ATLAS: Affine Transformation Learning with Action-Space Reduction for Safe Multi-UAV Formation Planning

Anonymous Authors

***Abstract*—This paper proposes a safety-constrained search-to-policy framework for scalable multi-UAV formation planning in affine transformation spaces. Although affine representations provide a compact description of collective formation motion, their discretized planning spaces can still introduce combinatorial decision complexity. First, a structured action-space reduction is introduced to reduce the planning complexity while preserving essential formation maneuvers. Then, geometric safety constraints exclude infeasible affine configurations. Subsequently, an offline beam-search teacher generates high-quality demonstrations to guide a masked Deep Q-Network, which is further refined through reinforcement learning. Finally, the learned policy enables low-latency formation planning without online search and is deployed through a leader–follower coordination architecture. Comprehensive simulations and real-world Crazyflie experiments demonstrate reliable planning under within-family distribution shifts, scalability across different swarm sizes, and low computational demands.**

Note to Practitioners– Coordinating a large UAV team by planning separate motions for every vehicle can become computationally burdensome as the team grows. This work instead plans the motion of the formation as a single deformable geometric object, allowing a compact set of affine variables to command collective translation, rotation, scaling, and shape change. For practical implementation, the proposed ATLAS framework reduces the formation-planning action space from 729 simultaneous affine actions to 13 interpretable transformation primitives, removes geometrically unsafe actions before they reach the learned planner, and uses computationally expensive beam search only during offline training. The resulting learned policy executes without online search and retains the same state and action dimensions as the number of UAVs changes. Experiments with 3–16 UAVs demonstrate nearly unchanged online decision cost, while physical five-Crazyflie experiments show that the resulting formation trajectories can be executed through a leader–follower architecture without retraining the high-level planner for vehicle dynamics. Practitioners should note, however, that formation-level geometric safety does not by itself guarantee collision-free physical execution: tracking errors and follower dynamics reduced inter-UAV separation in hardware and caused collisions in several trials. Consequently, deployment in tighter environments should incorporate conservative separation margins and runtime collision avoidance. Future work will extend the framework to fully three-dimensional planning, dynamic obstacles, dynamics-aware planning, and larger physical swarms.

***Index Terms*—Multi-UAV systems, formation planning, affine transformation, reinforcement learning, safe motion planning, swarm robotics.**

Paper under double-blind review.

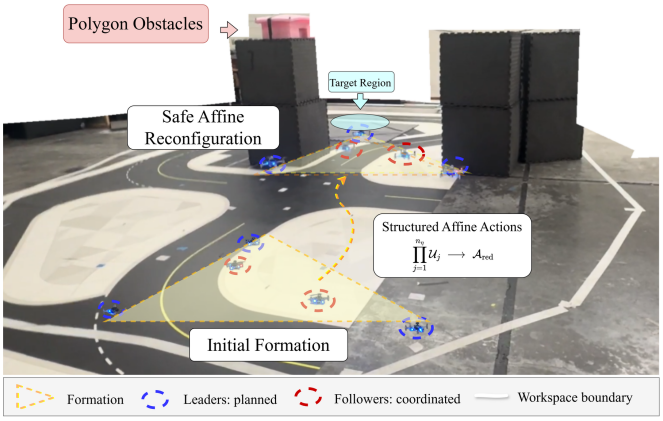


Fig. 1. **Motivating multi-UAV formation-planning scenario.** The formation adapts its geometry to safely navigate through obstacle-constrained environments while progressing toward the target region.

I. Introduction

In recent years, multi-UAV systems have shown great potential in emergency-response applications such as search and rescue [1], disaster assessment [2], and rapid situational awareness [3]. Compared with a single vehicle, UAV teams can provide wider spatial coverage, greater operational flexibility, and improved mission efficiency. Despite significant progress, many multi-UAV planning approaches still coordinate motion at the individual-vehicle level while simultaneously accounting for collision avoidance and formation maintenance [4]. As the number of UAVs increases, these vehicle-level decisions introduce increasing coordination complexity, making real-time formation planning increasingly challenging.

The advantages of formation-level planning are clear: collective motion can be represented compactly, the formation geometry can be modeled as a whole, and the high-level planning representation does not need to grow with the number of vehicles. Continuum-deformation and affine-transformation (AT) models provide a natural representation for this purpose by describing collective motion through the transformation of a reference configuration [5]. Recent work has further integrated affine transformations with deep reinforcement learning (DRL) for flexible multi-UAV formation control [6]. However, a compact representation does not directly guarantee efficient and safe planning. The remaining challenges include the combinatorial growth of discretized affine actions, geometric feasibility in obstacle-constrained environments, and efficient policy learning over the resulting decision space.

This paper proposes **ATLAS**: *Affine Transformation Learning with Action-Space Reduction*, a safety-constrained search-

to-policy framework for scalable multi-UAV formation planning. ATLAS addresses the above challenges through a structured reduce–filter–search–learn pipeline. First, a structured action-space reduction replaces the combinatorial set of simultaneous affine updates with a compact set of interpretable transformation primitives. Geometric safety conditions then filter infeasible transformations before planning and policy execution. Within the resulting safe action space, an offline beam-search teacher generates high-quality demonstrations that are distilled into a masked Deep Q-Network (DQN) and further refined through reinforcement learning. In this way, computationally expensive search is confined to offline training, while the learned policy enables low-latency formation planning without online search. Fig. 1 illustrates the motivating emergency-response scenario. Rather than independently routing each UAV around obstacles, ATLAS plans the evolution of the collective formation, allowing the team to translate, rotate, and deform as needed to safely traverse constrained regions while progressing toward the target.

The main contributions of this paper are summarized as follows:

- **Structured action-space reduction.** A compact set of interpretable affine transformation primitives is introduced to reduce the combinatorial decision space while retaining essential formation maneuvers.
- **Safety-constrained search-to-policy learning.** Safe beam-search demonstrations are distilled into a masked DQN and further refined through reinforcement learning, enabling low-cost policy execution without online search.
- **Scalable and sim-to-real multi-UAV validation.** ATLAS is evaluated under within-family distribution shifts and team sizes from 3 to 16 UAVs, and is further deployed on a physical Crazyflie swarm, demonstrating planning robustness, cardinality scalability, and sim-to-real applicability.

This article is organized as follows. Section II reviews the related work. Section III presents the affine transformation and safety formulation. Section IV details the proposed ATLAS framework. Section V presents the simulation and physical experiments. Finally, Section VI concludes this article.

II. Related Work

A. Affine and Deformable Multi-UAV Formation Planning

Formation control and motion planning for multi-UAV systems have been extensively studied [4]. Consensus and containment control remain fundamental paradigms for decentralized multi-agent coordination, with recent studies addressing formation control under dynamic communication networks [7], heterogeneous and constrained multi-agent systems [8], and robust containment under communication disruptions [9]. However, these methods primarily establish convergence and coordination properties and do not directly provide scalable geometric safety guarantees in obstacle-constrained environments. Recent safety-critical approaches have therefore incorporated control barrier functions (CBFs) and related safety filters for multi-robot collision avoidance and constrained navigation [10], [11]. While these methods provide formal safety mechanisms, agent-level collision constraints and joint decision spaces generally become more complex as the team size increases.

Affine transformation (AT) provides an alternative formation-level representation in which collective translation, rotation, scaling, and deformation are described by a small set of geometric variables rather than independently planned agent trajectories [5], [12], [13]. Recent work has further combined affine transformations with deep reinforcement learning for adaptive multi-UAV formation control [6]. However, a compact affine representation does not necessarily lead to a compact decision space. When multiple affine variables are discretized and jointly selected, the resulting action space can still grow combinatorially. This motivates a structured action-space representation that preserves essential formation maneuvers while reducing the planning and learning complexity.

B. Learning-Based Multi-UAV Planning and Coordination

Learning-based methods have been increasingly studied for multi-UAV planning and coordination due to their ability to provide efficient policy inference after training. Reinforcement learning (RL) enables UAVs to learn planning and control policies through interactions with the environment, while multi-agent reinforcement learning (MARL) extends this capability to cooperative and decentralized decision making [14]. Recent approaches have further incorporated formation and safety constraints into multi-UAV learning frameworks [15]. However, many MARL methods formulate decision making at the individual-agent level, where the joint state and action spaces can become increasingly complex as the number of agents grows. Formation-level learning provides an alternative by representing collective decisions through shared geometric variables [6]. In this work, learning is performed directly in the affine formation space, allowing the high-level policy to coordinate the collective formation without assigning an independent planning action to every UAV.

C. Safe and Search-Guided Policy Learning

Safety and efficient exploration remain important challenges in reinforcement-learning-based robotic planning. Action masking can restrict policy decisions by removing actions that violate predefined safety constraints [16], while reachability-based safety layers can identify and modify unsafe actions before execution [11]. Expert-guided learning provides another approach to reduce inefficient exploration, where demonstrations can be used to initialize or guide a learned policy [17]. More recently, model-based planning and policy learning have been combined to transfer high-quality planned behaviors into efficient neural policies [18]. These studies demonstrate the benefits of combining explicit safety mechanisms and expert guidance with policy learning. In contrast, ATLAS integrates structured action-space reduction, geometric safety filtering, offline beam-search demonstrations, and reinforcement-learning refinement within a unified formation-planning framework, while retaining only the lightweight learned policy for online deployment.

TABLE I
AFFINE TRANSFORMATION (AT) CLASSES AND PARAMETERIZATIONS CONSIDERED IN THIS WORK.

Problem	AT Class	Dim.	Decision Variables	Fixed Parameters
1	Horizontal	6	$d_1, d_2, \lambda_1, \lambda_2, \psi_r, \psi_d$	$\phi_r = \theta_r = \phi_d = \theta_d = 0,\ \lambda_3 = 1,\ d_3 = d_{3,0}$
2	2D AT in $\mathbb{R}^3$	9	$d_1, d_2, d_3, \lambda_1, \lambda_2, \phi_r, \theta_r, \psi_r, \psi_d$	$\phi_d = \theta_d = 0,\ \lambda_3 = 1$
3	3D AT in $\mathbb{R}^3$	12	$d_1, d_2, d_3, \lambda_1, \lambda_2, \lambda_3, \phi_r, \theta_r, \psi_r, \phi_d, \theta_d, \psi_d$	None

III. AFFINE TRANSFORMATION CONCEPT

Consider a team of N agents indexed by $\mathcal{V} = \{1, \dots, N\}$ evolving in $\mathbb{R}^3$. Rather than specifying the desired trajectory of each agent independently, the collective configuration is parameterized by a common affine transformation

$$\mathbf{p}_i = \mathbf{Q}\mathbf{a}_i + \mathbf{d}, \qquad i \in \mathcal{V}, \tag{1}$$

where $\mathbf{a}_i \in \mathbb{R}^3$ is the fixed *reference position* of agent i, $\mathbf{p}_i \in \mathbb{R}^3$ is its *global desired position*, $\mathbf{Q} \in \mathbb{R}^{3\times3}$ defines the deformation of the reference configuration, and $\mathbf{d} \in \mathbb{R}^3$ specifies its translation. Therefore, the desired configuration of the entire team is determined by a single affine transformation shared by all agents, replacing N independently parameterized desired trajectories with a low-dimensional representation of the collective motion. To expose the geometric structure of the collective motion, we apply the polar decomposition

$$\mathbf{Q} = \mathbf{R}(\phi_r, \theta_r, \psi_r)\mathbf{U}, \tag{2}$$

where $\mathbf{R} \in SO(3)$ represents the rigid-body rotation of the formation and $\mathbf{U} \succ 0$ is the symmetric deformation matrix. The latter admits the eigendecomposition

$$\mathbf{U} = \mathbf{R}(\phi_d, \theta_d, \psi_d)\,\mathrm{diag}(\lambda_1, \lambda_2, \lambda_3)\mathbf{R}^T(\phi_d, \theta_d, \psi_d), \tag{3}$$

where $\lambda_1, \lambda_2, \lambda_3 > 0$ are the principal stretches, and $(\phi_d, \theta_d, \psi_d)$ determine the corresponding principal deformation directions. Thus, a general 3D affine transformation is characterized by 12 geometric parameters, partitioned into decision variables and fixed parameters. We consider three AT classes of increasing dimensionality: 2D AT in a horizontal plane, 2D AT in $\mathbb{R}^3$, and 3D AT in $\mathbb{R}^3$, as summarized in Table I [12].

A. AT of Multi-Agent Systems

The agent set is partitioned as $\mathcal{V} = \mathcal{V}_B \cup \mathcal{V}_I$, with $\mathcal{V}_B \cap \mathcal{V}_I = \emptyset$, where $\mathcal{V}_B = \{1, \dots, M\}$ denotes the set of boundary (leader) agents and $\mathcal{V}_I = \mathcal{V} \setminus \mathcal{V}_B$ denotes the set of interior (follower) agents. The boundary agents define the vertices of a p-dimensional polytope embedded in $\mathbb{R}^3$., where $p = 2, 3$. Therefore, $M \geq p+1$ boundary agents are required to define and guide the desired collective configuration.

For every agent $i \in \mathcal{V}$, let $\mathbf{r}_i(t)$ denote its **actual position**, which is considered as the output of the corresponding agent-level control system. We further define the **reference position** $\mathbf{a}_i$, the **global desired position** $\mathbf{p}_i(t)$, and the **online desired position** $\mathbf{r}_{i,d}(t)$, as specified below.

Reference Positions: For every $i \in \mathcal{V}$, $\mathbf{a}_i \in \mathbb{R}^3$ is the fixed reference position of agent i. Let $\mathcal{V}_B = \{1, \dots, M\} \subset \mathcal{V}$ denote the set of boundary agents. The *reference polytope* is defined by

$$\mathcal{D}_0 = \mathrm{conv}\,\{\mathbf{a}_i \mid i \in \mathcal{V}_B\}\,. \tag{4}$$

Global Desired Positions: For every $i \in \mathcal{V}$, the global desired position $\mathbf{p}_i$ is determined by the affine transformation in (1). For compactness,

$$\mathbf{p}_i = \mathbf{f}(\boldsymbol{\eta}, \mathbf{a}_i)\,, \qquad i \in \mathcal{V}, \tag{5}$$

where

$$\boldsymbol{\eta} = [d_1\ d_2\ d_3\ \lambda_1\ \lambda_2\ \lambda_3\ \phi_r\ \theta_r\ \psi_r\ \phi_d\ \theta_d\ \psi_d]^\top \in \mathcal{H} \subseteq \mathbb{R}^{12} \tag{6}$$

collects the 12 geometric parameters of a general three-dimensional AT, where $\mathcal{H}$ denotes the admissible AT parameter set. The global desired positions of the boundary agents define the *leading polytope*

$$\mathcal{D}(\boldsymbol{\eta}) = \mathrm{conv}\,\{\mathbf{f}(\boldsymbol{\eta}, \mathbf{a}_i) \mid i \in \mathcal{V}_B\} = \mathrm{conv}\,\{\mathbf{p}_i \mid i \in \mathcal{V}_B\}\,. \tag{7}$$

Online Desired Trajectory: For each boundary agent $i \in \mathcal{V}_B$, the online and global desired positions coincide. In contrast, each interior agent $i \in \mathcal{V}_I$ updates its online desired position through local communication. The inter-agent communication is described by the directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Each interior agent $i \in \mathcal{V}_I$ has $p+1$ fixed in-neighbors $\mathcal{N}_i = \{j \in \mathcal{V} : (j, i) \in \mathcal{E}\}$ whose reference positions form a p-dimensional simplex in $\mathbb{R}^n$. The associated weights w_{ij} are uniquely determined from

$$\mathbf{a}_i = \sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{a}_j, \qquad \sum_{j\in\mathcal{N}_i} w_{ij} = 1. \tag{8}$$

Accordingly, the online desired position of every agent is formally defined as

$$\mathbf{r}_{i,d}(t) = \begin{cases} \mathbf{p}_i(t), & i \in \mathcal{V}_B, \\ \displaystyle\sum_{j\in\mathcal{N}_i} w_{ij}\mathbf{r}_j(t), & i \in \mathcal{V}_I. \end{cases} \tag{9}$$

B. Formal Specification and Scalable Safety Certification of MAS AT

We aim to enable a large-scale MAS to safely navigate an obstacle-laden position space represented by a compact polytope $\mathcal{P} \subset \mathbb{R}^3$. The restricted region is defined as

$$\mathcal{O} = \bigcup_{\ell=1}^{N_o} \mathcal{O}_\ell \subset \mathcal{P}, \tag{10}$$

where each $\mathcal{O}_\ell$ is a polytope representing an individual restricted zone. Given a reachable target domain $\mathcal{T} \subset \mathcal{P} \setminus \mathcal{O}$, we use linear temporal logic (LTL) to specify safety and liveness of the MAS AT.

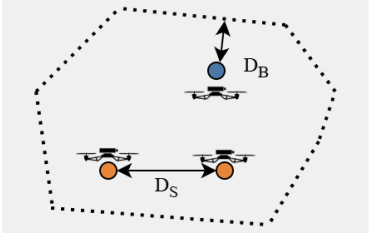


Fig. 2. **Geometric quantities used in the scalable safety certificate.** D_B denotes the minimum initial inter-agent separation, while D_S denotes the minimum initial distance between an interior agent and the boundary of the leading polytope.

Let $\square$ and $\lozenge$ denote the LTL operators "always" and "eventually," respectively. The desired MAS behavior is

$$\varphi_{\mathrm{MAS}} = \varphi_{\mathrm{safe}} \wedge \varphi_{\mathrm{live}}. \tag{11}$$

Safety and Liveness Specification: Safety requires the MAS to remain inside the position space, avoid restricted zones and inter-agent collisions, and preserve a nondegenerate AT:

$$\varphi_{\mathrm{safe}} = \square \Bigg[\mathcal{D}(\boldsymbol{\eta}) \subseteq \mathcal{P} \wedge \mathcal{D}(\boldsymbol{\eta}) \cap \mathcal{O} = \varnothing \\ \wedge \bigwedge_{\substack{i,j \in \mathcal{V} \\ i<j}} \|\mathbf{r}_i - \mathbf{r}_j\|_2 \geq 2\epsilon \wedge \bigwedge_{j=1}^{3} \lambda_j > 0 \Bigg], \tag{12}$$

where $\epsilon > 0$ denotes the agent radius and $\lambda_1, \lambda_2, \lambda_3$ are the principal deformation scales. Liveness requires the leading polytope to eventually reach the target domain:

$$\varphi_{\mathrm{live}} = \lozenge \left(\mathcal{D}(\boldsymbol{\eta}) \subseteq \mathcal{T} \right). \tag{13}$$

Scalability Challenge: Direct enforcement of (12) is computationally expensive for a large-scale MAS. In particular, pairwise collision verification requires $\mathcal{O}(N^2)$ distance evaluations, while containment and restricted-zone avoidance introduce additional constraints that grow with the number of agents and obstacles. We therefore replace these agent-level constraints with a low-dimensional safety certificate expressed directly in terms of the AT parameters $\boldsymbol{\eta}$.

Sufficient Safety Certificate: Inspired by [5], let D_B denote the minimum initial inter-agent separation and D_S the minimum initial distance between an interior agent and the boundary of the leading polytope, as illustrated in Fig. 2.

Define

$$\delta_{\max} = \min \left\{ \frac{1}{2} \left(D_B - 2\epsilon \right), D_S - \epsilon \right\}, \tag{14}$$

and

$$\lambda_{\mathrm{safe}} = \frac{\delta + \epsilon}{\delta_{\max} + \epsilon}, \qquad 0 \leq \delta \leq \delta_{\max}, \tag{15}$$

where δ is an upper bound on the agent tracking error. Let

$$\mathbf{U}_D = \left(\mathbf{Q}^\top \mathbf{Q} \right)^{1/2} \tag{16}$$

denote the pure-deformation matrix, whose eigenvalues $\lambda_1, \lambda_2, \lambda_3$ are the principal deformation scales.

The agent-level safety and liveness requirements can therefore be replaced by the following single low-dimensional AT specification:

$$\boxed{\varphi_{\mathrm{AT}} = \square \Bigg[\min_{j \in \{1,2,3\}} \lambda_j \geq \lambda_{\mathrm{safe}} \\ \wedge \ \mathcal{D}(\boldsymbol{\eta}) \subseteq \mathcal{P} \wedge \mathcal{D}(\boldsymbol{\eta}) \cap \mathcal{O} = \varnothing \Bigg] \\ \wedge \lozenge \left(\mathcal{D}(\boldsymbol{\eta}) \subseteq \mathcal{T} \right).} \tag{17}$$

Equation (17) reduces large-scale MAS coordination to a geometric planning problem over the 12 AT parameters $\boldsymbol{\eta}$, whose dimension is independent of the number of agents. This representation provides a scalable foundation for real-time planning, learning, and control of large-scale MASs.

IV. Proposed Method

Fig. 3 summarizes the proposed ATLAS framework, which is organized into three stages. In **(a) Safe Affine RL Dynamics**, the multi-UAV planning problem is represented at the formation level using affine transformation variables, while LTL-based geometric safety constraints define the admissible state transitions. In **(b) Safety-Constrained Search-to-Policy Learning**, an offline beam-search teacher generates safe planning demonstrations that supervise a masked DQN policy, which is subsequently refined through reinforcement learning. This transfers the guidance of computationally expensive search into a lightweight policy so that online deployment does not require beam search. In **(c) Formation-Level Sim-to-Real Deployment**, the learned affine policy generates formation-level references that are executed through the leader–follower coordination architecture and vehicle-level feedback controllers, enabling the same high-level planning architecture to be evaluated in simulation and on the physical multi-UAV platform.

A. Problem Formulation

We formulate safe AT planning as a finite-horizon dynamic programming (DP) problem

$$\mathfrak{M} = \left(\mathcal{S}, \mathcal{A}, \mathcal{F}, \mathcal{C}, s_0, \mathcal{S}_T \right), \tag{18}$$

where $\mathcal{S}$ and $\mathcal{A}$ denote the state and action spaces, $\mathcal{F}$ is the transition map, $\mathcal{C}$ is the stage cost, s_0 is the initial state, and $\mathcal{S}_T$ is the terminal set.

State Space: Let $\boldsymbol{\eta} \in \mathbb{R}^{n_\eta}$ collect the active AT parameters, where $n_\eta \in \{6, 9, 12\}$ depends on the AT class. Each DP state is defined by

$$s_k = \boldsymbol{\eta}_k \in \mathcal{S}, \tag{19}$$

which uniquely determines the leading polytope $\mathcal{D}(\boldsymbol{\eta}_k)$. Consistent with (17), the safe state space is

$$\mathcal{S} = \left\{ \boldsymbol{\eta} : \mathcal{D}(\boldsymbol{\eta}) \subseteq \mathcal{P}, \ \mathcal{D}(\boldsymbol{\eta}) \cap \mathcal{O} = \varnothing, \ \min_j \lambda_j \geq \lambda_{\mathrm{safe}} \right\}. \tag{20}$$

Thus, every state represents a collision-free and geometrically admissible configuration of the complete agent team.

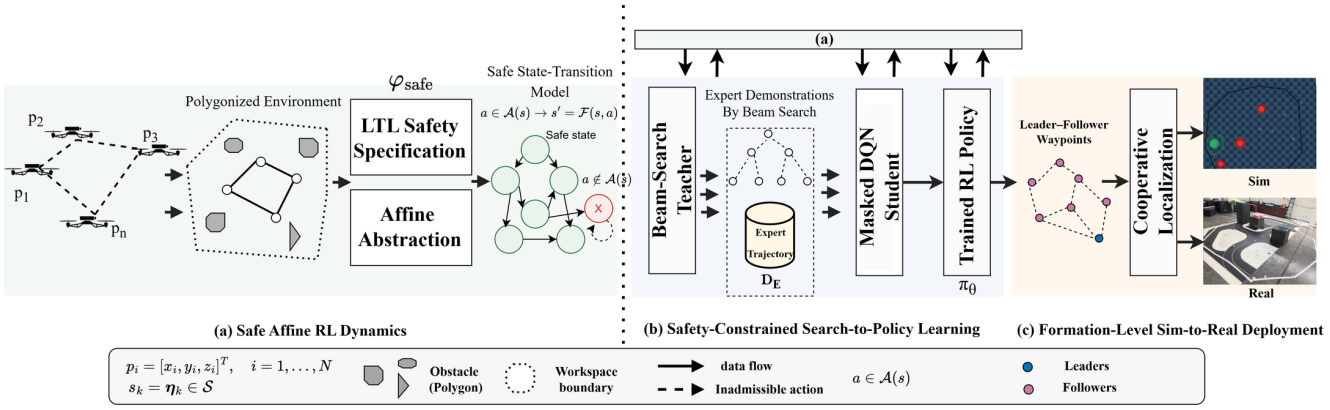


Fig. 3. **Overview of the proposed ATLAS framework.** (a) **Safe affine RL dynamics:** the multi-UAV formation and polygonized environment are mapped through affine abstraction and an LTL-based safety specification to a safe state-transition model. Admissible affine actions $a \in \mathcal{A}(s)$ generate transitions between safe formation states according to $s' = \mathcal{F}(s, a)$. (b) **Safety-constrained search-to-policy learning:** an offline beam-search teacher generates safe expert trajectories D_E, which provide supervision for a masked DQN policy before reinforcement-learning refinement to obtain π_θ. (c) **Formation-level sim-to-real deployment:** the learned policy generates affine formation commands that are executed through a leader–follower coordination architecture with vehicle-level feedback tracking in simulation and real-world multi-UAV experiments.

Action Space: An action specifies a discrete increment of the active AT parameters:

$$a_k = \Delta\boldsymbol{\eta}_k \in \mathcal{A}. \tag{21}$$

For each $s = \boldsymbol{\eta} \in \mathcal{S}$, define the admissible action set

$$\mathcal{A}(s) = \{a \in \mathcal{A} : \boldsymbol{\eta} + a \in \mathcal{S}\}. \tag{22}$$

Hence, actions producing unsafe configurations are excluded from the DP search.

State Transition: The AT dynamics are deterministic and defined by

$$s_{k+1} = \mathcal{F}(s_k, a_k) = \boldsymbol{\eta}_k + \Delta\boldsymbol{\eta}_k, \tag{23}$$

where $a_k \in \mathcal{A}(s_k)$. Therefore, every admissible action induces a transition between two safe AT configurations.

Stage Cost: For every admissible transition, let

$$c_k = \mathcal{C}(s_k, a_k, s_{k+1}) \geq 0 \tag{24}$$

denote its stage cost. The cost can encode mission objectives such as travel distance, deformation effort, and reconfiguration complexity.

Terminal Set: The terminal set consists of safe AT configurations whose leading polytope is contained in the target domain:

$$\mathcal{S}_T = \{\boldsymbol{\eta} \in \mathcal{S} : \mathcal{D}(\boldsymbol{\eta}) \subseteq \mathcal{T}\}. \tag{25}$$

Optimal AT Planning: Given $s_0 \in \mathcal{S}$, the objective is to determine a policy $\pi : \mathcal{S} \to \mathcal{A}$ that minimizes

$$J^\pi(s_0) = \sum_{k=0}^{K-1} \mathcal{C}(s_k, a_k, s_{k+1}), \tag{26}$$

subject to

$$s_{k+1} = \mathcal{F}(s_k, a_k), \qquad a_k \in \mathcal{A}(s_k), \qquad s_K \in \mathcal{S}_T. \tag{27}$$

The resulting optimal plan is

$$\tau^\star = \left(s_0, a_0^\star, s_1^\star, \ldots, a_{K-1}^\star, s_K^\star\right), \tag{28}$$

where $s_K^\star \in \mathcal{S}_T$.

Remark: The DP objective is to determine a minimum-cost sequence of admissible AT actions that safely drives the collective configuration to the terminal set $\mathcal{S}_T$. Although the optimal policy can, in principle, be obtained from the Bellman optimality equation, exhaustive evaluation of the constrained state-action space becomes computationally prohibitive as the planning horizon and discretization resolution increase. We therefore employ RL to learn the AT planning policy through interactions with the transition model, eliminating explicit Bellman optimization over the entire state space while preserving the long-horizon minimum-cost objective.

B. Safety-Constrained Search-to-Policy Learning

The teacher performs a bounded-width beam search over the safe state-transition model, following the general strategy in [19]. Unlike exhaustive tree search, which retains every reachable partial plan, beam search keeps only the B most promising candidates at each search depth. In ATLAS, the beam width is $B = 8$ and the look-ahead depth is $L = 2$. Each search node stores a partial safe trajectory

$$\xi_d = (s_t, a_t, s_{t+1}, \ldots, a_{t+d-1}, s_{t+d}), \qquad d \leq L. \tag{29}$$

Only actions in $\mathcal{A}(s)$ are expanded; consequently, every node retained by the teacher corresponds to a transition satisfying the same geometric constraints later used by the masked student.

Starting with the singleton beam $\mathcal{B}_0 = \{(s_t)\}$, the teacher forms the candidate set at depth $d+1$ as

$$\begin{aligned}\mathcal{X}_{d+1} = \{\, \xi_d \,\|\, (a, s') : \xi_d \in \mathcal{B}_d, \\ a \in \mathcal{A}(s),\ s' = \mathcal{F}(s, a) \,\},\end{aligned} \tag{30}$$

where s is the terminal state of ξ_d and $\|$ denotes trajectory extension. Candidate partial trajectories are ranked by

$$S(\xi_d) = \sum_{j=0}^{d-1} \gamma^j r(s_{t+j}, a_{t+j}, s_{t+j+1}), \tag{31}$$

with $\gamma = 0.98$. The shaped planning reward is defined as

$$r(s,a,s') = w_p\left[d(s) - d(s')\right] - w_s + w_A\left[A_{\text{out}}(s) - A_{\text{out}}(s')\right] + R_T\mathbb{I}_{\mathcal{S}_T}(s'), \tag{32}$$

where $d(s)$ denotes the distance between the formation centroid and the target centroid, $A_{\text{out}}(s)$ denotes the portion of the formation polygon outside the target region, and $\mathbb{I}_{\mathcal{S}_T}(s')$ indicates terminal success. In the reported experiments, we use $w_p = 100$, $w_s = 1$, $w_A = 500$, and $R_T = 10000$. Collision avoidance and workspace containment are enforced through the admissible action set $\mathcal{A}(s)$ rather than through collision penalties in the reward. This shaped reward provides the learning implementation of the stage objective in the DP formulation. The implemented learning objective uses the negative shaped reward as a surrogate stage cost, so that maximizing accumulated reward corresponds to minimizing the associated shaped learning objective. For TD updates, rewards are multiplied by 10^{-3} for numerical stability; this positive scaling preserves the reward ordering.

After two expansion levels, the teacher selects the highest-scoring leaf and executes only its first action. The search is then restarted from the resulting state, yielding a receding-horizon expert that repeatedly accounts for the near-term consequence of each decision. This procedure terminates when the formation reaches $\mathcal{S}_T$, when no admissible successor exists, or when the episode horizon is reached. Beam search therefore serves only as an offline demonstration generator; it is not invoked by ATLAS during deployment.

A successful expert trajectory can be written as

$$\tau = (s_0, a_0, s_1, a_1, \ldots, s_T)\,. \tag{33}$$

The resulting trajectories form the demonstration dataset D_E. These demonstrations provide structured supervision for the student policy and reduce the reliance on purely exploratory reinforcement learning in the large formation-planning state space.

The student is first pretrained on the beam-search state–action labels using cross-entropy imitation loss, following the policy-distillation principle [20]. During the subsequent fine-tuning stage, the network is updated using a combined temporal-difference and auxiliary imitation objective:

$$\mathcal{L} = \lambda_{\text{TD}}\mathcal{L}_{\text{TD}} + \lambda_{\text{IL}}\mathcal{L}_{\text{IL}}. \tag{34}$$

For the reported experiments, $\lambda_{\text{IL}} = 1.0$ and $\lambda_{\text{TD}} = 0.2$, and TD updates begin after episode 100. Retaining the imitation term during fine-tuning preserves the teacher signal while allowing value updates from environment rollouts. Consequently, beam search is required only during training, while the learned policy is retained for low-latency online deployment.

C. Masked DQN Policy Learning

The student policy is represented using a Deep Q-Network. For each state s_i, the network predicts action values

$$Q_\theta(s_i, a), \qquad a \in \mathcal{A}. \tag{35}$$

Before selecting an action, unsafe actions are removed using the admissible action set $\mathcal{A}(s_i)$ defined above. The DQN therefore evaluates only actions that satisfy the geometric safety constraints.

During deployment, the action is selected greedily as

$$a_t = \arg\max_{a\in\mathcal{A}(s_t)} Q_\theta(s_t, a). \tag{36}$$

Exploration is applied only during reinforcement-learning training.

This masking mechanism prevents the DQN from selecting affine transitions that violate the safety constraints, while reinforcement learning optimizes performance over the remaining admissible actions. The network architecture and training hyperparameters used for the reported experiments are summarized in Table II.

TABLE II
NETWORK ARCHITECTURE AND TRAINING HYPERPARAMETERS.

Parameter	Value	Parameter	Value
Hidden layers	256–256–128	Optimizer	Adam
Learning rate	10^{-4}	Discount factor	0.99
Replay buffer	10^5	Batch size	128
Min. replay	10^3	Target update	500 steps
TD update	4 steps	IL pretraining	30 epochs
Beam demos.	200 episodes	IL / TD weights	1.0 / 0.2
TD reward scale	10^{-3}	TD start	Episode 100
Exploration ϵ	$0.05 \rightarrow 0.02$	Seeds	71–73

D. Formation-Level Sim-to-Real Deployment

The learned ATLAS operates at the formation-planning level and does not directly generate low-level control commands. At each planning step k, the policy selects an admissible affine action a_k and updates the formation state according to

$$\boldsymbol{\eta}_{k+1} = \boldsymbol{\eta}_k + a_k. \tag{37}$$

The updated affine state is then mapped to the desired position of each boundary agent through

$$\mathbf{p}_{i,k+1} = \mathbf{f}\left(\boldsymbol{\eta}_{k+1}, \mathbf{a}_i\right), \qquad i \in \mathcal{V}_B. \tag{38}$$

These formation-level references define the desired trajectories executed by the boundary UAVs.

For physical deployment, we adopt the decentralized affine-transformation control architecture in [21]. The interior agents do not require separate high-level planning policies or access to the planned leader trajectories. Instead, their online desired positions are reconstructed from the measured positions of their in-neighbors using fixed barycentric weights, according to Eq. (9). Therefore, the high-level planner determines the collective formation evolution through the boundary agents, while the interior agents acquire the resulting affine deformation through local coordination. Importantly, the state and action dimensions of ATLAS depend on the affine representation rather than the number of UAVs, allowing the same formation-level policy structure to be used as the team size changes.

For deployment, the planned affine trajectory is executed through the same leader–follower architecture in both simulation and the physical Crazyflie swarm. Following [21], the boundary agents use feedback tracking with trajectory

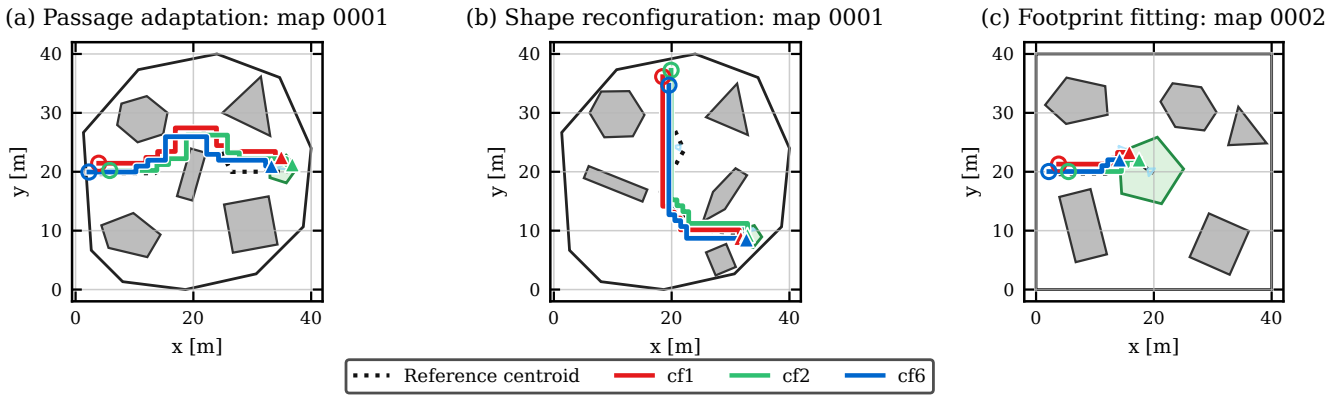


Fig. 4. **Representative held-out maps and ATLAS leader trajectories.** Each panel shows the lowest-index shifted-test map in its scenario family on which all three policy seeds succeed; trajectories are from the deterministic seed-71 execution. Orange, magenta, and blue denote the three boundary leaders. Circles and triangles mark initial and terminal positions, the black dotted line denotes the reference centroid, and light-blue polygons show selected reference formations. Gray polygons are obstacles and green polygons indicate the target. These panels illustrate representative policy behavior; aggregate results include all randomized maps and three seeds.

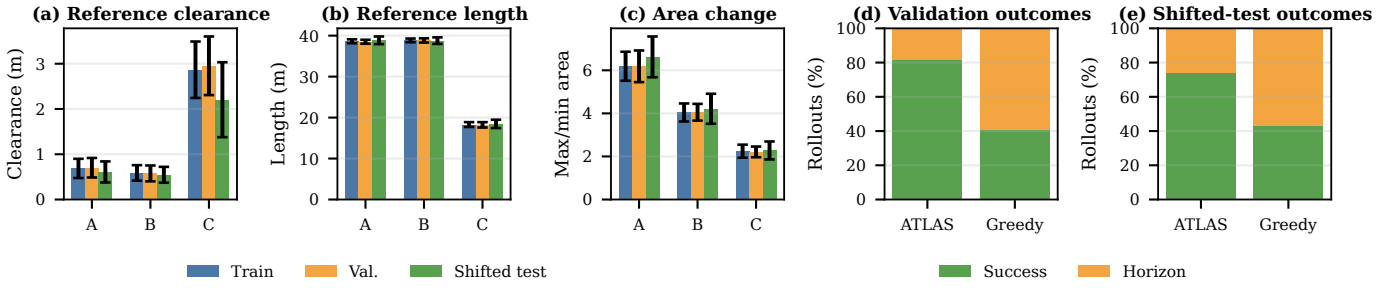


Fig. 5. **Map difficulty and outcome decomposition.** Panels (a)–(c) report the reference-trajectory minimum obstacle clearance, centroid path length, and maximum-to-minimum formation-area ratio for families A/B/C; bars are map means and error bars are sample standard deviations. Panels (d)–(e) decompose validation and within-family shifted-test rollouts into success, no-safe-action, horizon-limit, and other failures. ATLAS aggregates three seeds (450 validation and 900 shifted-test rollouts), whereas masked greedy is deterministic (150 and 300 rollouts).

feedforward, whereas the interior agents track the neighbor-derived position references. In the physical implementation, vehicle states are measured by a Vicon motion-capture system and used as feedback during trajectory execution. By separating formation-level planning from vehicle-level trajectory tracking, the learned high-level policy can be deployed on the physical platform without retraining it for the low-level vehicle dynamics. We evaluate this deployment pipeline using matched formation-reconfiguration tasks in MuJoCo and physical Crazyflie experiments, comparing trajectory tracking accuracy, inter-UAV separation, and obstacle clearance.

V. Experimental Results

We evaluate ATLAS through three simulation studies and a physical multi-UAV experiment, addressing four research questions: **Q1:** Does search-guided learning improve planning reliability while retaining low online decision cost? **Q2:** Does structured action-space reduction reduce computational complexity without sacrificing planning reliability? **Q3:** Does the learned formation-level policy remain robust under within-family distribution shifts and changes in swarm size? **Q4:** Can the resulting formation trajectories be executed on physical UAVs, and what sim-to-real limitations emerge?

A. Planner Comparison

This experiment addresses **Q1: whether search-guided learning improves planning reliability while retaining low online decision cost.** ATLAS uses beam-search demonstrations only during training; at test time, the learned DQN selects actions independently without invoking search. Beam search is therefore reported as an offline teacher and upper-reference planner rather than a deployable component of ATLAS.

All planners are evaluated on the same randomized polygon-map benchmark and fixed train/validation/test partitions. A rollout is successful only if the entire formation reaches the target within 160 steps without violating the planner's geometric feasibility conditions. The affine DQN variants share the same state representation, feasibility checker, horizon, and 1,800-episode budget, making them the controlled comparisons used to isolate the effects of action-space reduction and beam-guided initialization. MAPPO and Def-MARL require method-specific adapters and are therefore reported as adapted external references rather than strictly controlled ablations. We additionally evaluate a deterministic masked-greedy baseline implemented for this study. At each state it applies the same 13-action feasibility mask as ATLAS, evaluates every admissible action with the one-step environment reward, and selects the highest-reward action (ties are resolved by target-centroid

TABLE III

PLANNER COMPARISON ON THE RANDOMIZED A/B/C BENCHMARK. LEARNED METHODS REPORT MEAN ± SAMPLE STANDARD DEVIATION OVER SEEDS 71–73 ON 150 VALIDATION AND 300 WITHIN-FAMILY DISTRIBUTION-SHIFTED TEST MAPS. DETERMINISTIC BASELINES ARE SINGLE EVALUATIONS. BEAM SEARCH IS THE OFFLINE TEACHER; ATLAS DOES NOT INVOKE BEAM SEARCH AT TEST TIME. MAPPO AND DEF-MARL ARE ADAPTED EXTERNAL REFERENCES RATHER THAN CONTROLLED ABLATIONS.

Planner variant	Actions	Signal	Validation success (%)	Test success (%)	Decision (ms/step)	Online deployment
Controlled affine-planner comparisons						
Full-action DQN [22]	729	RL	52.89 ± 11.10	49.56 ± 9.83	0.275 ± 0.003	Yes
Reduced-action DQN [22]	13	RL	53.78 ± 17.91	50.78 ± 15.22	0.287 ± 0.046	Yes
Masked greedy (baseline)	13	One-step reward	40.67	43.33	2.727	Yes
IL-only student [20]	13	IL	62.22 ± 3.67	58.67 ± 7.02	0.335 ± 0.003	Yes
Beam-search teacher [19]	13	None	**100.00**	**100.00**	51.982	No
ATLAS (proposed)	13	IL + RL	81.78 ± 2.52	74.00 ± 5.51	0.242 ± 0.009	Yes
Adapted external references (not controlled ablations)						
MAPPO (NeurIPS 2022) [14]	3^6 joint	RL	13.78 ± 17.40	12.89 ± 14.06	**0.186 ± 0.004**	Yes
Def-MARL (RSS 2025) [15]	Continuous	Safe MARL	0.00 ± 0.00	0.00 ± 0.00	**0.186 ± 0.017**	Yes

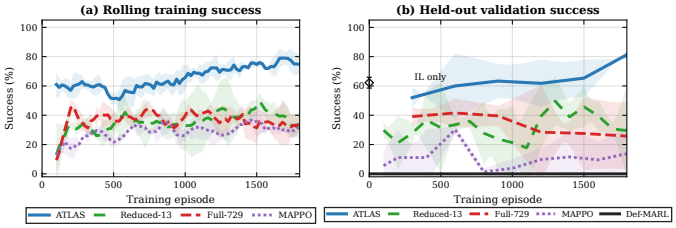


Fig. 6. **Learning curves for trainable planner variants.** The curve labeled ATLAS is the proposed beam-guided IL+RL method. Training and validation are shown side by side. Lines denote cross-seed averages, and shaded regions denote one standard deviation over seeds 71–73. IL-only is the post-pretraining reference point. Def-MARL is checkpoint-evaluated on the same held-out maps and is shown in the validation panel only. Its curve reports a transition-matched A/B/C adaptation and is included as an external reference, not as a controlled ablation or a reproduction of performance on the method's original benchmark.

progress and action index). It uses neither learning nor look-ahead search and therefore isolates the value of non-myopic policy learning under the same safety filter. The high-level planner is trained in the deterministic affine abstraction, while MuJoCo is used for selected trajectory replay and sim-to-real evaluation. This separation is consistent with the formation-level planning architecture described in Section IV-D.

Among the controlled DQN variants, ATLAS improves validation success by 28.00 percentage points over the seed-matched reduced-action RL baseline and by 28.89 points over the full-action DQN. The corresponding test-set gains are 23.22 and 24.44 points, respectively. IL only also exceeds both RL-only variants, but RL fine-tuning adds a further 19.56 validation points and 15.33 test points over IL only. This decomposition shows that the demonstrations provide a substantial initial reliability gain, while subsequent RL further improves performance beyond direct policy distillation. ATLAS retains sub-millisecond learned decision latency. Beam search solves every evaluated map, but its 51.982-ms decision time is approximately 215 times that of ATLAS and is not incurred during deployment. The one-step masked-greedy baseline reaches only 40.67% validation and 43.33% shifted-test success despite using the identical 13-action safety mask. ATLAS improves these rates by 41.11 and 30.67 percentage points, respectively, showing that immediate reward maximization cannot replace the learned multi-step strategy. Its 2.727-ms decision cost also reflects evaluating all feasible one-step successors, whereas ATLAS performs one network action selection.

The adapted MAPPO and Def-MARL results are reported as external references because their observation and action interfaces differ from those of ATLAS. Under the documented adapters and training budgets, neither method achieves competitive performance on the strict joint reachability-and-safety criterion. We therefore do not use these comparisons to attribute the gains of individual ATLAS components. Nevertheless, the observed performance gap is consistent with the hypothesis that explicitly representing formation structure in the decision space can simplify the coordination problem compared with learning inter-agent coordination directly from decentralized actions. Because the interfaces and training procedures differ, this observation should be interpreted as suggestive rather than as a controlled causal comparison.

B. Action-Space Reduction Ablation

This experiment addresses **Q2: whether structured action-space reduction can lower computational complexity without sacrificing planning performance.** Starting from the full ternary Cartesian product, we construct nested action sets that retain actions affecting at most $k \in \{1, \ldots, 6\}$ affine parameters.

The nested study reveals how computational cost scales with increasing action-space complexity rather than relying only on a 13-versus-729 endpoint comparison. Increasing k from 1 to 6 expands the nominal action set by 56.1 times and the mean feasible branching factor by 54.0 times, whereas the feasible fraction changes by only 3.31 percentage points. Thus, geometric masking removes unsafe actions but does not collapse the combinatorial branching of simultaneous affine

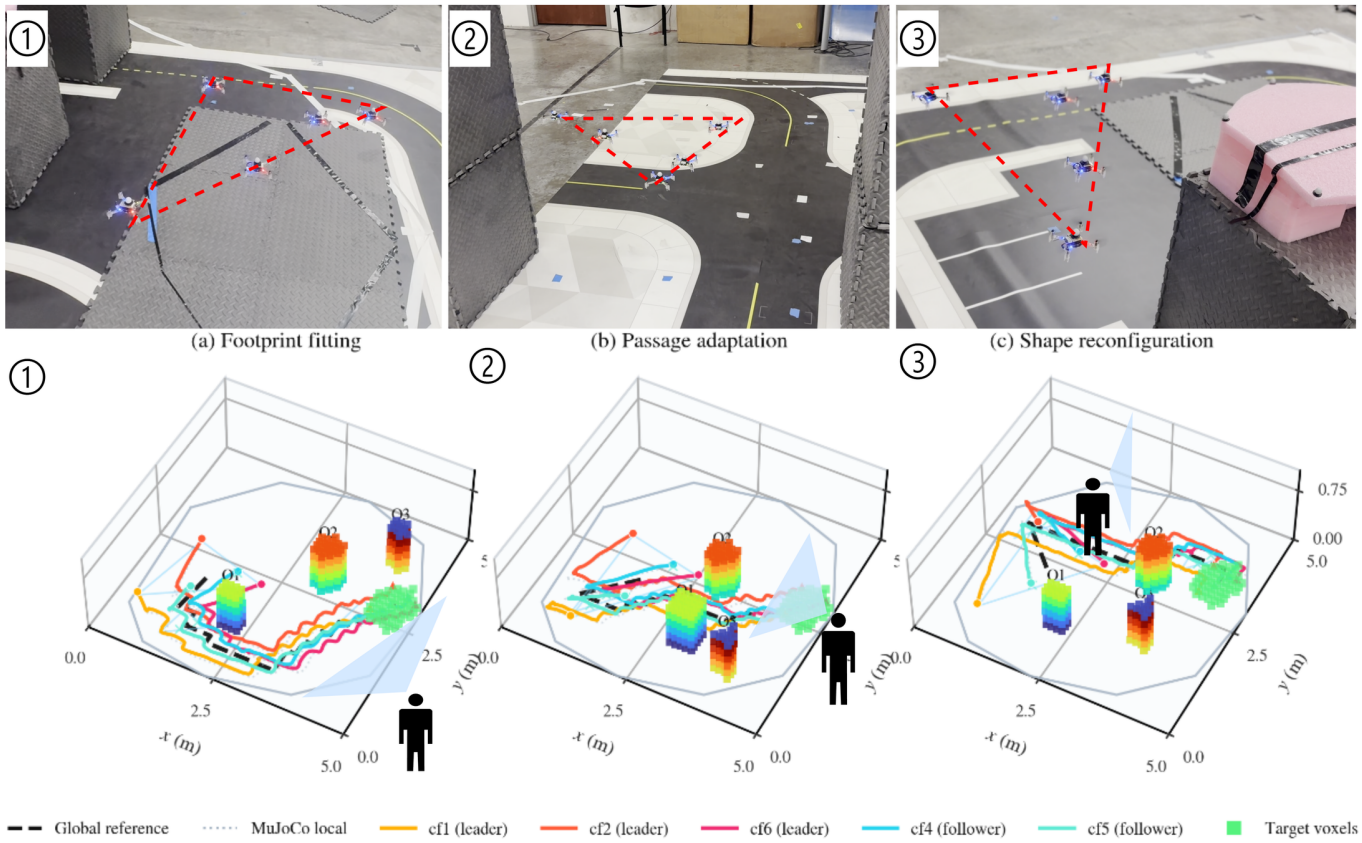


Fig. 7. **Three-dimensional visualization of the matched-layout simulation and physical-flight trajectories.** The columns show (a) footprint fitting, (b) passage adaptation, and (c) shape reconfiguration. The upper row presents representative physical-flight images; the circled numerals label the photograph associated with each corresponding column. Colored voxel point clouds depict polygonal obstacle volumes, whereas green voxels identify the target region. For clarity, the fixed-altitude trajectories are projected onto the workspace plane with small offsets between visual layers; the displayed voxel height is therefore illustrative. The black dashed curve denotes the global formation-centroid reference obtained from the deployment trajectory, and the light-blue polygons show the reference leader formation at selected instants. Gray dotted curves denote MuJoCo executions. The five Vicon trajectories are assigned distinct colors to preserve vehicle identity: three warm hues distinguish the boundary leaders cf1, cf2, and cf6, while two cool hues distinguish the interior followers cf4 and cf5. Thus, the legend colors indicate role categories rather than a single color for every vehicle in a category. Circles and triangles indicate the initial and pre-landing positions, respectively.

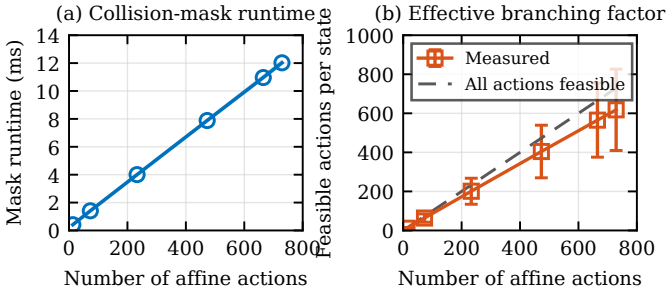


Fig. 8. **Action-space scaling on 5,250 states from all 1,050 maps.** (a) Median collision-mask runtime with the shaded median-to-95th-percentile band. (b) Mean feasible branching factor with one-standard-deviation bars.

TABLE IV
NESTED AFFINE ACTION-SPACE ABLATION. EACH SET CONTAINS JOINT UPDATES AFFECTING AT MOST k OF THE SIX AFFINE DIMENSIONS.

k	Actions	Feasible	Feasible (%)	Mask ms
1	**13**	11.45	**88.07**	**0.414**
2	73	63.60	87.13	1.410
3	233	200.96	86.25	4.005
4	473	404.40	85.50	7.883
5	665	565.04	84.97	10.966
6	729	617.88	84.76	12.022

updates. The 729-action mask is 29.0 times slower than the 13-action mask. The trained endpoints supply the complementary outcome measure: under the same 1,800-episode budget, the 13-action and 729-action RL-only DQNs obtain similar validation success (53.78% versus 52.89%) and test success (50.78% versus 49.56%) relative to their substantial cross-seed variation. The larger action space therefore adds masking cost without a measured reliability benefit under this budget. Together, these results support the 13-action set as a favorable empirical trade-off for the evaluated benchmark; they do not claim that coordinated multi-parameter actions are unnecessary for every affine planning problem.

C. Within-Family Distribution Shift and Swarm-Size Scalability

This experiment addresses **Q3: whether ATLAS remains robust under within-family distribution shifts and changes**

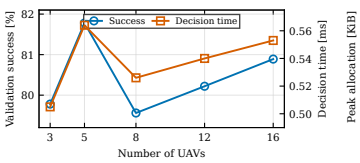


Fig. 9. **Planning reliability and online decision cost.** Validation success remains near 80% from 3 to 16 vehicles, while mean policy-decision latency changes only mildly.

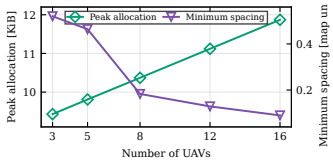


Fig. 10. **Planner allocation and formation separation.** Measured Python working allocation grows mildly with team size, whereas minimum pairwise spacing decreases as more vehicles occupy the same reference footprint.

in swarm size. We evaluate generalization across three scenario families: passage adaptation (A), shape reconfiguration (B), and footprint fitting (C). Each family contains 200 training, 50 validation, and 100 independently generated within-family distribution-shifted test maps. Starting from one family template, training/validation maps perturb each obstacle center by $\mathcal{U}[-0.55, 0.55]$ m, its two size axes by $\mathcal{U}[0.90, 1.10]$, and its orientation by $\mathcal{U}[-12°, 12°]$. Within each scenario family, shifted-test maps broaden center and orientation perturbations and shift obstacle sizes toward larger values, using $\mathcal{U}[-0.85, 0.85]$ m, $\mathcal{U}[0.94, 1.18]$, and $\mathcal{U}[-20°, 20°]$, respectively. Path waypoints are perturbed by up to 0.45 m during training/validation and 0.80 m at test time; target center/radius ranges likewise widen from 0.30 m and $[0.94, 1.08]$ to 0.55 m and $[0.86, 1.14]$. Rejection sampling retains only maps with at least 0.35 m trajectory–obstacle clearance, contained geometry, and mutually separated obstacle and target polygons. Separate deterministic seeds are used for every family and split. Because train, validation, and shifted-test maps retain the same three family templates and obstacle-shape vocabulary, this protocol measures within-family distribution shift, not transfer to novel task families or arbitrary environments. Table V reports validation performance over three seeds; no validation map is used for gradient updates or beam-demonstration generation.

The shifted-test maps preserve reference path length but increase variability and reduce mean reference clearance in every family (for example, family C drops from 2.95 ± 0.65 m on validation maps to 2.20 ± 0.83 m). Their area change factors also increase modestly, from 6.19, 4.05, and 2.21 to 6.62, 4.21, and 2.28 for A/B/C. These statistics verify that the test split changes geometric difficulty without changing the family identity. All observed ATLAS failures are horizon-limit failures: 18.22% on validation and 26.00% under the within-family shift; no rollout exhausts the feasibility mask. Masked greedy similarly has no mask-exhaustion failure but times out on 59.33% and 56.67% of the two splits, consistent with locally safe yet non-myopic progress being the principal challenge.

The generalist exhibits family-dependent performance, reaching 95.33% on passage adaptation, 76.00% on shape reconfiguration, and 74.00% on footprint fitting. As a secondary diagnostic, we further evaluate whether balanced specialist distillation can redistribute performance across these families. Successful rollouts from the generalist on A and the shape-focused specialist on B/C are distilled into a single 13-action network using balanced family sampling. The resulting policy improves B from 76.00% to 94.00% and C from 74.00% to 100.00%, while reducing A to 72.00%. Its aggregate validation success increases from 81.78% to 88.67%. Because this diagnostic uses a single independently evaluated seed, it is not substituted into the three-seed scalability analysis and should not be interpreted as evidence of a variance-robust improvement.

TABLE V
VALIDATION PERFORMANCE ACROSS THE THREE SCENARIO FAMILIES. GENERALIST VALUES AGGREGATE 50 VALIDATION MAPS PER FAMILY AND THREE SEEDS (150 ROLLOUTS); THE DISTILLED FUSION IS ONE INDEPENDENTLY EVALUATED SEED.

Scenario family	Rollouts	Generalist success (%)	Distilled success (%)	Decision (ms/step)
A: Passage adaptation	150	**95.33**	72.00	0.566
B: Shape reconfiguration	150	76.00	**94.00**	0.568
C: Footprint fitting	150	74.00	**100.00**	**0.559**
Overall	450	81.78	**88.67**	0.564

TABLE VI
SWARM-SIZE SCALABILITY SUMMARY. COLLISION-CHECK VALIDITY IS THE FRACTION OF VISITED STATES WITH AT LEAST ONE ADMISSIBLE SAFE ACTION.

UAVs	Success (%)	Decision (ms)	Peak alloc. (KiB)	Min spacing
3	79.78	**0.505**	**9.43**	**0.520**
5	**81.78**	0.564	9.81	0.464
8	79.56	0.526	10.37	0.183
12	80.22	0.540	11.12	0.129
16	80.89	0.553	11.87	0.090

We further evaluate 3, 5, 8, 12, and 16 vehicles. For the original five-vehicle case, the reference formation is unchanged. For other sizes, vehicles are sampled uniformly along the same reference-footprint boundary; thus, the test varies team cardinality without enlarging the corridor. Every entry aggregates 450 rollouts (150 maps and three policy seeds).

All evaluated rollouts retained a nonempty safety-filtered action set. Success varies by only 2.22 percentage points across 3–16 vehicles, with no monotonic degradation as cardinality increases. The policy's state and output dimensions are independent of vehicle count, so the observed latency change comes primarily from polygon construction and collision checking. From 3 to 16 vehicles, mean decision latency increases by less than 0.05 ms and measured Python allocation by 2.44 KiB. The principal scaling limit is geometric rather than computational: placing more vehicles on the same reference footprint reduces worst-case spacing from 0.520 to 0.090 map units. The experiment therefore supports computational and policy-level cardinality scalability within the tested range, while exposing a density-dependent separation limit; it does not establish hardware scalability.

D. Physical Experiments: Sim-to-Real Evaluation

This experiment addresses **Q4: whether ATLAS-generated formation trajectories can be executed on physical UAVs and what sim-to-real limitations emerge.** We evaluate three

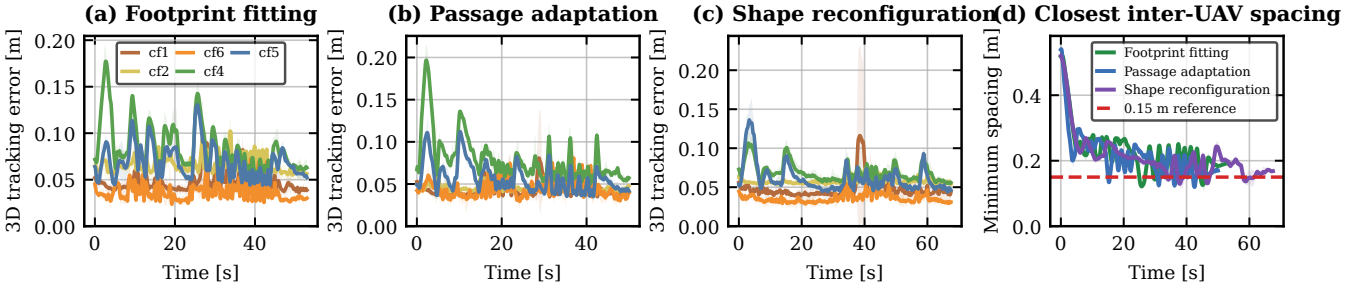


Fig. 11. **Physical-flight tracking and separation diagnostics.** Panels (a)–(c) report the per-vehicle 3D tracking errors for footprint fitting, passage adaptation, and shape reconfiguration, respectively. Curves are means and shaded bands are one sample standard deviation over three completed runs. Panel (d) shows the instantaneous closest center-to-center inter-UAV spacing averaged over the same runs; the dashed line marks the 0.15 m separation reference.

TABLE VII
MATCHED-LAYOUT SIMULATION AND PHYSICAL-FLIGHT RESULTS.

Task	Domain	Runs	RMSE (m)	Max. error (m)	Min. spacing (m)	Min. clearance (m)	Obs. entries
C	MuJoCo	3	0.078 ± 0.000	0.230	0.170	0.099	0
	Crazyflie	3	$\mathbf{0.065 \pm 0.004}$	0.195	0.107	0.117	0
A	MuJoCo	3	0.095 ± 0.000	0.429	0.156	0.046	0
	Crazyflie	3	$\mathbf{0.059 \pm 0.002}$	0.218	0.108	0.015	0
B	MuJoCo	3	0.060 ± 0.000	0.209	0.160	0.076	0
	Crazyflie	3	$\mathbf{0.058 \pm 0.003}$	0.250	0.112	0.043	0

Note: Metrics are computed over three successful executions per task. RMSE is mean $\pm$ sample standard deviation; the remaining metrics report the worst observed values. MuJoCo repeats are deterministic.

representative layouts corresponding to task families C (footprint fitting), A (passage adaptation), and B (shape reconfiguration). For each layout, three deterministic MuJoCo replays are compared with three successful five-Crazyflie flights. The physical layouts retain the same polygonal workspace, polygonal target representation, and obstacle-shape vocabulary as their simulation counterparts, while obstacle coordinates and clearances are adjusted to the laboratory setup. Thus, the physical layouts instantiate the same three task families but are not coordinate-identical replicas of the randomized simulation maps. Both simulation and hardware experiments use three boundary leaders and two interior followers at an altitude of approximately 0.75 m. Desired and measured 3D trajectories are recorded from MuJoCo and Vicon, respectively. Across the three physical layouts, we conducted 13 flight attempts: five for family C, three for family A, and five for family B. Nine flights successfully completed the planned task, with three successful trials per family. Four trials were unsuccessful because of insufficient inter-UAV separation during execution, with two collision events observed in family C and two in family B. Table VII reports trajectory-level metrics for the three successful trials in each family and their matched MuJoCo replays. The MuJoCo replays are deterministic and therefore exhibit zero run-to-run standard deviation.

All nine successful hardware flights place every drone inside the target before the separate landing maneuver and contain no measured-center entry into a planned obstacle polygon. Navigation success is evaluated at the end of the planning trajectory; landing completion is reported separately, so a landing-spread position outside the target is excluded from the navigation success criterion. The table therefore characterizes trajectory quality for successful executions rather than implying a 100% physical success rate over all flight attempts.

The desired leader-triangle area varies by factors of 7.0, 9.9, and 10.3 in families C, A, and B, respectively, confirming substantial formation-footprint adaptation in the successful hardware flights. MuJoCo maintains minimum spacing above 0.15 m in all three layouts (0.156–0.170 m), whereas the corresponding hardware minima decrease to 0.107–0.112 m. The narrow-passage clearance also decreases from 0.046 m in MuJoCo to 0.015 m in hardware. These discrepancies expose a sim-to-real safety gap between formation-level planning and physical execution.

The four failed trials further illustrate this gap. In these runs, insufficient inter-UAV separation during execution resulted in vehicle collisions. This observation is consistent with the reduced separation margins measured in the successful hardware flights and indicates that tracking errors and follower dynamics can erode the geometric safety margin available at the planning level. These observations motivate the future integration of formation-level safety constraints with runtime collision-avoidance mechanisms and more conservative deployment margins.

Hardware RMSE is not consistently higher than in MuJoCo because the physical and simulated executions do not isolate vehicle dynamics alone. Overall, the experiments demonstrate the hardware executability and tracking fidelity of the affine leader–follower deployment pipeline while also identifying inter-UAV separation as a practical sim-to-real limitation.

VI. Conclusion

Despite the current success, several limitations of ATLAS remain. The current experiments primarily consider planar formation planning at a fixed flight altitude, while extension to full 3D environments remains for future work. We will extend the search-to-policy framework to 3D affine formation planning, allowing UAVs to adapt their position, orientation, and formation geometry in three-dimensional obstacle-constrained environments. For physical deployment, we will further integrate runtime separation mechanisms with formation-level safety constraints to mitigate tracking errors and reduced inter-UAV margins. Other promising directions include dynamic-obstacle avoidance, dynamics-aware planning, and validation on larger physical swarms.

In summary, we propose ATLAS, a safety-constrained search-to-policy framework for scalable multi-UAV formation planning. ATLAS combines structured action-space reduction, geometric safety filtering, offline beam-search demonstrations, and reinforcement-learning refinement to achieve efficient formation-level decision making. Simulation results demonstrate improved planning reliability, reduced computational complexity, and scalability across different swarm sizes. Physical Crazyflie experiments further demonstrate the feasibility of the affine leader–follower deployment while revealing practical sim-to-real safety limitations.

References

[1] A. Sojo, A. Perea, M. Castell, J. Perrela, I. Maza, A. Caballero, and A. Ollero, "Multi-uav planning in search and rescue missions using optimal search effort allocation," in *2025 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, 2025, pp. 649–656.

[2] J. Guo, Y. Wan, J. Wang, A. Ma, and Y. Zhong, "Disasterkd: Frequency-guided cross-decoder knowledge distillation for uav real-time disaster damage assessment," *Pattern Recognition*, p. 112968, 2025.

[3] I. Chandran and K. Vipin, "Multi-uav networks for disaster monitoring: challenges and opportunities from a network perspective," *Drone Systems and Applications*, vol. 12, pp. 1–28, 2024.

[4] Y. Liu and R. Bucknall, "A survey of formation control and motion planning of multiple unmanned vehicles," *Robotica*, vol. 36, no. 7, pp. 1019–1047, 2018.

[5] H. Rastgoftar, *Continuum Deformation of Multi-Agent Systems*. Springer, 2016.

[6] Y. Liu, Z. Liu, G. Wang, C. Yan, X. Wang, and Z. Huang, "Flexible multi-uav formation control via integrating deep reinforcement learning and affine transformations," *Aerospace Science and Technology*, vol. 157, p. 109812, 2025.

[7] X. Chang, B. Li, and T. Wang, "Consensus-based formation control for heterogeneous multi-agent systems in complex environments," *Drones*, vol. 9, no. 3, p. 175, 2025. [Online]. Available: https://www.mdpi.com/2504-446X/9/3/175

[8] J. Xu, Y. Wang, and W. Zhang, "Containment control for high-order heterogeneous continuous-time multi-agent systems with input nonconvex constraints," *Mathematics*, vol. 13, no. 3, p. 509, 2025.

[9] C. P. Bechlioulis, "Robust distributed containment control with adaptive performance and collision avoidance for multi-agent systems," *Electronics*, vol. 13, no. 8, p. 1439, 2024.

[10] W. Xiao, S. Bharadwaj, K.-C. Hsu, A. Hashemi, and A. D. Ames, "On feasibly enforcing safety in control barrier functions," in *2021 American Control Conference (ACC)*. IEEE, 2021, pp. 3400–3407.

[11] M. Selim, A. Alanwar, S. Kousik, G. Gao, M. Pavone, and K. H. Johansson, "Safe reinforcement learning using black-box reachability analysis," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 665–10 672, 2022.

[12] H. Rastgoftar, E. M. Atkins, and I. V. Kolmanovsky, "Scalable vehicle team continuum deformation coordination with eigen decomposition," *IEEE Transactions on Automatic Control*, vol. 67, no. 5, pp. 2514–2521, 2021.

[13] H. Rastgoftar and I. Kolmanovsky, "A spatio-temporal reference trajectory planner approach to collision-free continuum deformation coordination," *Automatica*, vol. 140, p. 110255, 2022.

[14] 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.

[15] S. Zhang, O. So, M. Black, Z. Serlin, and C. Fan, "Solving multi-agent safe optimal control with distributed epigraph form marl," *arXiv preprint arXiv:2504.15425*, 2025.

[16] G. Varricchione, N. Alechina, M. Dastani, G. De Giacomo, B. Logan, and G. Perelli, "Pure-past action masking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 19, 2024, pp. 21 646–21 655.

[17] Y. U. Ciftci, D. Chiu, Z. Feng, G. S. Sukhatme, and S. Bansal, "Safe-gil: Safety guided imitation learning for robotic systems," 11 2024. [Online]. Available: http://arxiv.org/abs/2404.05249

[18] T. Wu, G. Xu, Z. Wang, J. Lin, T. Chen, Y. Wu, Z. Han, Z. Liu, and F. Gao, "Precise aggressive aerial maneuvers with sensorimotor policies," 4 2026. [Online]. Available: http://arxiv.org/abs/2604.05828

[19] R. Negrinho, M. Gormley, and G. J. Gordon, "Learning beam search policies via imitation learning," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[20] A. A. Rusu, S. G. Colmenarejo, C. Gulcehre, G. Desjardins, J. Kirkpatrick, R. Pascanu, V. Mnih, K. Kavukcuoglu, and R. Hadsell, "Policy distillation," *arXiv preprint arXiv:1511.06295*, 2015.

[21] G. Mazmanyan and H. Rastgoftar, "Decentralized affine transformation for scalable and safe multi-agent aerial coordination," *Aerospace Science and Technology*, p. 113539, 2026.

[22] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.