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

Temporal Action Representation Learning for Tactical Resource Control and Subsequent Maneuver Generation

Hoseong Jung, Sungil Son, Daesol Cho, Jonghae Park, Changhyun Choi, H. Jin Kim*

***Abstract*— Autonomous robotic systems should reason about resource control and its impact on subsequent maneuvers, especially when operating with limited energy budgets or restricted sensing. Learning-based control is effective in handling complex dynamics and represents the problem as a hybrid action space unifying discrete resource usage and continuous maneuvers. However, prior works on hybrid action space have not sufficiently captured the causal dependencies between resource usage and maneuvers. They have also overlooked the multi-modal nature of tactical decisions, both of which are critical in fast-evolving scenarios. In this paper, we propose TART, a Temporal Action Representation learning framework for Tactical resource control and subsequent maneuver generation. TART leverages contrastive learning based on a mutual information objective, designed to capture inherent temporal dependencies in resource-maneuver interactions. These learned representations are quantized into discrete codebook entries that condition the policy, capturing recurring tactical patterns and enabling multi-modal and temporally coherent behaviors. We evaluate TART in two domains where resource deployment is critical: (i) a maze navigation task where a limited budget of discrete actions provides enhanced mobility, and (ii) a high-fidelity air combat simulator in which an F-16 agent operates weapons and defensive systems in coordination with flight maneuvers. Across both domains, TART consistently outperforms hybrid-action baselines, demonstrating its effectiveness in leveraging limited resources and producing context-aware subsequent maneuvers.**

I. INTRODUCTION

Autonomous robotic systems are required to operate under limited resources (e.g., task allocation, computational budget, and battery capacity), making efficient resource utilization a critical requirement for real-world deployment [1]–[5]. This challenge is particularly critical in dynamic environments, where agents must make rapid decisions in response to fast-changing situations [4], [5]. In this work, we consider a tactical decision-making problem by viewing resource usage as part of robotic actions, where each action incurs a finite resource cost. Effective control over such resource-usage actions must be coordinated with the generation of subsequent maneuvers, as each decision determines not only how resources are consumed but also how their effects can be maximized through follow-up actions.

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)] and Hyundai Motor Chung Mong-Koo Foundation.

Hoseong Jung, Jonghae Park, Changhyun Choi, and H. Jin Kim are with Seoul National University, Republic of Korea. Sungil Son is with Seoul National University and Life Assistant Robotics, Republic of Korea. Daesol Cho is with the Georgia Institute of Technology, USA.

*Corresponding author

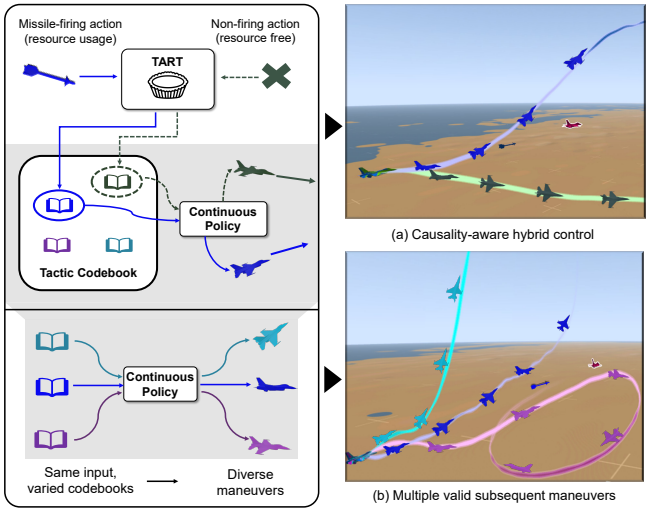


Fig. 1: Snapshots of tactical decision-making in an air combat scenario. (a) A discrete action (e.g., weapon release) both constrains the set of feasible follow-up maneuvers (*causal dependency*) and (b) gives rise to multiple valid maneuver modes (*multi-modality*). TART is designed to capture these temporal dependencies and multi-modal outcomes, conditioning the policy to select context-appropriate maneuvers.

Addressing tactical decision-making under resource constraints requires explicit modeling of hybrid action spaces that combine resource usage with maneuver control. These hybrid structures naturally arise in practice: discrete actions govern resource usage (e.g., weapon release in Fig. 1), while continuous actions guide low-level behaviors such as motion control [6]. Although reinforcement learning (RL) has been widely applied in these domains [6]–[10], existing approaches often overlook two aspects that are particularly critical in tactical decision-making: *causal dependency* and *multi-modality*. Causal reasoning is necessary to predict how resource usage constrains future maneuvers. Multi-modality is essential in dynamic environments such as air combat or mobile robot navigation, where the same discrete decision may lead to multiple valid follow-up maneuvers depending on the evolving situation (Fig. 1). Conventional RL approaches neglect these properties, thereby converging to a single dominant policy and limiting tactical flexibility [11].

These challenges highlight the need for temporal representation learning of robotic actions that can jointly capture resource usage, causal effects, and maneuver diversity. This necessity arises from the hierarchical structure of decision-making, where discrete resource-allocation decisions are temporally coupled with continuous maneuver controls. Representation learning provides a means to abstract these interactions into structured latent embeddings, enabling policies

to reason over temporal dependencies [12]–[14] while preserving the diversity of valid maneuver outcomes [15]–[17]. Such compact representations are largely absent in prior RL approaches to scalable and resource-aware robotic decision-making, underscoring the importance of temporally grounded action representations.

In this paper, we propose **T**emporal **A**ction **R**epresentation learning for **T**actical resource control and subsequent maneuver generation (TART), a framework for learning temporally grounded representations for hybrid action policies. The core idea is to model the conditional distribution of continuous maneuvers given recent state history with the current discrete resource usage decision. We realize this by maximizing a mutual information objective via a trajectory-level contrastive loss that aligns matched context–future pairs and separates mismatched pairs [18]. The resulting context embeddings are vector-quantized into a compact, interpretable codebook of tactical modes [19]. These codes condition a factorized hybrid policy where discrete actions feed into the representation to select a tactical mode, and the resulting mode guides a continuous actor to produce a multi-modal maneuver distribution. Our representation learning framework integrates into the standard on-policy RL loop, being updated jointly with policy optimization [20]. By grounding action representations temporally and structuring them into discrete tactical modes, TART preserves maneuver multi-modality while enforcing resource-aware consistency under hybrid action spaces.

We evaluate TART in two domains that combine resource constraints with hybrid action spaces. The first is a maze navigation task where agents have a limited budget of discrete boost and wall penetration actions that enhance mobility alongside continuous movement. The second is a high-fidelity air-to-air combat simulator where an F-16 agent coordinates continuous flight control with discrete weapon and defense system deployment. Across both domains, TART outperforms standard hybrid-action baselines and generates behaviors that capture causal dependencies and support multi-modal tactical decisions. These results underscore the importance of structured action representations that encode temporal dependencies, enabling resource-aware robotic decision-making.

The contributions of our work are as follows:

- We introduce TART, a temporal action representation learning framework that unifies resource control and maneuver generation to address tactical decision-making under resource limits in hybrid action spaces.
- We propose a mutual information-based objective with trajectory-level contrastive learning and a vector-quantized codebook, resulting in representations that capture causal dependencies and support multi-modal, temporally coherent behaviors.
- We conduct extensive experiments in both maze navigation and air-to-air combat environments, demonstrating (i) superior performance over hybrid-action baselines, (ii) improved modeling of maneuver multi-modality and temporal coherence in hybrid action sequences.

II. Background and Related Work

A. Parameterized Action Markov Decision Process

In this paper, we build on a Parameterized Action Markov Decision Process (PAMDP) $\langle \mathcal{S}, \mathcal{X}, P, \gamma, \mathcal{R} \rangle$, defined with a state space $\mathcal{S}$, parameterized action space $\mathcal{X}$, transition function $P : \mathcal{S} \times \mathcal{X} \times \mathcal{S} \to \mathbb{R}$, discount factor $\gamma \in [0, 1)$, and reward function $\mathcal{R} : \mathcal{S} \times \mathcal{X} \to \mathbb{R}$ [7]. Specifically, we extend the standard PAMDP framework to incorporate a discrete-continuous hybrid action space defined as Cartesian product $\mathcal{X} = \mathcal{A}_d \times \mathcal{A}_c$, where $\mathcal{A}_d = \{a_{d,1}, ..., a_{d,m}\}$ is a finite set of m discrete resource decisions and $\mathcal{A}_c \subseteq \mathbb{R}^n$ is the n-dimensional continuous control space. A hybrid action is denoted $a = (a_d, a_c)$. The agent maximizes the expected discounted return $J(\pi) = \mathbb{E}_\pi[\sum_{t=0}^{\infty} \gamma^t \mathcal{R}(s_t, a_t)]$, where the policy factorizes as $\pi(a_d, a_c|s) = \pi_d(a_d|s)\pi_c(a_c|s, a_d)$. We recall the value functions, $V^\pi(s) := \mathbb{E}_\pi[\sum_{t\geq 0} \gamma^t \mathcal{R}(s_t, a_t)|s_0 = s]$ and $Q^\pi(s, a) := \mathcal{R}(s, a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V^\pi(s')]$. The optimal value functions are denoted as $V^*(s) = \sup_\pi V^\pi(s)$ and $Q^*(s, a) = \sup_\pi Q^\pi(s, a)$.

B. Reinforcement Learning for Hybrid Action Spaces

Hybrid action spaces commonly arise in practical domains such as traffic signal control [21], financial trading [22], robotics [23], [24], where agents must coordinate discrete and continuous actions. This formulation is also natural for resource-control problems, where discrete resource-usage decisions must be integrated with continuous maneuvering. These settings have motivated RL methods tailored to hybrid actions, including extensions of Q-learning [7], [8] and actor-critic approaches [6], [9]. More recently, Li *et al.* proposed HyAR [10], which uses a conditional Variational Autoencoder (cVAE) to embed single-step hybrid actions and reduce redundancy in the enlarged action space. However, single-step embeddings cannot capture long-horizon tactical dependencies, motivating temporally grounded action representations that couple discrete resource control with continuous maneuver generation under resource limits.

C. Representation Learning for Reinforcement Learning

Representation learning in RL has primarily focused on learning compact state representations for sample efficiency, often through self-supervised objectives such as contrastive learning [13], [18], [25]. Beyond state encoding, several methods learn embeddings of state-action pairs [25], [26]. For example, TACO [26] maximizes mutual information between state-action pairs and future states to obtain action embeddings. Another key problem is capturing behavioral multi-modality, where the same context can lead to diverse plausible futures [17]. To address this, another line of work discretizes trajectories into vector-quantized codebooks, encouraging multi-modal behaviors [16], [19]. However, these approaches do not capture the temporal coupling between a discrete resource-usage decision and the distribution over subsequent continuous maneuvers in hybrid action spaces.

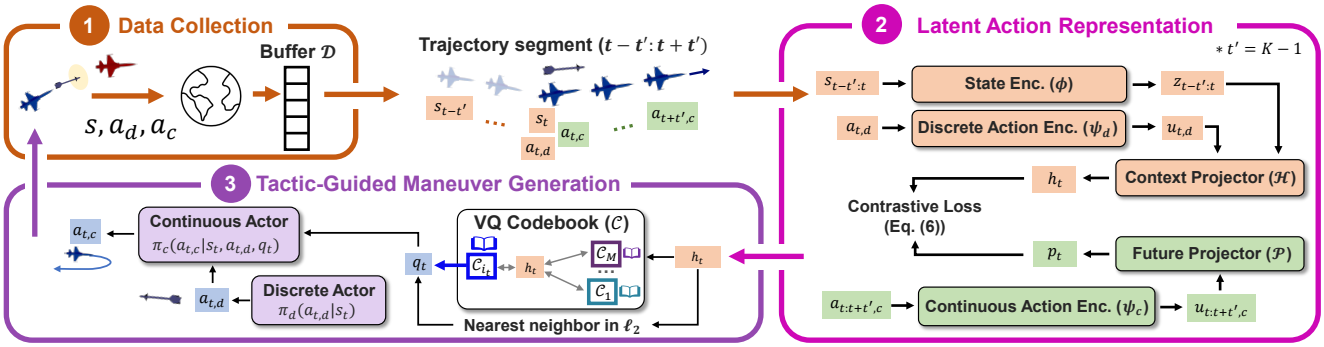


Fig. 2: Overview of TART: (1) The agent interacts with the environment and collects a set of trajectories. (2) A mutual information objective guides the clustering of given trajectories into multiple tactical modes through contrastive learning (Sec. III-B). (3) The resulting distinct modes are then mapped to discrete vectors via vector quantization (VQ). The continuous actor distinguishes between the modes and generates multi-modal maneuver distributions accordingly (Sec. III-C).

III. METHODS

This section mainly introduces TART, a framework for learning discrete-continuous (hybrid) tactical representations that condition a policy to produce multi-modal hybrid actions. TART learns via mutual information-based objectives with a practical contrastive loss to align temporal state-action representations with future maneuver sequences, enabling effective use of limited resources. The disentangled representations are transformed into discrete, interpretable codebooks, which condition a policy network to generate multi-modal maneuver distributions. Fig. 2 provides an overview of the proposed method.

A. Objective for Temporal Action Representation Learning

We begin by presenting the temporal action representation learning objective of TART. The guiding principle of our method is to learn state and action representations that capture temporal dependencies essential for effective control under hybrid actions and resource constraints. We quantify statistical dependence using mutual information (MI), denoted $\mathcal{I}(X;Y)$, which is a reparameterization-invariant measure of dependency:

$$\mathcal{I}(X;Y) = \mathbb{E}_{p(x,y)} \log[\frac{p(x,y)}{p(x)p(y)}] = H(X) - H(X|Y), \quad (1)$$

where X and Y represent either raw samples or stochastic representations from a data distribution.

We define the state embedding $z_t = \phi(s_t)$ and the hybrid action embeddings $u_{t,d} = \psi_d(a_{t,d})$ and $u_{t,c} = \psi_c(a_{t,c})$ for the discrete and continuous components, where s_t, $a_{t,d}$, and $a_{t,c}$ denote the raw state and hybrid action components. Here, ϕ and ψ_d, ψ_c serve as encoders for the state s_t, and the action components $a_{t,d}$ and $a_{t,c}$, respectively. An MI objective is adopted over a fixed horizon K:

$$\mathbb{J}_{\text{TART}} = \mathcal{I}(u_{t:t+K-1,c}; [z_{t-K+1:t}, u_{t,d}]). \quad (2)$$

Intuitively, maximizing $\mathbb{J}_{\text{TART}}$ preserves the predictive information linking the state history (including the current state) and the discrete decision to the future continuous maneuver sequence, thereby promoting temporally coherent control.

Eq. (2) aims to learn state-action representations sufficient for optimal value estimation in hybrid action spaces. To formalize this property targeted by TART, we extend the Q^*-sufficiency notions introduced by [27] to hybrid action settings. Intuitively, joint Q^*-sufficiency is a representational property: if (ϕ, ψ_d, ψ_c) is jointly Q^*-sufficient, then the optimal action-value Q^* can be evaluated without loss of information in the representation space.

Definition 1: (Joint Q^*-sufficiency for hybrid actions). Let $\phi : \mathcal{S} \to \mathcal{Z}$, $\psi_d : \mathcal{A}_d \to \mathcal{U}_d$, and $\psi_c : \mathcal{A}_c \to \mathcal{U}_c$ be state, discrete-action, and continuous-action encoders. For a set of reward functions $\mathcal{R}$, the triplet (ϕ, ψ_d, ψ_c) is jointly Q^*-sufficient w.r.t. $\mathcal{R}$ if $\forall r \in \mathcal{R}, s_1, s_2 \in \mathcal{S}, a_1, a_2 \in \mathcal{X}$,

$$\phi(s_1) = \phi(s_2),\ \psi_d(a_{1,d}) = \psi_d(a_{2,d}),\ \psi_c(a_{1,c}) = \psi_c(a_{2,c}) \Rightarrow Q_r^*(s_1, (a_{1,d}, a_{1,c})) = Q_r^*(s_2, (a_{2,d}, a_{2,c})).$$

Equivalently, there exists a measurable $\tilde{Q}$ such that $Q_r^*(s,a) = \tilde{Q}(\phi(s), (\psi_d(a_d), \psi_c(a_c)))$.
For a generic objective $\mathbb{J}$, we define the maximizer set:

$$\Phi_{\mathbb{J}} = \underset{(\phi', \psi_d', \psi_c') \in \mathcal{F} \times \mathcal{G}_d \times \mathcal{G}_c}{\text{argmax}} \mathbb{J}(\phi', \psi_d', \psi_c'), \quad (3)$$

where $\mathcal{F}, \mathcal{G}_d$ and $\mathcal{G}_c$ denote the function classes for the state, discrete-action, and continuous-action encoders. Then an objective $\mathbb{J}$ is jointly Q^*-sufficient with respect to $\mathcal{R}$ if every maximizer $(\phi, \psi_d, \psi_c) \in \Phi_{\mathbb{J}}$ satisfies Definition 1. In our method, $\mathbb{J}$ is instantiated as Eq. (2); under standard assumptions on the PAMDP and consistent MI estimation, this objective promotes representations approaching joint Q^*-sufficiency for value estimation in hybrid action settings. Consequently, there exist measurable $\tilde{Q}, \tilde{V}$ that factor through (ϕ, ψ_d, ψ_c), justifying their use for actor-critic training built on these representations.

B. Learning a Latent Action Representation

Since direct computation of mutual information is typically intractable, we instead optimize a tractable lower bound using InfoNCE [18]:

$$\mathcal{I}(X;Y) \geq \log(N) - \mathcal{L}_{\text{NCE}}, \quad (4)$$

where N denotes the number of samples and $\mathcal{L}_{\text{NCE}}$ is the InfoNCE loss. To instantiate this bound for $\mathbb{J}_{\text{TART}}$, we form pairs between a context summarizing the state history and the discrete action, and a future continuous maneuver descriptor.

Let the context projector $\mathcal{H}$ and the future projector $\mathcal{P}$ map to a common embedding space:

$$h_t = \mathcal{H}([z_{t-K+1:t}, u_{t,d}]), \quad p_t = \mathcal{P}(u_{t:t+K-1,c}). \tag{5}$$

Given an anchor representation h_t, the positive sample is p_t and the negatives are the in-batch items $\mathbf{v} \in \mathcal{N}_t$, where $\mathcal{N}_t := \{\mathcal{P}(u^{(n)}_{s:s+K-1,c}) | (n,s) \neq (\text{current traj}, t)\}$ denotes embeddings from other time steps or trajectories in the batch. The InfoNCE loss encourages the anchor to be similar to the positive while dissimilar to the negatives:

$$\mathcal{L}_{\text{NCE}} = -\log \frac{\exp(\text{sim}(h_t, p_t))}{\exp(\text{sim}(h_t, p_t)) + \sum_{\mathbf{v}\in\mathcal{N}_t} \exp(\text{sim}(h_t, \mathbf{v}))}, \tag{6}$$

where $\text{sim}(x, y) = x^\top W y/\tau$ is a learnable bilinear similarity function with parameter W and temperature $\tau > 0$. This construction directly matches the objective Eq. (2): the context h_t encodes the state trajectory and the discrete action choice, while the positive p_t encodes future continuous maneuvers over the horizon K. Inspired by TACO [26], we additionally augment positives with temporally adjacent segments from the same episode to improve sample efficiency; negatives are temporally shuffled within a batch to reduce shortcut cues.

C. Tactic-Guided Maneuver Generation

To exploit the clustered action representations obtained via contrastive learning, we aim to represent them as quantized tactical modes that guide the policy in generating diverse maneuver distributions. We instantiate these modes with a Vector Quantization (VQ) codebook [19]. Let $\mathcal{C} = \{c_1, ..., c_M\}$ denote the set of learnable codebook entries, where each $c_i \in \mathbb{R}^d$. Given a state-action segment $\tau_t = \{(s_{t'}, a_{t'}), ..., (s_t, a_t)\}$, with $t' = t - K + 1$, we obtain embeddings $\{z_{t'}, ..., z_t\}$ and $\{u_{t',d}, ..., u_{t,d}\}$, $\{u_{t',c}, ..., u_{t,c}\}$ via the encoders ϕ, ψ_d, ψ_c (Section III.A.). The context encoder used in Eq. (6) summarizes $h_t = \mathcal{H}([z_{t':t}, u_{t,d}])$ and we quantize by applying nearest neighbor in ℓ_2-distance:

$$i_t = \underset{m}{\text{argmin}} ||h_t - c_m||_2, \quad q_t = c_{i_t}. \tag{7}$$

The latent code q_t serves as a tactical mode and conditions the maneuver policy. We factorize the hybrid policy as:

$$\pi_\theta(a_{t,d}, a_{t,c} | s_t, q_t) = \underbrace{\pi_d(a_{t,d}|s_t)}_{\text{discrete actor}} \cdot \underbrace{\pi_c(a_{t,c}|s_t, a_{t,d}, q_t)}_{\text{continuous actor}}. \tag{8}$$

At inference, the pipeline is $s_t \xrightarrow{\pi_d} a_{t,d}$, then h_t is encoded and quantized to q_t, and finally $\pi_c(a_{t,c}|s_t, a_{t,d}, q_t)$ generates the continuous maneuver; thus the discrete actor precedes, then the continuous actor is tactic-guided.

To learn effective quantized representations, we employ the standard vector quantization objective, consisting of a reconstruction loss and a commitment loss:

$$\mathcal{L}_{\text{VQ}} = ||h_t - \hat{h}_t||^2 + ||\text{sg}[h_t] - c_{i_t}||^2 + \beta||h_t - \text{sg}[c_{i_t}]||^2, \tag{9}$$

Algorithm 1: TART-PPO

Init actor π_d, π_c and critic V_ω
Init encoders ϕ, ψ_d, ψ_c and projectors $\mathcal{H}, \mathcal{P}$
Init codebook $C = \{c_1, ..., c_M\}, c_i \sim \mathcal{N}(0, \sigma^2 I_d)$
Prepare replay buffer $\mathcal{D}$
repeat Stage 1: Warm-up
  Sample batch $\mathcal{B}$ from replay buffer $\mathcal{D}$
  Update $\phi, \psi_d, \psi_c, \mathcal{H}, \mathcal{P}$ using Eqs. (6) and (9)
until *reaching maximum warm-up steps*;
repeat Stage 2: Main loop
  for $t \leftarrow 1$ *to* T **do**
    // select discrete tactical modes to guide policy
    observe s_t; $a_{t,d} \sim \pi_d(\cdot|s_t), z_t = \phi(s_t)$,
    $u_{t,d} = \psi_d(a_{t,d}), h_t = \mathcal{H}([z_{t-K+1:t}, u_{t,d}])$
    $i_t = \text{argmin}_m ||h_t - c_m||_2, q_t = \text{sg}[c_{i_t}]$
    // generate continuous maneuvers
    $a_{t,c} \sim \pi_c(\cdot|s_t, a_{t,d}, q_t)$
    Execute $a_{t,d}, a_{t,c}$, observe r_t and new state s_{t+1}
    Store $\{s, a_{t,d}, a_{t,c}, r, s_{t+1}, q_t, \log \pi_d, \log \pi_c\}$ in $\mathcal{D}$
    Sample a mini-batch of N experiences from $\mathcal{D}$
    Update π_d, π_c, V_ω with the PPO objective [20]
  repeat
    Update $\phi, \psi_d, \psi_c, \mathcal{H}, \mathcal{P}$ using Eqs. (6) and (9)
  until *reaching maximum representation training steps*;
until *reaching maximum total environment steps*;

where $\hat{h}_t$ is the reconstruction of the embedding from the codebook, $\text{sg}[\cdot]$ denotes the stop-gradient operator, and $\beta > 0$ is a hyperparameter that balances the commitment strength. The VQ encoder is optimized to produce embeddings h_t that closely match their assigned codebook entries, while the codebook itself adapts to reflect the encoder outputs.

D. Training Protocol and Overall Objective

We train TART with a unified on-policy workflow that interleaves representation learning and policy optimization, as shown in Algorithm 1. The procedure consists of two stages. In the warm-up stage, we collect exploratory rollouts and fit the representation modules by minimizing the loss function in Eqs. (6) and (9). This yields latent codes that capture tactical structure before RL training.

In the main loop, PPO [20] optimizes the factorized hybrid policy. We implement the critic V_ω with a shared state backbone and two value heads. The discrete head depends on the state s_t and provides a baseline for the discrete actor $\pi_d(a_{t,d}|s_t)$. The continuous head takes the state s_t, the chosen discrete action a_d, and the tactical code q_t to support the continuous actor $\pi_c(a_{t,c}|s_t, a_{t,d}, q_t)$. The shared critic backbone improves sample efficiency, while the distinct heads provide actor-specific baselines that reduce variance, stabilize training, and capture multi-modality.

The overall loss function is as follows:

$$\mathcal{L}_{\text{total}} = \underbrace{\mathcal{L}_{\text{RL}}}_{\text{PPO for } \pi_d, \pi_c} + \lambda_{\text{NCE}}\mathcal{L}_{\text{NCE}} + \lambda_{\text{VQ}}\mathcal{L}_{\text{VQ}}, \tag{10}$$

where $\mathcal{L}_{\text{NCE}}$ and $\mathcal{L}_{\text{VQ}}$ are the InfoNCE loss and VQ loss as in Eqs. (6)) and (9). λ_{NCE} and λ_{VQ} are balancing parameters. $\mathcal{L}_{\text{RL}}$ is the standard PPO objective with a shared critic, entropy regularization, and separate discrete and continuous heads; gradients from $\mathcal{L}_{\text{RL}}$ do not update the codebook.

IV. Evaluation Environments

In this section, we introduce two budgeted hybrid-action environments to evaluate TART: (A) a maze navigation task where the agent can utilize limited resources to enhance mobility, and (B) a high-fidelity air combat environment with weapon and defense systems. The overview of each environment is shown in Fig. 3.

A. Maze Navigation

1) Task & Actions: We extend POGEMA [28] to a budgeted hybrid-action maze navigation task with a single start-goal pair and horizon $T = 100$. At each timestep t, the policy outputs a continuous heading vector $a_c = (x_t, y_t) \in [-1, 1]^2$, which is discretized into the nearest cardinal direction. The environment executes one grid move accordingly. A discrete option $a_d \in \{\text{NOOP}, \text{PENETRATION}, \text{BOOST}\}$ is available with two uses per episode, and each activation lasts two steps. BOOST performs two sequential sub-moves along the chosen heading, and we check for collisions after each sub-move. PENETRATION allows the agent to traverse at most one wall cell per sub-move. If PENETRATION expires inside a wall, the agent is relocated to the nearest free cell.

2) States & Rewards: We adopt the Learn-to-Follow [29] settings to reduce exploration complexity. In this setup, the policy receives a waypoint sequence computed by a heuristic path decider as an auxiliary input. The state consists of the local observation, active waypoint, and remaining option budget. The local observation is a two-channel egocentric $m \times m$ tensor centered on the agent, where m denotes the observation range. One channel encodes walls and the waypoints, while the other encodes background agent positions. The reward combines sparse goal-reaching terms and dense shaping with a constant timestep penalty, waypoint rewards, collision penalties, and costs for option activations.

3) Scenarios & Metrics: Evaluation is conducted on three maze scenarios (Fig. 3(a)-(c)) with increasing difficulty.

- *Easy*: 10×10 simple mazes with a trivial shortest path.
- *Medium*: 20×20 mazes with complex layouts.
- *Hard*: 20×20 non-stationary mazes with background agents that navigate with A* toward randomly sampled goals, which creates congestion and blockages around narrow passages.

Each difficulty includes ten distinct maze layouts to promote generalization. We report *Success Rate*, *Time-to-Goal* (**TTG**) in decision steps with timeouts set to the episode horizon $T = 100$, and *Occupancy Coverage*, measured as the fraction of unique free cells visited to reflect path efficiency beyond goal completion.

4) Training Details: We follow the training details and network architecture of Learn-to-Follow [29]. The agent uses a ResNet spatial encoder and MLP heads for the policy and critic, comprising approximately 5M parameters. Action masking prevents invalid moves and option activations that violate budget or duration. For each difficulty, the agent is trained jointly on ten maps and evaluated on unseen maps to assess generalization. The final policy is trained for 20 million steps, and evaluation uses the best checkpoint.

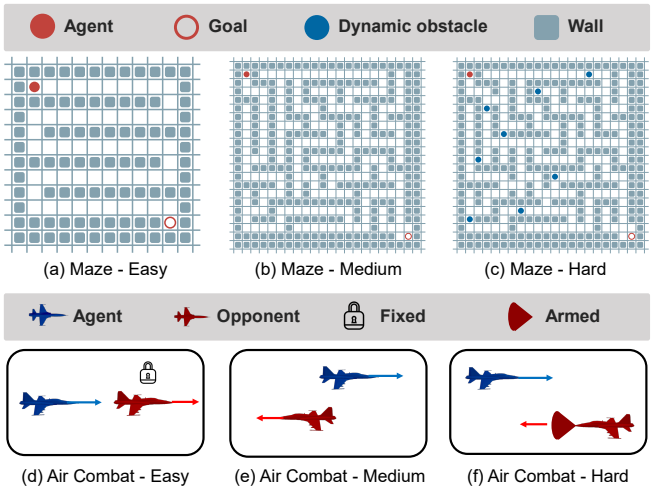


Fig. 3: Overviews and difficulty settings of the evaluation environments. (a)–(c) Maze Navigation: *Easy* (trivially solvable), *Medium* (complex), *Hard* (dynamic obstacles). (d)–(f) Air-to-Air Combat: *Easy* (fixed-maneuver opponent), *Medium* (unarmed evasive opponent), *Hard* (armed pursuing opponent).

B. Air-to-Air Combat

1) Task & Actions: We build a high-fidelity air-to-air combat environment on top of Light Aircraft Game [30] and NeuralPlane [31]. The platform models an F-16 aircraft with six-degree-of-freedom aerodynamics in JSBSim [32], equipped with short-range missiles, a gun, and defensive countermeasures. Defensive countermeasures can neutralize an incoming missile within their effective envelope. The objective is to eliminate the opponent before being defeated.

At each timestep t, the agent selects a discrete option $a_d \in \{\text{NOOP}, \text{MISSILE}, \text{GUN}, \text{DEFENSE}\}$ and continuous controls $a_c \in [0, 1]^4$ corresponding to aileron, elevator, rudder and throttle. Each episode provides a budget of five missile and five defensive countermeasures, while the gun is unrestricted but constrained by strict firing conditions [33]. Missile launch requires a valid lock-on defined by range and aspect [30], and is effective only if the lock persists after firing. Gun usage follows the engagement envelope of [33], and defensive countermeasures are effective only within a circular envelope defined as twice the missile radius. Invalid action requests are masked without consuming the budget. The simulator runs at 10 Hz with episodes last at most three minutes, giving a maximum horizon of $T = 1800$ steps. Episodes terminate upon opponent elimination, ownship loss (including crash), or timeout.

2) States & Rewards: Following prior work [34], the state includes ownship kinematics (attitude, altitude, velocity, acceleration) and ego-opponent geometry (range, relative aspect), extended with remaining resources for missiles and defensive countermeasures. The reward function consists of a sparse term for opponent elimination with dense shaping. Tail-chase shaping [34] rewards maintaining favorable pursuit geometry behind the opponent. Penalties apply for being shot down or crashing at low altitude, and for each missile or countermeasure used to encourage efficiency.

TABLE I: Performance (*Success Rate*) for different PAMDP methods over resource-limited environments.
Results are averaged over five random seeds, $\pm$ indicates the standard deviation across seeds, and bold entries denote the best performance.

Task/ Scenario		TART	PADDPG [6]	PDQN [8]	HPPO [9]	HyAR [10]	TART w/o $\mathcal{L}_{\text{NCE}}$	TART w/o $\mathcal{L}_{\text{VQ}}$
Maze Navigation	Easy	$\mathbf{97.2 \pm 1.3}$	88.2 ± 6.9	85.8 ± 6.1	87.2 ± 4.6	94.5 ± 3.2	92.8 ± 3.3	95.4 ± 1.9
	Medium	$\mathbf{90.8 \pm 4.2}$	74.2 ± 6.1	68.8 ± 6.2	74.6 ± 9.0	80.6 ± 8.5	89.0 ± 3.5	88.2 ± 5.3
	Hard	$\mathbf{72.8 \pm 9.9}$	38.8 ± 10.5	38.4 ± 13.5	46.2 ± 6.7	60.4 ± 9.1	69.2 ± 5.7	67.0 ± 3.2
Air-to-Air Combat	Easy	$\mathbf{94.8 \pm 3.1}$	79.6 ± 5.6	81.2 ± 5.9	87.8 ± 4.6	86.6 ± 4.8	92.4 ± 5.1	91.8 ± 2.2
	Medium	$\mathbf{90.8 \pm 4.2}$	68.6 ± 5.9	74.2 ± 4.3	76.2 ± 4.5	77.6 ± 6.3	87.0 ± 6.2	89.8 ± 2.6
	Hard	$\mathbf{76.8 \pm 5.0}$	61.8 ± 7.8	57.2 ± 7.4	65.4 ± 3.0	64.4 ± 7.2	73.6 ± 6.2	70.8 ± 2.1
Average Success Rate		**87.2**	68.5	67.6	72.9	77.4	84.0	83.8

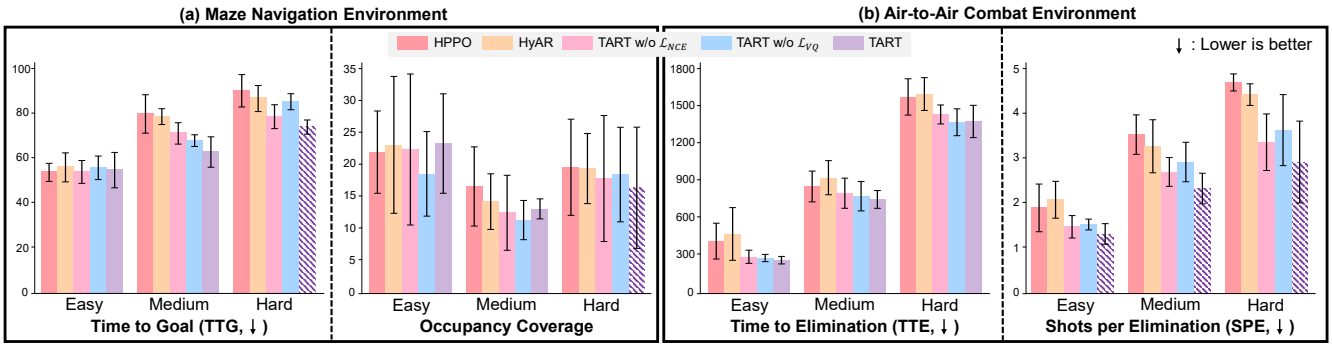


Fig. 4: Experimental results across the designed environments and metrics. Values are averaged over five seeds and black bars indicate standard deviation. For failed episodes, TTG and TTE are set to their maximum values (100 and 1800, respectively), while SPE is reported only for successful episodes. Hatched bars indicate the results discussed in the text and exhibiting superior performance.

3) Scenarios & Metrics: We evaluate agents in three air-to-air combat scenarios (Fig. 3(d)-(f)) of increasing difficulty, defined by opponent capability and initial geometry.

- *Easy*: Scripted opponent [30] with simple pursuit and evade behavior and no weapons or countermeasures. Episodes start in a pursuit geometry with the ownship behind the opponent.
- *Medium*: Scripted opponent [30] equipped with reactive weapons and countermeasures. Episodes start in neutral geometry with opposite headings, leading to a merge after roughly one turn.
- *Hard*: A single-agent variant of the learned pursuer [35] equipped with reactive weapons and countermeasures. Episodes start in neutral with an immediate merge.

Evaluation metrics are *Success Rate*, *Time-to-Elimination* (**TTE**) measured in decision steps with a timeout $T = 1800$, and *Shots-per-Elimination* (**SPE**) defined as the number of missiles required to eliminate the opponent. *Success Rate* counts episodes where the agent eliminates the opponent. For defeats, **TTE** is set to the timeout and **SPE** to the maximum budget of five. If an elimination occurs by gun after all missile attempts are invalidated, **SPE** is also recorded as five.

4) Training Details: We adopt curriculum learning [34], which gradually increases task difficulty to stabilize and accelerate training. The curriculum starts with pursuit geometry and no weapons, then gradually transitions to neutral geometry while enabling weapons and defensive countermeasures. The policy and critic follow the GRU-MLP architecture of [34], trained with PPO [20]. Opponents are pre-trained and remain fixed during both training and evaluation. To ensure fairness, all baselines are trained under the same curriculum.

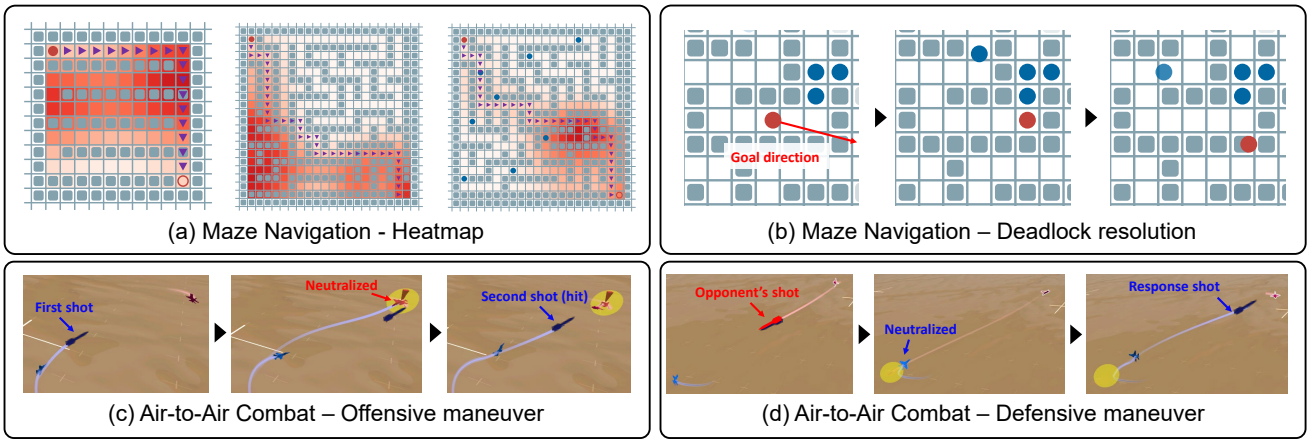
V. Experimental Results

In this section, we evaluate TART's ability to learn temporally coherent action representations that follow the hybrid action structure and capture maneuver multi-modality. We compare TART against standard PAMDP baseline methods: PADDPG [6], PDQN [8], HPPO [9], and HyAR [10]. Table I reports success rates for all baselines. Fig. 4 focuses on the two strongest PAMDP methods in our re-implementation, HPPO and HyAR, which also align with TART's multi-head actor design for hybrid actions. All baselines share identical training settings for fair comparison.

A. Comparison with PAMDP Baselines

Table I shows that TART outperforms the strongest PAMDP baseline in every scenario, with *Success Rate* gains from $+2.7$ to $+13.2$ points. Across all six scenarios, TART attains an average success rate of 87.2%, while the best baseline method (HyAR) achieves 77.4%. Fig. 4 further shows that these gains do not come at the cost of resource efficiency. In Maze Navigation, TART achieves lower or comparable **TTG**, indicating more efficient path completion. In Air-to-Air Combat, TART also demonstrates efficiency in weapon usage by yielding lower or comparable **TTE** and **SPE**. These results imply that TART leverages the fixed action budget more effectively with coordinated continuous maneuvers, as further evidenced by the qualitative analysis in Section V-C.

1) Maze Navigation: Since *Easy* and *Medium* do not contain dynamic obstacles, all methods yield *Occupancy Coverage* near the shortest-path value. The primary difference is efficiency, where TART demonstrates lower **TTG**



(a) Maze Navigation - Heatmap
(b) Maze Navigation – Deadlock resolution
(c) Air-to-Air Combat – Offensive maneuver
(d) Air-to-Air Combat – Defensive maneuver

Fig. 5: Representative qualitative results in (a)-(b) Maze Navigation and (c)-(d) Air-to-Air Combat. (a) Heatmaps in *Easy*, *Medium*, and *Hard* scenarios, where the triangle marks the optimal path. Red indicates higher, while white indicates lower visitation frequency. (b) Deadlock examples, where the agent employs the PENETRATION action to navigate through cluttered corridors. (c) An offensive maneuver: the agent consecutively launches shots using the MISSILE action. (d) A defensive maneuver: the agent neutralizes the opponent's missile with a DEFENSE action and responds with a counter shot.

and higher *Success Rate*, while keeping *Occupancy Coverage* near the shortest-path value. In *Hard*, background agents cause corridor congestion. TART achieves lower *Occupancy Coverage* and shorter **TTG**, which suggests that it unlocks bottlenecks with PENETRATION rather than exploring widely. The PAMDP actor-head baselines (PADDPG, PDQN, HPPO) and HyAR's cVAE approach do not couple discrete actions to subsequent maneuvers. In contrast, TART learns a temporal representation and conditions the actor on a learned code. This design triggers PENETRATION precisely when it yields progress. This behavior is consistent with the lower *Occupancy Coverage* and shorter **TTG** observed on *Hard*.

2) Air-to-Air Combat: Across all difficulty levels, TART achieves higher *Success Rate* across all difficulties than baselines while maintaining **TTE** and **SPE** similar or lower. This outcome is critical under limited-resource constraints. Although deploying more missiles could raise success, the non-increasing **SPE** shows that TART achieves gains without higher consumption. The reduced **SPE** observed in *Easy* and *Medium* reflects TART's ability to maintain lock-on after missile releases, resulting in more valid shots. In *Hard* scenario, success depends on effective use of the remaining DEFENSE actions against the armed opponent. TART employs these actions more effectively through context-aware coordination of offensive and defensive maneuvers. Overall, these findings show that temporal coupling between resource options (*Missile, Defense*) and subsequent maneuvers enables efficient resource use and effective combat performance.

B. Ablation Studies

We validate the design by ablating the contrastive loss $\mathcal{L}_{\text{NCE}}$ and the vector-quantized loss $\mathcal{L}_{\text{VQ}}$. The contrastive loss aligns discrete resource usage decisions with the following continuous trajectories. The vector-quantized loss induces a codebook supporting diverse maneuver patterns under similar discrete actions and state histories. We denote the ablations as TART w/o $\mathcal{L}_{\text{NCE}}$ and TART w/o $\mathcal{L}_{\text{VQ}}$. Removing both eliminates the learned representation and conditioning, and the method becomes close to an HPPO [9] learner.

In both Maze Navigation and Air-to-Air Combat, ablating either objective reduces *Success Rate*, whereas the full model consistently performs best. In *Hard* Maze Navigation, both **TTG** and *Occupancy Coverage* increase in ablated models, reflecting difficulty with clutter and delayed PENETRATION. In Air-to-Air Combat environment, both ablations increase **TTE** and **SPE** with identical budgets, indicating degraded launch timing and weaker follow-up maneuvers. Variance across seeds is larger for TART w/o $\mathcal{L}_{\text{NCE}}$ than for TART w/o $\mathcal{L}_{\text{VQ}}$, suggesting that multi-modality without contrastive alignment results in unstable learning with mode drift. These findings highlight the complementary roles of temporal alignment and maneuver diversity for mastering hybrid actions with limited resources under dynamic conditions.

C. Qualitative Analysis

1) Maze Navigation: Figure 5(a) shows agent visitation heatmaps in Maze Navigation across different difficulty levels. In *Easy* and *Medium*, the density concentrates near the shortest paths, while retaining some pre-convergence spread, indicating a reliable goal-reaching policy. Consistent with corridor geometry, BOOST is used early to accelerate along straight segments while staying within the per-episode budget. In *Hard*, the agent can encounter deadlocks, which can be resolved by activating the PENETRATION option at chokepoints. Fig. 5(b) further illustrates such deadlock cases. Overall, the heatmap patterns align with the quantitative metrics, showing that BOOST enables faster progress along straight corridors and PENETRATION resolves blockages under budget constraints.

2) Air-to-Air Combat: TART exhibits both strong representation ability and adaptive combat behaviors. In Fig. 1(a), maneuvers differ with or without missile launch, showing causal dependency between discrete and continuous actions. Figure 1(b) demonstrates multi-modality, where distinct VQ codes yield different maneuvers under the same conditions. Figures 5(c)-(d) illustrate adaptive maneuvers in dynamic combat scenarios. In (c), the agent executes an offensive maneuver with consecutive MISSILE actions. In (d), it performs

a defensive maneuver, neutralizing the opponent's missile with a DEFENSE action and following with a counter shot. These results show that integrating TART enables the RL agent to capture causal dependencies and multi-modality while coordinating offensive and defensive options in a context-aware manner.

VI. CONCLUSION

This paper introduces TART, a representation learning framework for reinforcement learning in hybrid action spaces, designed to enable effective resource control and maneuver generation. Across both maze navigation and air-to-air combat domains, TART consistently outperforms representative baselines, underscoring the importance of temporally grounded action representations for resource-constrained decision-making. Our study is limited to simulated environments with simplified resource constraints, modeled primarily as action budgets rather than persistent physical costs. As future work, we plan to incorporate real-world considerations such as energy consumption, communication bandwidth, and other hardware-level limitations, and extend the evaluation of TART to broader settings, including multi-agent and lifelong learning.

REFERENCES

[1] B. P. Gerkey and M. J. Matarić, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[2] M. Afrin, J. Jin, A. Rahman, *et al.*, "Resource allocation and service provisioning in multi-agent cloud robotics: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 842–870, 2021.

[3] Z. Dai, Y. Zhang, W. Zhang, *et al.*, "A multi-agent collaborative environment learning method for UAV development and resource allocation," *IEEE Transactions on Signal and Information Processing over Networks*, vol. 8, pp. 120–130, 2022.

[4] X. Hai, L. Tan, Q. Feng, *et al.*, "Replanning-oriented framework for efficient real-time decision-making in multi-UAV systems," *IEEE Transactions on Industrial Informatics*, 2025.

[5] H. Liao, Z. Li, K. Zhu, *et al.*, "SA-TP^2: A safety-aware trajectory prediction and planning model for autonomous driving," *IEEE Transactions on Robotics*, 2025.

[6] M. J. Hausknecht and P. Stone, "Deep reinforcement learning in parameterized action space," in *Proceedings of the International Conference on Learning Representations*, 2016.

[7] W. Masson, P. Ranchod, and G. Konidaris, "Reinforcement learning with parameterized actions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[8] J. Xiong, Q. Wang, Z. Yang, *et al.*, "Parametrized deep Q-networks learning: Reinforcement learning with discrete-continuous hybrid action space," *arXiv preprint arXiv:1810.06394*, 2018.

[9] Z. Fan, R. Su, W. Zhang, and Y. Yu, "Hybrid actor-critic reinforcement learning in parameterized action space," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019, pp. 2279–2285.

[10] B. Li, H. Tang, Y. Zheng, *et al.*, "HyAR: Addressing discrete-continuous action reinforcement learning via action representation," in *Proceedings of the International Conference on Learning Representations*, 2022.

[11] K. Wu, Y. Zhu, J. Li, *et al.*, "Discrete policy: Learning disentangled action space for multi-task robotic manipulation," in *Proceedings of the International Conference on Robotics and Automation*, 2025, pp. 8811–8818.

[12] Y. Ze, N. Hansen, Y. Chen, *et al.*, "Visual reinforcement learning with self-supervised 3d representations," *IEEE Robotics and Automation Letters*, vol. 8, no. 5, pp. 2890–2897, 2023.

[13] O. Biza, T. Weng, L. Sun, *et al.*, "On-robot reinforcement learning with goal-contrastive rewards," in *Proceedings of the International Conference on Robotics and Automation*, 2025, pp. 4797–4805.

[14] Q. Bu, Y. Yang, J. Cai, *et al.*, "UniVLA: Learning to act anywhere with task-centric latent actions," in *Proceedings of the Robotics: Science and Systems*, 2025.

[15] S. A. Mehta, S. Parekh, and D. P. Losey, "Learning latent actions without human demonstrations," in *Proceedings of the International Conference on Robotics and Automation*, 2022, pp. 7437–7443.

[16] J. Luo, P. Dong, J. Wu, *et al.*, "Action-quantized offline reinforcement learning for robotic skill learning," in *Proceedings of the Conference on Robot Learning,* 2023, pp. 1348–1361.

[17] S. Zhu, R. Kaushik, S. Kaski, and V. Kyrki, "Imitation-guided multimodal policy generation from behaviourally diverse demonstrations," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2022, pp. 1675–1682.

[18] A. Van Den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[19] A. Van Den Oord and O. Vinyals, "Neural discrete representation learning," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[20] J. Schulman, F. Wolski, P. Dhariwal, *et al.*, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[21] H. Luo, Y. Bie, and S. Jin, "Reinforcement learning for traffic signal control in hybrid action space," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 6, pp. 5225–5241, 2024.

[22] F. Pan, T. Zhang, L. Luo, *et al.*, "Learn continuously, act discretely: Hybrid action-space reinforcement learning for optimal execution," in *arXiv preprint arXiv:2207.11152*, 2022.

[23] M. Neunert, A. Abdolmaleki, M. Wulfmeier, *et al.*, "Continuous-discrete reinforcement learning for hybrid control in robotics," in *Proceedings of the Conference on Robot Learning*, 2020, pp. 735–751.

[24] H. Fu, K. Tang, P. Li, *et al.*, "Multi-agent reinforcement learning with hybrid action space for free gait motion planning of hexapod robots," in *Proceedings of the Annual Conference on Robot Learning*, 2024.

[25] A. Allshire, R. Martín-Martín, S. Shawn, *et al.*, "LASER: Learning a latent action space for efficient reinforcement learning," in *Proceedings of the International Conference on Robotics and Automation*, 2021, pp. 6650–6656.

[26] R. Zheng, X. Wang, Y. Sun, *et al.*, "TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 36, pp. 48203–48225, 2023.

[27] K. Rakelly, A. Gupta, C. Florensa, and S. Levine, "Which mutual-information representation learning objectives are sufficient for control?," *Advances in Neural Information Processing Systems*, vol. 34, pp. 26345–26357, 2021.

[28] A. Skrynnik, A. Andreychuk, A. Borzilov, *et al.*, "POGEMA: A benchmark platform for cooperative multi-agent pathfinding," in *Proceedings of the International Conference on Learning Representations*, 2024.

[29] A. Skrynnik, A. Andreychuk, M. Nesterova, *et al.*, "Learn to follow: Decentralized lifelong multi-agent pathfinding via planning and learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, 2024, pp. 17541–17549.

[30] Q. Liu, Y. Jiang, and X. Ma, "Light aircraft game: A lightweight, scalable, gym-wrapped aircraft competitive environment with baseline reinforcement learning algorithms," 2022, URL https://github.com/liuqh16/CloseAirCombat

[31] C. Xue, Q. Liu, *et al.*, "NeuralPlane: An efficiently parallelizable platform for fixed-wing aircraft control with reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 37, pp. 96939–96962.

[32] J. Berndt, "JSBSim: An open source flight dynamics model in C++," in *Proceedings of the AIAA Modeling and Simulation Technologies Conference and Exhibit*, 2004, pp. 4923.

[33] A. P. Pope, J. S. Ide, D. Mićović, *et al.*, "Hierarchical reinforcement learning for air combat at DARPA's AlphaDogfight Trials," *IEEE Transactions on Artificial Intelligence*, vol. 4, no. 6, pp. 1371–1385, 2022.

[34] J. Bae, H. Jung, S. Kim, *et al.*, "Deep reinforcement learning-based air-to-air combat maneuver generation in a realistic environment," *IEEE Access*, vol. 11, no. 1, pp. 26427–26440, 2023.

[35] H. He, Q. Dong, X. Shang, *et al.*, "Autonomous decision-making algorithm for multi-agent beyond-visual-range air combat," in *Proceedings of the Chinese Conference on Swarm Intelligence and Cooperative Control*, 2023, pp. 646–660.