iv) Seeds:

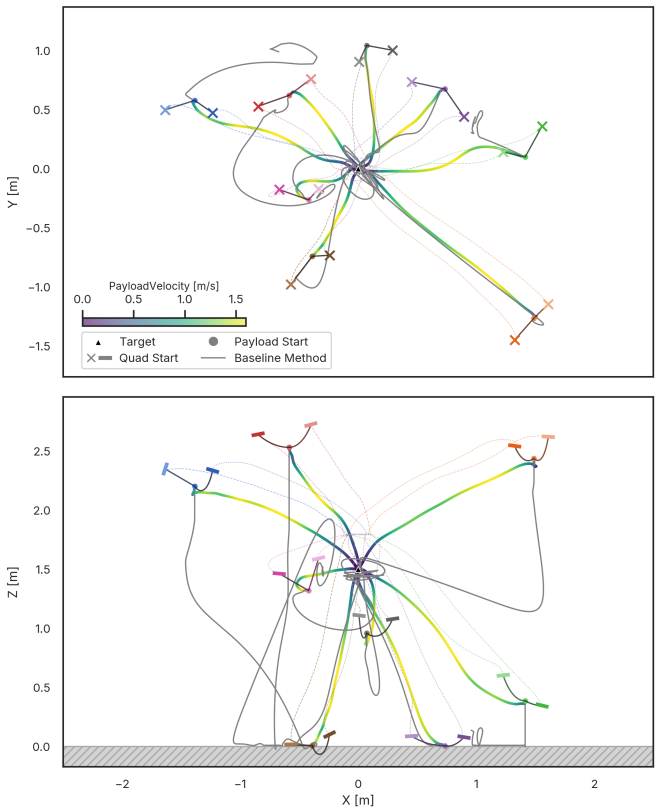

Fig. 3. Example recovery trajectories from eight harsh initializations. Top: XY; bottom: XZ. Our Method with visualized payload velocity (colormap) vs. baseline (thin gray).

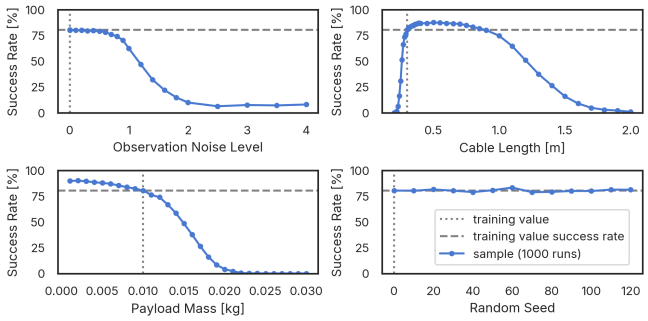

Fig. 4. Evaluation of the learned policy's generalization capabilities in the two quadrotors with payload scenario. Each datapoint represents the percentage of successfully recovered runs out of 1000 runs in an environment that only differs in the specific value adjusted. The policy is trained on a 0.3 m cable length and a payload mass of 0.01 kg.

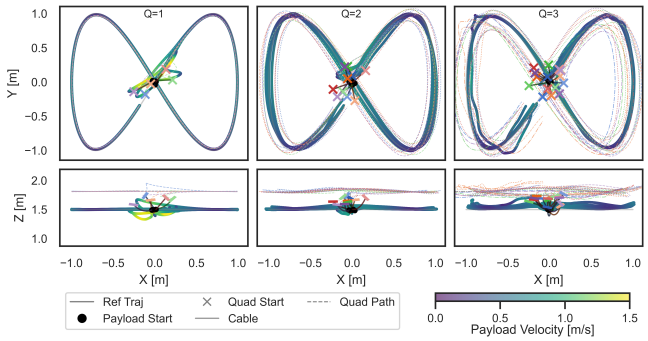

Fig. 5. Figure eight trajectory tracking in simulation with one, two, and three quadrotors carrying a payload. Each column shows five runs starting at harsh initial conditions. Left $Q{=}1$ quadrotors, middle $Q{=}2$, right $Q{=}3$. The solid colormap curve is the payload trajectory with color indicating speed magnitude. Dotted curves are vehicle trajectories.

results are largely insensitive to initialization, consistent with training across 16k parallel environments. Overall, the policy generalizes well to unseen dynamics without explicit domain randomization, indicating strong robustness. Future work could explore randomizing payload weight, shape, and cable length to further expand the operational range.

C. Scalability

Our framework scales from a single vehicle to larger teams. We evaluate $Q \in \{1, 2, 3, 6\}$ on two tasks in simulation, harsh recovery from initialization as shown in Fig. 2 to a fixed target and tracking of a figure eight reference trajectory as shown in Fig. 5. Each setting uses 1000 trials with a policy trained for the given team size.

For $Q{=}1$ the system stabilizes from up to $1\,\mathrm{m}$ displacement in about $2\,\mathrm{s}$ with a 99% success rate and tracks the figure eight with a small phase lag. For $Q{=}2$ recovery succeeds in 81% of trials with a similar settling time and failures occur mainly at the beginning under extreme initial states that cause collisions. For $Q{=}3$ with a $20\,\mathrm{g}$ payload, the team reaches 60% recovery success and shows coordinated load sharing. For $Q{=}6$ with a $40\,\mathrm{g}$ payload, most runs terminate early, which exposes a coordination limit.

The main bottleneck seems to be permutation sensitivity in peer observations. Sorting peers by distance or adding attention-based aggregation and a richer payload representation are promising next steps.

D. Sim-to-Real Transfer

We deploy the learned decentralized policy on Crazyflie quadrotors by exporting it to TFLite and compiling with STM32Cube.AI for the STM32F405. An identical policy runs fully onboard each quadrotor at $250\,\mathrm{Hz}$, without a centralized coordinator or cross-vehicle communication. Each robot builds its observation from the onboard Extended Kalman Filter (EKF) state (position, velocity, attitude, and rates) together with motion-capture positions of the payload and teammates. The policy outputs individual motor commands converted to PWM and sent directly to the motors.

Flight tests demonstrate robust autonomous takeoff, strong disturbance rejection, and stable flight in wind for a single quadrotor (with and without payload) and two quadrotors carrying a payload. In wind trials, the measured average wind speed at the target and figure-eight trajectory midpoint is $3.5\,\mathrm{m/s}$. Figures 1 and 6 show agile behavior, performing a rapid takeoff from the ground to a $1\,\mathrm{m}$ altitude in $2.5\,\mathrm{s}$. Despite wind, the policy maintains a steady-state position-holding Root Mean Square Error (RMSE) of $0.077\,\mathrm{m}$, only a marginal increase over the $0.064\,\mathrm{m}$ RMSE without wind. Figure 1 also illustrates disturbance rejection with two quads and a $10\,\mathrm{g}$ payload, while Fig. 7 demonstrates maintaining a figure-eight payload trajectory under wind.

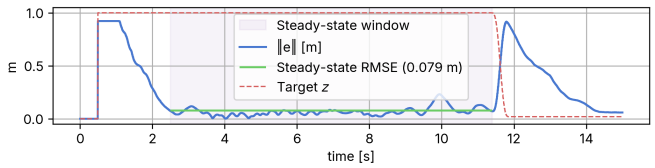

Fig. 6. Payload position error during a 15s flight on the real hardware with takeoff and landing under wind disturbance and 10 g payload.

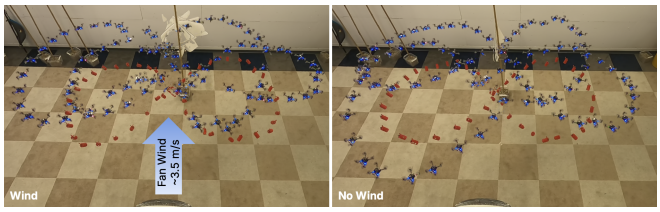

Fig. 7. Two quadrotors with RL policy cooperating to ensure the payload stays on a figure eight trajectory with and without wind disturbance.

V. Conclusion

We present a unified framework for training and deploying reinforcement learning controllers for single and multi-agent cable payload transport while taking into account hybrid cable dynamics. Our end-to-end decentralized policy directly outputs motor-level PWM commands at 250 Hz, enabling deployment on the resource-constrained Crazyflie 2.1 platform, which operates in our test scenarios near its actuation limits due to its low thrust-to-weight ratio. In both simulation and hardware, the approach achieves agile, robust transport, including zero-shot transfer with disturbance rejection and autonomous takeoff, capabilities that are unattainable with the baseline controller. These results highlight the potential of decentralized MARL with motor-level control as a scalable solution for cooperative aerial manipulation.

An interesting avenue for future research is to extend our work to larger multi-UAV teams by adopting order-invariant peer encodings through sorting, a centralized critic, or attention-based aggregation, enabling policies that generalize across team sizes. Additionally, integrating observation history to address partial observability and improve action smoothness, along with safety and obstacle-avoidance mechanisms, would support scalable deployment in cluttered environments.

APPENDIX

A. Reward Function Specifics

Here we describe the components of our reward defined in equation (8) as $r = r_{\text{track}} r_{\text{stable}} + r_{\text{safe}}$. In the following, the mean over agents or motors is denoted by $\text{avg}[\cdot]$. We use a reward shaping function $\Phi_s(x) = \exp(-s\,|x|)$ to shape reward components so that each term is bounded and the derivatives are well-behaved. Further, let $d = \|\mathbf{e}^P\|$ and $g(d) = \min(3d, 1) + c_f$, which helps to shrink the allowed payload speed near the target and keeps a small floor through the use of the scalar constant c_f.

Tracking: The reward for tracking is defined as

$$r_{\text{track}} = \tfrac{1}{2}\big(r_{\text{pos}} + r_{\text{dir}}\big), \tag{9}$$

where

$$r_{\text{pos}} = \Phi\big(\|\mathbf{e}^P\|\big), \quad \mathbf{v}_{\text{dir}} = \frac{\mathbf{v}^P}{\|\mathbf{v}^P\| + \varepsilon}, \quad \mathbf{e}_{\text{dir}} = \frac{\mathbf{e}^P}{\|\mathbf{e}^P\| + \varepsilon},$$
$$r_{\text{dir}} = \Phi_{s_{\text{align}}}\big(1 - \mathbf{v}_{\text{dir}} \cdot \mathbf{e}_{\text{dir}}\big), \quad s_{\text{align}} = \min\big(c_g\,\|\mathbf{e}^P\|,\, c_s\big).$$

Here r_{pos} rewards small payload errors. The direction term aligns the payload velocity with the target direction for a linear trajectory. The factor c_g yields strong guidance when far away, and the cap c_s reduces sharpness near the goal.

Stability: The stability reward is defined as:

$$r_{\text{stable}} = \tfrac{1}{5}\Big(r_{\text{velP}} + r_{\text{velQ}} + \lambda_{\text{yaw}} r_{\text{yaw}} + \lambda_{\text{up}} r_{\text{up}} + r_{\text{taut}}\Big), \tag{10}$$

where

$$r_{\text{velP}} = \exp\left[-\left(\frac{\|\mathbf{v}^P\|}{c_{\text{swing}}\, v_{\max}\, g(d)}\right)^{c_{\text{exp}}}\right],$$
$$r_{\text{velQ}} = \operatorname*{avg}_i\left[\exp\left[-\left(\frac{\|\mathbf{v}^i\|}{v_{\max}\, g(d)}\right)^{c_{\text{exp}}}\right]\right],$$
$$r_{\text{yaw}} = \operatorname*{avg}_i\big[\Phi(\omega_z^i)\big], \qquad r_{\text{up}} = \operatorname*{avg}_i\big[\Phi(\theta^i)\big],$$
$$r_{\text{taut}} = \frac{1}{L}\left(\operatorname*{avg}_i\big[\|\mathbf{p}^i - \mathbf{p}^P\|\big] + \operatorname*{avg}_i\big[p_z^i - p_z^P\big]\right).$$

The velocity term caps the speed smoothly. The exponent c_{exp} gives a soft wall that becomes strict at the boundary. The factor c_{swing} for the payload allows a slightly lower speed than the vehicles to limit swing. The yaw term discourages excessive yaw rate and tilt keeps the body z axis near vertical. The taut term increases radial and vertical separation relative to the payload to keep cables engaged, normalized by cable length L. ω_z^i denotes the yaw rate and θ^i denotes the tilt angle between each quadrotor's body-frame z-axis and the world z-axis. λ_{yaw} and λ_{up} are scaling terms.

Safety: The reward for safety is defined as

$$r_{\text{safe}} = \tfrac{1}{5}\Big(-r_{\text{coll}} - r_{\text{oob}} - \lambda_{\text{s}} r_{\text{smooth}} - r_{\text{energy}} + r_{\text{dist}}\Big), \tag{11}$$

where

$$r_{\text{dist}} = \begin{cases} 1, & Q = 1, \\ \operatorname{avg}_{i \neq j}\left[\operatorname{clip}\left(\frac{\|\mathbf{p}^i - \mathbf{p}^j\| - d_{\min}}{d_{\text{safe}} - d_{\min}}, 0, 1\right)\right], & Q > 1, \end{cases}$$
$$r_{\text{coll}} = c_{\text{coll}}\, \mathbb{I}_{\text{coll}}, \quad r_{\text{oob}} = c_{\text{oob}}\, \mathbb{I}_{\text{oob}}, \quad \bar{a}_t^i = \operatorname*{avg}_j\big[a_{t,j}^i\big],$$
$$r_{\text{smooth}} = \tfrac{1}{2}\left(\operatorname*{avg}_i\big[\|\mathbf{a}_t^i - \mathbf{a}_{t-1}^i\|_1\big] + \operatorname*{avg}_i\big[\|\mathbf{a}_t^i - \bar{a}_t^i \mathbf{1}\|_1\big]\right),$$
$$r_{\text{energy}} = \operatorname*{avg}_i\left[\operatorname*{avg}_j\big[\exp\big(-c_{\text{b}}|a_{t,j}^i|\big) + \exp\big(c_{\text{b}}(a_{t,j}^i - 1)\big)\big]\right].$$

where $d_{\min}, d_{\text{safe}}$ reflect Crazyflie arm span and cable clearance, r_{coll}, r_{oob} denote binary collision and out-of-bounds indicators, and $c_{\text{coll}}, c_{\text{oob}}$ dominate the other bounded positive terms, and strongly discourage unsafe behavior.

Here the left term in r_{smooth} penalizes changes between consecutive actions and enforces temporal smoothness, while the right term penalizes deviations of the four motor commands from their mean and encourages a balanced spatial thrust distribution. The energy barrier r_{energy} uses

the coefficient c_b to softly repel actions near 0 and 1, which discourages saturation while remaining bounded and smooth. We use the following values of the empirically determined scalar constants: $s = 2, c_f = 0.02, c_g = 40, c_s = 2, c_{\text{exp}} = 8, c_{\text{swing}} = 0.75, c_{\text{coll}} = c_{\text{oob}} = 10, c_b = 50$. We set $d_{\min}$ to 0.15m, and d_{safe} to 0.18m. The coefficients λ_i and the cap $v_{\max}$ tune the trade off between agility and caution. The rest of the reward coefficients do not require tuning. For two quadrotors we use λ_{yaw}=10, λ_{up}=5, λ_{s}=10 and $v_{\max} = 1.5m/s$ to get desired robust but agile behavior.

REFERENCES

[1] M. Idrissi, M. Salami, and F. Annaz, "A review of quadrotor unmanned aerial vehicles: applications, architectural design and control algorithms," *Journal of Intelligent & Robotic Systems*, vol. 104, no. 2, p. 22, 2022.

[2] M. Lyu, Y. Zhao, C. Huang, and H. Huang, "Unmanned aerial vehicles for search and rescue: A survey," *Remote Sensing*, vol. 15, no. 13, p. 3266, 2023.

[3] S. Batra, Z. Huang, A. Petrenko, T. Kumar, A. Molchanov, and G. S. Sukhatme, "Decentralized control of quadrotor swarms with end-to-end deep reinforcement learning," in *Conference on Robot Learning*, 2022, pp. 576–586.

[4] J. Estevez, G. Garate, J. M. Lopez-Guede, and M. Larrea, "Review of aerial transportation of suspended-cable payloads with quadrotors," *Drones*, vol. 8, no. 2, p. 35, 2024.

[5] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," *Nature*, vol. 620, no. 7976, pp. 982–987, 2023.

[6] J. Eschmann, D. Albani, and G. Loianno, "Learning to fly in seconds," *IEEE Robotics and Automation Letters*, vol. 9, no. 7, pp. 6336–6343, 2024.

[7] K. Wahba, J. Ortiz-Haro, M. Toussaint, and W. Hönig, "Kinodynamic motion planning for a team of multirotors transporting a cable-suspended payload in cluttered environments," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2024, pp. 12 750–12 757.

[8] Z. Huang, Z. Yang, R. Krupani, B. Şenbaşlar, S. Batra, and G. S. Sukhatme, "Collision avoidance and navigation for a quadrotor swarm using end-to-end deep reinforcement learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 300–306.

[9] K. Sreenath and V. Kumar, "Dynamics, control and planning for cooperative manipulation of payloads suspended by cables from multiple quadrotor robots," in *Robotics: Science and Systems IX*, P. Newman, D. Fox, and D. Hsu, Eds., 2013.

[10] M. Tognon, C. Gabellieri, L. Pallottino, and A. Franchi, "Aerial co-manipulation with cables: The role of internal force for equilibria, stability, and passivity," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2577–2583, 2018.

[11] S. Sun and A. Franchi, "Nonlinear MPC for full-pose manipulation of a cable-suspended load using multiple uavs," in *International Conference on Unmanned Aircraft Systems (ICUAS)*, 2023, pp. 969–975.

[12] N. De Carli, R. Belletti, E. Buzzurro, A. Testa, G. Notarstefano, and M. Tognon, "Distributed NMPC for cooperative aerial manipulation of cable-suspended loads," *IEEE Robotics and Automation Letters*, vol. 10, no. 10, pp. 10 546–10 553, 2025.

[13] K. Wahba and W. Hönig, "pc-dbcbs: Kinodynamic motion planning of physically-coupled robot teams," *IEEE Robotics and Automation Letters*, vol. 10, no. 11, pp. 11 118–11 125, 2025.

[14] Y. Wang, J. Wang, X. Zhou, T. Yang, C. Xu, and F. Gao, "Safe and agile transportation of cable-suspended payload via multiple aerial robots," *arXiv preprint arXiv:2501.15272*, 2025.

[15] S. Sun, X. Wang, D. Sanalitro, A. Franchi, M. Tognon, and J. Alonso-Mora, "Agile and cooperative aerial manipulation of a cable-suspended load," *Science Robotics*, vol. 10, no. 107, p. eadu8015, 2025.

[16] H. Wang, H. Li, B. Zhou, F. Gao, and S. Shen, "Impact-aware planning and control for aerial robots with suspended payloads," *IEEE Transactions on Robotics*, vol. 40, pp. 2478–2497, 2024.

[17] L. F. Recalde, M. Sarvaiya, G. Loianno, and G. Li, "Es-hpc-mpc: Exponentially stable hybrid perception constrained MPC for quadrotor with suspended payloads," *IEEE Robotics and Automation Letters*, vol. 11, no. 1, pp. 266–273, 2025.

[18] W. Koch, R. Mancuso, R. West, and A. Bestavros, "Reinforcement learning for uav attitude control," *ACM Transactions on Cyber-Physical Systems*, vol. 3, no. 2, pp. 1–21, 2019.

[19] Y. Song, A. Romero, M. Müller, V. Koltun, and D. Scaramuzza, "Reaching the limit in autonomous racing: Optimal control versus reinforcement learning," *Science Robotics*, vol. 8, no. 82, p. eadg1462, 2023.

[20] J. Xing, I. Geles, Y. Song, E. Aljalbout, and D. Scaramuzza, "Multi-task reinforcement learning for quadrotors," *IEEE Robotics and Automation Letters*, vol. 10, no. 3, pp. 2112–2119, 2025.

[21] H. Hua, Y. Fang, X. Zhang, and C. Qian, "A new nonlinear control strategy embedded with reinforcement learning for a multirotor transporting a suspended payload," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 2, pp. 1174–1184, 2021.

[22] D. Cao, J. Zhou, X. Wang, and S. Li, "FLARE: Agile flights for quadrotor cable-suspended payload system via reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 11, no. 3, pp. 3653–3660, 2026.

[23] 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.

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

[25] C. S. De Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?" *arXiv preprint arXiv:2011.09533*, 2020.

[26] B. Pandit, A. Gupta, M. S. Gadde, A. Johnson, A. K. Shrestha, H. Duan, J. Dao, and A. Fern, "Learning decentralized multi-biped control for payload transport," in *Conference on Robot Learning*, 2025, pp. 1021–1034.

[27] W.-T. Chen, M. Nguyen, Z. Li, G. N. Sue, and K. Sreenath, "Decentralized navigation of a cable-towed load using quadrupedal robot team via marl," *arXiv preprint arXiv:2503.18221*, 2025.

[28] Z. Zhao, Y. Wan, and Y. Chen, "Deep reinforcement learning-driven collaborative rounding-up for multiple unmanned aerial vehicles in obstacle environments," *Drones*, vol. 8, no. 9, p. 464, 2024.

[29] D. Lin, J. Han, K. Li, J. Zhang, and C. Zhang, "Payload transporting with two quadrotors by centralized reinforcement learning method," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 1, pp. 239–251, 2023.

[30] J. Estevez, J. M. Lopez-Guede, J. del Valle-Echavarri, and M. Graña, "Reinforcement learning based trajectory planning for multi-uav load transportation," *IEEE Access*, 2024.

[31] B. Xu, F. Gao, C. Yu, R. Zhang, Y. Wu, and Y. Wang, "Omnidrones: An efficient and flexible platform for reinforcement learning in drone control," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2838–2844, 2024.

[32] J. Zeng, A. M. Gimenez, E. Vinitsky, J. Alonso-Mora, and S. Sun, "Decentralized aerial manipulation of a cable-suspended load using multi-agent reinforcement learning," in *Conference on Robot Learning*, 2025, pp. 3850–3868.

[33] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *IEEE/RSJ international conference on intelligent robots and systems*, 2012, pp. 5026–5033.

[34] A. Rutherford, B. Ellis, M. Gallici, J. Cook, A. Lupu, G. Ingvarsson Juto, T. Willi, R. Hammond, A. Khan, C. Schroeder de Witt *et al.*, "Jaxmarl: Multi-agent rl environments and algorithms in jax," *Advances in Neural Information Processing Systems*, vol. 37, pp. 50 925–50 951, 2024.

[35] A. Molchanov, T. Chen, W. Hönig, J. A. Preiss, N. Ayanian, and G. S. Sukhatme, "Sim-to-(multi)-real: Transfer of low-level robust control policies to multiple quadrotors," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 59–66.

[36] E. Kaufmann, L. Bauersfeld, and D. Scaramuzza, "A benchmark comparison of learned control policies for agile quadrotor flight," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 10 504–10 510.