ALLOTS: ADAPTIVE MULTI-AGENT REINFORCEMENT LEARNING WITH VARIABLE OBSERVATION AND ACTION SPACES

Anonymous authors
Paper under double-blind review

ABSTRACT

Many real-world applications involve dynamic entities, e.g., heat pumps and electric vehicles with flexible electricity consumption, which, in multi-agent reinforcement learning (MARL), lead to variable observation and action spaces. Existing MARL methods handle variability in observation or action spaces in isolation, but cannot directly generate entity-aligned action sets that adapt to a changing entity set. To address this issue, we propose the **A**daptive Mu**L**ti-Agent Reinforcement **L**earning with Variable **O**bservation and ac**T**ion **S**paces (**ALLOTS**) method that simultaneously handles variable observation and action spaces. First, we present a permutation-equivariant set-to-set actor that maps variable observations to entity-aligned variable actions, preserving valid control semantics as the entity set evolves. Second, we present a set-structured critic that encodes entity-level state–action pairs as joint tokens and bootstraps value targets from the next-step active set, enabling stable learning and efficient adaptation without retraining. Experiments on energy system, multi-agent particle, and multi-drone environments show that ALLOTS outperforms state-of-the-art methods in dealing with variable observation and action spaces. Code is available at `https://anonymous.4open.science/r/ALLOTS-EFB4`.

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) has achieved remarkable success in domains such as robotics and energy systems. However, many real-world MARL applications involve dynamic entities that join or leave over time. In multi-robot coordination, robots in a formation may be reassigned or fail, changing the set of units a coordinator must control. In energy systems, flexible devices participate intermittently based on user behavior or availability. This leads to variable observation and action spaces. As a concrete example, consider bottom-up FlexOffer (FO)-based energy management (Pedersen et al., 2018; Šikšnys et al., 2019), where a manager coordinates flexible devices such as batteries, heat pumps, and electric vehicles (EVs) by issuing continuous scheduling decisions. As illustrated in Figure 1, devices may join or leave over time due to mobility, user engagement, or failures. After such changes, the manager must output a different number of actions for the new active set, and the optimal policy may shift accordingly. This makes standard MARL pipelines with fixed observation and action spaces difficult to apply directly.

This example shows why MARL methods capable of handling variable observation and action spaces are needed. Existing methods, however, do not directly address this setting, and fall into three lines of work. The first assumes fixed observation and fixed action spaces over a static agent set, as in MADDPG (Lowe et al., 2017), QMIX (Zhang et al., 2024), and MAPPO (Yu et al., 2022). The second improves scalability under variable-size observation sets via set-based or attention-based encoders (Zaheer et al., 2017; Lee et al., 2019; Locatello et al., 2020; Iqbal & Sha, 2019). The third varies the feasible action set through action-availability or slate formulations, where the candidate pool changes but the action space remains fixed (Ie et al., 2019; Jain et al., 2021). In parallel, fast adaptation has been studied to cope with changing environments, including meta-RL (Finn et al., 2017), RL^2 (Grigsby et al., 2024), continual RL under distribution shifts (Rolnick et al., 2019), and policy consolidation across time scales (Kaplanis et al., 2019), as well as MARL under changing participation or team composition (Liu et al., 2021; Gu et al., 2022). Despite this progress, these

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

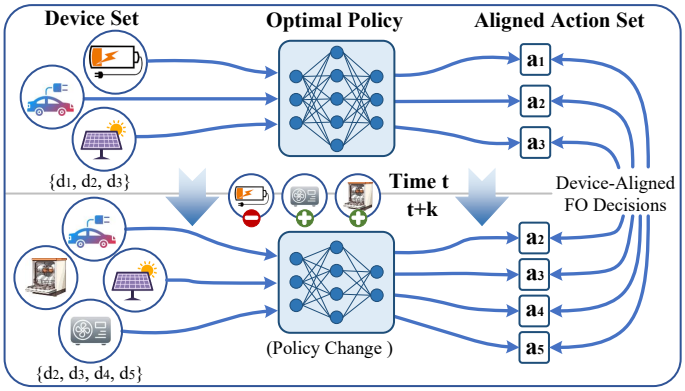

Figure 1: Dynamic participation in FO-based energy management.

methods do not directly handle the setting, where the controller must generate entity-aligned action sets that adapt to a changing entity set.

In practice, dynamic entities are a recurring condition rather than an occasional disturbance in many real-world systems. A controller designed for a fixed set of entities will either fail to issue valid decisions after entity set changes or require costly retraining to recover performance, limiting the practicality of MARL for real-world applications. This motivates methods that can adapt to variable observation and action spaces induced by a variable entity set, which raises two technical challenges.

Challenge I: How to achieve structural compatibility for variable-length, entity-aligned actions? With dynamic entities, the policy must map a variable-size set of entity observations to a variable-length set of continuous actions aligned to the active entity set. Existing set encoders such as Deep Sets (Zaheer et al., 2017), Set Transformer (Lee et al., 2019), and attention-based MARL (Iqbal & Sha, 2019; Locatello et al., 2020) achieve permutation invariance or equivariance for variable-size inputs, but couple entity pathways strongly through a shared pooled latent or through attention over the full set, so a localized entity set change perturbs every output. Variable action availability methods (Ie et al., 2019; Chandak et al., 2020) allow the candidate pool to change, but the output dimension remains fixed. In contrast, our setting requires the output length itself to vary with the active entity set. Moreover, the control semantics are assignment-sensitive, meaning that swapping actions between entities changes outcomes. The policy must be permutation-equivariant (PE), ensuring that permuting the input entities produces the same permutation in the output actions, thereby preserving entity-action correspondence.

Challenge II: How to maintain stable learning under structural non-stationarity? When the entity set changes, the structure of the control problem changes, and the agent must adapt efficiently rather than retrain from scratch. Meta-RL (Finn et al., 2017), RL^2 (Grigsby et al., 2024), and continual RL (Rolnick et al., 2019) address distribution shifts but assume a fixed policy interface and do not handle cases where the action space itself expands or contracts. Standard actor-critic methods rely on a fixed interface between the policy and the critic (Yu et al., 2022; Zhang et al., 2024). When entities join or leave, the critic must still provide valid value estimates and gradients for the new configuration. Moreover, changes in the entity set are typically localized—affecting only a subset of entities—so a critic that outputs only a global value signal cannot identify which specific entity–action pairs need to be adjusted. Without entity-level credit assignment, learning signals are diluted across all entities, slowing adaptation and potentially destabilizing updates for unaffected entities.

To tackle these challenges, we propose **A**daptive Mu**L**ti-Agent Reinforcement **L**earning with Variable **O**bservation and ac**T**ion **S**paces (**ALLOTS**), an adaptive actor-critic framework for multi-agent control under structural non-stationarity, which adapts to variable observation and action spaces through set-based architectures. First, we design a PE set-to-set actor that processes variable-size entity observations via shared encoding and set pooling. This enables a variable-length set of continuous actions aligned to each entity, thereby achieving structural compatibility. Second, we introduce a set-structured critic that encodes entity-level state-action pairs as joint tokens and bootstraps from the realized post-churn entity set, enabling temporal difference (TD)-consistent value estimation across entity set changes. Third, we derive an entity-level advantage signal that decomposes the global return into per-entity contributions, providing fine-grained credit assignment that stabilizes

actor updates and accelerates adaptation to changing entity sets. In summary, our main contributions are as follows.

- We formalize MARL with variable observation and action spaces that evolve with dynamically changing entities, and propose ALLOTS, an adaptive MARL framework addressing this setting.
- We design a PE set-to-set actor that maps variable-size observations to entity-aligned variable-length continuous actions via shared set encoding, achieving structural compatibility.
- We introduce a set-structured critic with TD-consistent bootstrapping on the actual next-step active set, together with an entity-level advantage signal for fine-grained credit assignment, enabling stable and efficient adaptation to changing entity sets.
- We instantiate ALLOTS in three environments with an intermittent churn protocol, and benchmark against adapted MARL baselines across churn intensities.

2 Related Work

2.1 Variable-dimensional MARL

Handling variability in observation and action spaces is a fundamental challenge in multi-agent systems with dynamic entities. On the observation side, permutation-invariant (PI) and PE architectures such as Deep Sets (Zaheer et al., 2017), Set Transformer (Lee et al., 2019), and attention-based critics such as MAAC (Iqbal & Sha, 2019) process varying numbers of agents, and object-centric models (Locatello et al., 2020) learn representations over entity sets. Graph neural networks such as GCNs (Kipf & Welling, 2017) and Transformers over entity tokens (Tang & Ha, 2021) enable flexible encoding of graph-structured or set-based observations. On the action side, variable action availability has been studied through state-dependent candidate sets. SAS-MDPs (Boutilier et al., 2018) and lifelong RL (Chandak et al., 2020) address scenarios where feasible actions vary with state or change over time. Slate-based methods such as SlateQ (Ie et al., 2019) and AGILE (Jain et al., 2021) select fixed-size subsets from changing candidate pools. These methods vary either the observation size or the feasible action set, but not the output dimension, leaving the challenge of generating entity-aligned action sets unaddressed.

2.2 Adaptation under Non-Stationarity

Meta-RL methods train agents to adapt quickly to new tasks with limited experience, such as MAML (Finn et al., 2017), recurrent adaptation via RL^2 (Grigsby et al., 2024), off-policy context-based methods (Rakelly et al., 2019), and Bayes-adaptive methods (Zintgraf et al., 2021). Continual RL addresses distribution shifts and catastrophic forgetting through replay-based methods (Rolnick et al., 2019) and policy consolidation (Kaplanis et al., 2019). In MARL, dynamic team composition has been studied where agents join or leave during execution: COPA (Liu et al., 2021) learns compositional policies across team configurations, online ad hoc teamwork methods (Gu et al., 2022) adapt to unknown teammates, and mean-field methods (Yang et al., 2018) enable scalability to varying agent counts. However, these methods assume a fixed policy interface and do not address settings where the controlled entity set itself changes, requiring both the policy output and value estimation to adapt structurally.

3 Preliminaries

3.1 Problem Formulation

Standard MARL formulations, e.g., Dec-POMDP (Peralez et al., 2025) assume a fixed agent set throughout the episode, with fixed observation and action spaces for each agent. In our setting, each agent (e.g., a manager) controls a *dynamic* set of entities (e.g., devices) that may join or leave over time, inducing *variable* observation and action spaces. We formalize this extended setting below, where $\mathcal{D}$ denotes the entity set controlled by a single agent.

As a concrete instantiation, we consider FO-based energy management system, where devices generate FOs and the manager issues continuous decisions[1]. Since all agents face the same structural challenge, we present the formulation from a single agent's perspective without loss of generality. Time is discretized into decision steps $t = 0, 1, \ldots, T-1$. Let $\mathcal{D}_t \subseteq \mathcal{D}$ denote the active entity set at time t. Entity set change corresponds to $\mathcal{D}_{t+1} \neq \mathcal{D}_t$.

Definition 3.1 (Agent Observation)**.** For each active entity $d \in \mathcal{D}_t$, let x_t^d denote its local state at time t (e.g., SoC/thermal state, flexibility status, availability window, static attributes). The variable-size entity-state set is $X_t = \{x_t^d\}_{d \in \mathcal{D}_t}$, which varies with entity set changes. At time t, the agent also observes a fixed-dimensional system-level context $g_t \in \mathbb{R}^m$ (e.g., prices and grid indicators). We denote the full observation by $s_t = (g_t, X_t)$, so the observation space is variable-size due to entity set changes.

For example, in FO-based energy management system, when an EV leaves the portfolio, its state x_t^d is removed from X_t, reducing the observation size; when a new heat pump joins, its state is added, expanding the observation.

Definition 3.2 (Aligned Action Set)**.** For each active entity $d \in \mathcal{D}_t$, the agent issues a continuous action $a_t^d \in \mathcal{A}_d \subseteq \mathbb{R}^{H_a}$, where $\mathcal{A}_d$ denotes the feasible action set for entity d (e.g., power bounds determined by device capacity and current state). The aligned joint action at time t is the set of entity-action pairs $A_t = \{(d, a_t^d)\}_{d \in \mathcal{D}_t}$.

This representation explicitly encodes entity-action alignment, e.g., for devices $d_1, d_2 \in \mathcal{D}_t$, the aligned action set $\{(d_1, a_t^{d_1}), (d_2, a_t^{d_2})\}$ differs from $\{(d_1, a_t^{d_2}), (d_2, a_t^{d_1})\}$ in outcome, even though both contain the same action vectors $\{a_t^{d_1}, a_t^{d_2}\}$.

With variable-size observations and entity-aligned actions, the policy must map inputs to outputs while preserving the correspondence between entities and their assigned actions.

Definition 3.3 (Set-to-Set Policy)**.** An entity-aligned policy maps the agent observation to one action per active entity: $\pi : s_t \to \{a_t^d\}_{d \in \mathcal{D}_t}$, equivalently producing the aligned set A_t (Def. 3.2). We require π to be PE: permuting the presentation order of entities in X_t results in the same permutation of the per-entity outputs, thereby preserving entity-action alignment.

To train such a policy effectively, the learning algorithm must provide stable signals even as the controlled set evolves. We therefore define the environment dynamics and value function as follows.

Definition 3.4 (Transition with Dynamic Entities)**.** The environment evolves according to a transition kernel $p(g_{t+1}, X_{t+1}, \mathcal{D}_{t+1} \mid g_t, X_t, \mathcal{D}_t, A_t)$, where the next active set $\mathcal{D}_{t+1}$ is part of the transition and captures entity set changes. After applying A_t, the environment emits a scalar reward $r_t = r(g_t, X_t, A_t)$.

To evaluate such a policy, the value function must account for the entity-action alignment, as different assignments lead to different outcomes.

Definition 3.5 (Assignment-Sensitive Value Function)**.** The assignment-sensitive action-value function evaluates expected returns as a function of aligned pairs: $Q(g_t, X_t, A_t) = \mathbb{E}[\sum_{k=t}^{T-1} \gamma^{k-t} r_k \mid g_t, X_t, A_t]$, where $\gamma \in (0, 1]$ is the discount factor and the expectation is over environment stochasticity and future actions induced by π.

3.2 Learning Objective

We learn a parameterized entity-aligned policy π_θ (Def. 3.3) to maximize the expected discounted return:

$$\max_\theta \; J(\theta) \triangleq \mathbb{E}\left[\sum_{t=0}^{T-1} \gamma^t r(g_t, X_t, A_t)\right], \tag{1}$$

where A_t is induced by π_θ. Solving this problem requires a policy architecture that accommodates variable-size inputs and outputs while preserving entity-action alignment, together with a critic that provides stable value estimates and learning signals as the controlled set evolves.

[1] Appendix A.1 provides a detailed example illustrating how dynamic entities arise in this setting.

4 ALLOTS

We propose ALLOTS, an entity-aligned set adaptive learning framework for variable observation and action spaces that addresses both challenges through complementary designs. For structural compatibility, we design a PE set-to-set actor that handles variable-size inputs and outputs while preserving entity-action alignment. For stable adaptation, we introduce a set-structured critic with TD-consistent bootstrapping and entity-level credit assignment, enabling efficient learning across entity set changes without retraining. Figure 2 reports the overview of ALLOTS.

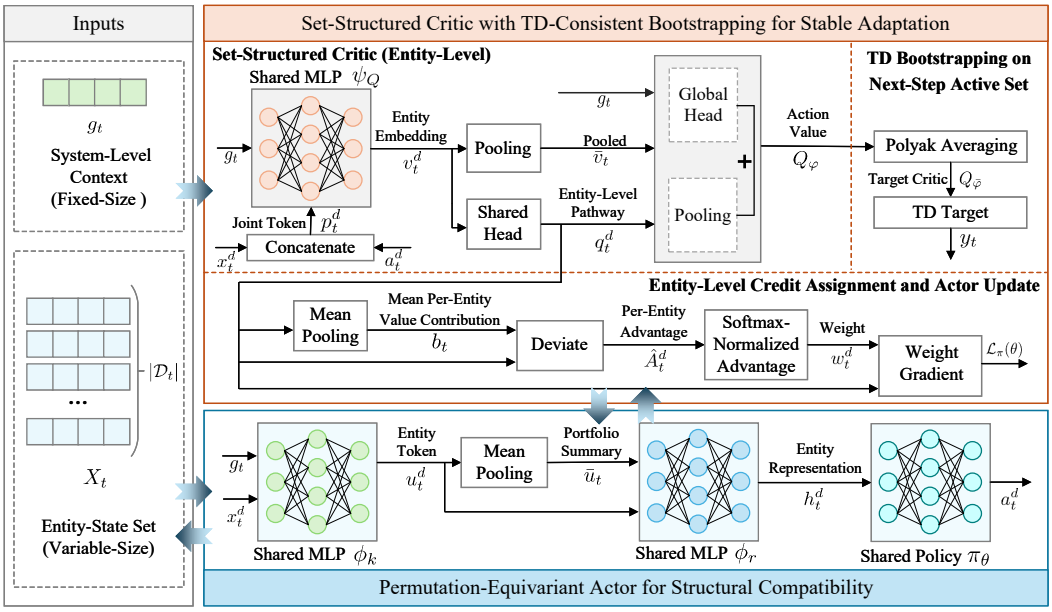

Figure 2: Overview of the ALLOTS framework.

4.1 Permutation-Equivariant Actor for Structural Compatibility

To achieve structural compatibility, we design an actor that maps variable-size observations to variable-length actions while preserving entity-action alignment. Under entity set changes, the required output is the aligned set A_t (Def. 3.2), whose cardinality changes with $|\mathcal{D}_t|$. A fixed pool of action heads cannot represent such an interface, and strong cross-entity coupling would let a localized entity set change propagate to all outputs. Moreover, entity control (e.g., FO control) semantics are assignment-sensitive (Def. 3.2), so any reordering of the input entity presentation must only reorder the per-entity outputs accordingly (Def. 3.3), otherwise the controller may produce mismatched entity-action assignments.

Given the observation $s_t = (g_t, X_t)$ (Def. 3.1), we represent the active portfolio using one token per entity and apply shared transformations across entities. For each $d \in \mathcal{D}_t$, we construct an entity token by combining its local state with the shared context:

$$u_t^d = \phi_k\big([x_t^d \,\|\, g_t]\big) \in \mathbb{R}^H, \tag{2}$$

where $[\cdot||\cdot]$ denotes vector concatenation, ϕ_k is a shared multilayer perceptron (MLP), and H is the token dimension. If each entity were processed independently, the actor could not adjust actions based on the current portfolio composition (e.g., how many flexible devices are active, or how scarce flexibility is), which is essential when constraints and costs depend on aggregate behavior. We therefore compute a PI portfolio summary by mean pooling:

$$\bar{u}_t = \frac{1}{|\mathcal{D}_t|} \sum_{d \in \mathcal{D}_t} u_t^d \in \mathbb{R}^H. \tag{3}$$

Mean pooling preserves permutation symmetry and keeps representation magnitudes comparable across different set sizes. We then refine each entity $d \in \mathcal{D}_t$ representation by injecting this portfolio

summary:

$$h_t^d = \phi_r\big([u_t^d \,\|\, \bar{u}_t]\big) \in \mathbb{R}^{H'}, \tag{4}$$

where ϕ_r is another shared MLP and H' is the refined token dimension. Finally, a shared policy head outputs one continuous FO vector per active entity $d \in \mathcal{D}_t$:

$$a_t^d = \pi_\theta(h_t^d), \tag{5}$$

where $a_t^d \in \mathcal{A}_d$ and $\mathcal{A}_d$ denotes the feasible action set for entity d. Since ϕ_k and ϕ_r are shared and the only cross-entity operation is the PI pooling in Eq. (3), permuting the entity order permutes $\{h_t^d\}$ and $\{a_t^d\}$ identically, which satisfies PE in Def. 3.3; by Assumption B.1, the policy output is therefore independent of the arbitrary ordering of entities in the input (Proposition B.3). Applying the same policy head to each entity token produces exactly $|\mathcal{D}_t|$ action vectors, so the aligned joint action A_t (Def. 3.2) expands and contracts with entity set changes while preserving entity-action alignment.

4.2 Set-Structured Critic with TD-Consistent Bootstrapping for Stable Adaptation

Our critic design has three components: (1) a set-structured architecture that preserves assignment semantics, (2) TD-consistent bootstrapping that maintains valid Bellman targets across entity set changes, and (3) entity-level advantage for fine-grained credit assignment. Together, these enable stable and efficient adaptation to changing entity sets.

Entity set changes induce variable observation and action spaces: the controlled set changes ($\mathcal{D}_{t+1} \neq \mathcal{D}_t$). Thus, both the critic input and the bootstrapping target must remain semantically well-defined. With a fixed-slot critic implemented via padding or placeholders, the meaning of "slot i" can drift across replay samples (e.g., slot 1 may correspond to device d_1 at time t but to device d_2 (or to padding) at time $t{+}1$). Thus, the bootstrapping term evaluates a mixture of real-device and placeholder semantics, which injects noise/bias into TD targets and makes updates unstable or sample-inefficient. To support adaptation to changing entity sets (rather than retraining from scratch), the critic should (i) be assignment-sensitive with respect to the aligned action set A_t (Def. 3.2), and (ii) perform TD bootstrapping on the realized next active set produced by the transition (Def. 3.4), so that the Bellman target evaluates the value of the next-step control problem.

Set-structured critic with an explicit entity-level pathway. To preserve state-action pairing semantics, we form a joint token for each active entity by concatenating its state and assigned action:

$$p_t^d = [x_t^d \,\|\, a_t^d], \tag{6}$$

where p_t^d is the state-action joint token for entity d and $P_t = \{p_t^d\}_{d \in \mathcal{D}_t}$ is the set of joint tokens over the active set $\mathcal{D}_t$. In our setting, outcomes depend on which entity receives which action (Def. 3.2); therefore, two action sets that contain the same vectors but are assigned to different entities should not be treated as equivalent. Using joint tokens p_t^d makes the assignment explicit, so swapping actions between entities changes the critic input and can lead to a different value estimate. We embed each joint token (conditioned on g_t) with a shared critic encoder to obtain an entity-level embedding v_t^d:

$$v_t^d = \psi_Q\big([p_t^d \,\|\, g_t]\big) \in \mathbb{R}^{H_Q}, \tag{7}$$

where ψ_Q is a shared MLP and H_Q is the critic embedding dimension. The pooled summary is computed as:

$$\bar{v}_t = \frac{1}{|\mathcal{D}_t|} \sum_{d \in \mathcal{D}_t} v_t^d \in \mathbb{R}^{H_Q}. \tag{8}$$

This pooling aggregates information from all active entities into a single portfolio-level representation $\bar{v}_t$, enabling the critic to capture global portfolio characteristics. This operation is permutation invariant. Thus, the critic output does not depend on the arbitrary ordering of entities in the input. Moreover, averaging rather than summing keeps the representation magnitude stable as entities join or leave, which enables preventing the critic scale from drifting with portfolio size changes.

Entity set changes typically affect only part of the structure (e.g., a small fraction of devices join/leave). It is undesirable for policy updates to collapse into a purely averaged "global" signal.

We therefore expose an entity-level scalar pathway to make learning signals entity-discriminative:

$$q_t^d = \rho_{\mathrm{dev}}(v_t^d) \in \mathbb{R}, \tag{9}$$

where ρ_{dev} is a shared scalar head that maps each entity embedding to a scalar value contribution q_t^d. This entity-level pathway enables the critic to attribute value to individual entities rather than only producing a monolithic global estimate, which is essential for fine-grained credit assignment when only a subset of entities are affected by entity set changes. Combining the per-entity contributions with a global term, the critic outputs the centralized action-value:

$$Q_\varphi(g_t, X_t, A_t) = \frac{1}{|\mathcal{D}_t|} \sum_{d \in \mathcal{D}_t} q_t^d \; + \; \rho_{\mathrm{glob}}\big([\bar{v}_t \,\|\, g_t]\big), \tag{10}$$

where ρ_{glob} is a global head that captures residual portfolio-level interactions beyond the per-entity contributions. The first term aggregates individual entity value contributions, providing a decomposable signal for entity-level credit assignment. The second term models collective effects that cannot be attributed to any single entity (e.g., grid-level constraints or synergies among devices in energy management). This decomposition allows the critic to provide both a global Q value for TD learning and entity-level signals for targeted actor updates. Since the global head can absorb any term that does not depend on an individual entity's action, q_t^d is determined only up to a per-state offset; the actor update below depends solely on the relative ordering of q_t^d across the active set, which this offset leaves unchanged. All critic mappings are shared across entities and use PI pooling, so Q_φ is PI over the entity set, while remaining assignment-sensitive because p_t^d changes when entity-action assignments change. Under Assumption B.2, the joint tokens ensure that the critic can distinguish different entity-action assignments (Proposition B.4).

TD-consistent bootstrapping on the realized next active set. For each transition (Def. 3.4), the environment realizes the next active set $\mathcal{D}_{t+1}$. We therefore construct the next-step aligned action set on $\mathcal{D}_{t+1}$ by applying the actor to (g_{t+1}, X_{t+1}):

$$A_{t+1} = \{(d, a_{t+1}^d)\}_{d \in \mathcal{D}_{t+1}}, \tag{11}$$

where $a_{t+1}^d = \pi_\theta(h_{t+1}^d)$ and A_{t+1} is defined only on the realized next active set $\mathcal{D}_{t+1}$. For example, for devices that depart ($d \in |\mathcal{D}_{t+1}| < |\mathcal{D}_t|$), no actions are created; for devices that join ($d \in |\mathcal{D}_{t+1}| > |\mathcal{D}_t|$), the actor generates actions based solely on their current state x_{t+1}^d and the global context g_{t+1}, requiring no historical information. Because all actor components (ϕ_k, ϕ_r, and the policy head) are shared across entities, newly joined entities are processed identically to existing ones, ensuring semantic consistency.

We then define the TD target using a slowly-updated target critic $Q_{\bar{\varphi}}$:

$$y_t = r_t + \gamma(1 - \mathrm{done}_t)\, Q_{\bar{\varphi}}(g_{t+1}, X_{t+1}, A_{t+1}), \tag{12}$$

where $\gamma \in (0, 1)$ is the discount factor and $\mathrm{done}_t \in \{0, 1\}$ indicates episode termination. $Q_{\bar{\varphi}}$ is a target critic updated by Polyak averaging, i.e., $\bar{\varphi} \leftarrow \tau\varphi + (1-\tau)\bar{\varphi}$ with $\tau \in (0, 1)$ denoting the averaging rate. Because A_{t+1} is defined on the realized $\mathcal{D}_{t+1}$, the bootstrapping term always evaluates the value of the actual next-step control problem. This allows the critic to be continuously updated without relearning the entire mapping, providing stable gradients for the actor. The complete critic update procedure is given in Algorithm 1 (Appendix C.2).

Entity-level advantage and actor update. To enable entity-discriminative actor updates and encourage localized adaptation, we form an entity-level advantage from the entity-level pathway. We first compute a set-level baseline as the mean entity-level value contribution:

$$b_t = \frac{1}{|\mathcal{D}_t|} \sum_{d \in \mathcal{D}_t} q_t^d. \tag{13}$$

The entity-level advantage then measures how much each entity's value contribution deviates from this baseline:

$$\hat{A}_t^d = q_t^d - b_t. \tag{14}$$

The weighting is monotone in $\hat{A}_t^d$, so the largest weights are assigned to entities whose value score lies furthest above that of their peers. These are the entities on which the actor update concentrates,

which typically includes newly joined entities and a small subset of high-impact entities under the current context. For example, if a heat pump contributes more to cost savings than a dishwasher, the weighting assigns higher credit to the heat pump, directing more policy updates toward improving its control. We then update the actor with a deterministic policy objective that emphasizes localized adaptation. Each entity's gradient contribution is scaled by a nonnegative gate w_t^d derived from the entity-level advantage $\hat{A}_t^d$:

$$\mathcal{L}_\pi(\theta) = -\mathbb{E}\left[\sum_{d\in\mathcal{D}_t} w_t^d \, q_t^d\right], \tag{15}$$

where q_t^d is the critic's entity-level value contribution in Eq. (9), evaluated at the current policy-induced actions a_t^d through the joint tokens p_t^d. This objective can be viewed as an entity-wise, credit-weighted variant of deterministic policy gradient, which aligns actor updates with improving the centralized value in Eq. (10) while preventing unnecessary changes on unaffected entities. We weight each entity's contribution by a softmax-normalized advantage:

$$w_t^d = \frac{\exp(\hat{A}_t^d/\tau_w)}{\sum_{d'\in\mathcal{D}_t}\exp(\hat{A}_t^{d'}/\tau_w)}, \tag{16}$$

where $\tau_w > 0$ is a temperature hyperparameter. Since $\sum_{d\in\mathcal{D}_t} w_t^d = 1$, the total weight assigned to the active set is invariant to $|\mathcal{D}_t|$, so the actor update magnitude does not scale with the number of controlled entities; within that budget, larger advantages receive proportionally more weight. The softmax is invariant to constant shifts of the entity-level scores, so b_t enters the weights only through the differences $\hat{A}_t^d$. As $\tau_w \to \infty$ the weights become uniform and Eq. (15) reduces to the unweighted average in Eq. (27). We treat w_t^d as a constant with respect to θ, so gradients do not flow through the weights. The complete actor update procedure is given in Algorithm 2 (Appendix C.2).

5 Experiments

5.1 Experimental Setup

Environments. We evaluate ALLOTS across three domains: FOgym (Xu et al., 2026), a bottom-up FO-based energy management simulator; a modified Multi-Agent Particle Environment (MPE) implemented in PettingZoo (Terry et al., 2021) for cooperative 2D coverage; and MultiDrone, a cooperative 3D coverage environment built on the PyFlyt quadrotor simulator (Tai et al., 2023). Across all domains, we introduce intra-episode entity churn at three intensity levels to evaluate adaptation to variable observation and action spaces. Detailed environment configurations and experimental protocols are provided in Appendix C.1.

Baselines. We compare against seven representative MARL methods: Multi-agent Deep Deterministic Policy Gradient (MADDPG) (Lowe et al., 2017) and Multi-agent Twin Delayed Deep Deterministic Policy Gradient (MATD3) (Ackermann et al., 2019), Shapley Q-value Deep Deterministic Policy Gradient (SQDDPG) (Wang et al., 2020), Multi-agent Proximal Policy Optimization (MAPPO) (Yu et al., 2022), Multi-agent Independent Proximal Policy Optimization (MAIPPO) (Chou et al., 2025), Action Graph for Interdependence Learning (AGILE) (Jain et al., 2021), and Multi-Actor-Attention-Critic (MAAC) (Iqbal & Sha, 2019). These baselines cover diverse design choices in critic architecture and credit assignment. Since some of them assume fixed observation/action dimensions, we augment each with a compatibility layer that pads variable-size inputs and extracts valid actions via masking, without introducing ALLOTS's core mechanisms (Details in Appendix C.1.1).

5.2 Main Results

Comparison with baselines. Table 1 compares ALLOTS against the baselines on the two FOgym testbeds under three churn intensities. On FOgym-4M, ALLOTS achieves average rewards of 60.56, 60.30, and 61.69 under Low, Mid, and High Churn, outperforming the best-performing baseline by 36.5%, 28.1%, and 48.3%, respectively. On FOgym-10M, ALLOTS attains 55.03, 56.87, and 58.58, improving over the best baseline by 16.2%, 57.5%, and 94.7%. The widening gap indicates that

Table 1: Average reward of ALLOTS and baselines on FOgym environments under three churn intensities, where each result is averaged over 10 test episodes and 3 random seeds (higher is better).

Method	FOgym-4M			FOgym-10M		
	Low Churn	Mid Churn	High Churn	Low Churn	Mid Churn	High Churn
MADDPG	14.49 ± 5.27	11.43 ± 5.53	12.94 ± 4.97	13.84 ± 4.20	12.69 ± 4.18	12.68 ± 4.31
MATD3	9.20 ± 4.70	8.15 ± 4.16	8.88 ± 4.59	9.48 ± 4.03	8.68 ± 3.97	9.99 ± 4.31
MAPPO	-182.84 ± 4.39	-184.87 ± 5.02	-190.13 ± 4.88	-196.46 ± 3.48	-197.75 ± 7.96	-197.80 ± 2.87
MAIPPO	-170.54 ± 2.89	-180.56 ± 4.83	-183.18 ± 4.49	-178.76 ± 2.72	-190.15 ± 4.53	-191.68 ± 2.25
AGILE	44.37 ± 3.44	34.22 ± 3.95	25.02 ± 3.46	47.37 ± 2.47	33.11 ± 4.41	21.09 ± 5.25
MAAC	43.98 ± 3.12	47.08 ± 4.01	41.62 ± 3.67	36.28 ± 2.97	36.11 ± 3.88	30.09 ± 2.94
SQDDPG	33.41 ± 4.31	26.52 ± 4.22	27.54 ± 4.21	27.38 ± 4.68	24.66 ± 4.57	24.73 ± 2.34
ALLOTS	**60.56 ± 2.46**	**60.30 ± 3.18**	**61.69 ± 3.45**	**55.03 ± 7.73**	**56.87 ± 4.35**	**58.58 ± 4.14**

ALLOTS's set-structured design is particularly effective when frequent entity-set changes stress the adaptation mechanisms. Training curves are provided in Appendix C.3.

Among the baselines, MAAC and AGILE generally outperform standard actor-critic methods (MADDPG, MATD3, SQDDPG), likely because their architectures provide greater flexibility in handling variable inputs. However, their fixed output interfaces still limit adaptation to changing entity sets. On-policy methods (MAPPO, MAIPPO) perform poorly across all settings, likely because samples become stale quickly under entity-set changes and they lack replay buffers that retain useful experience from previous configurations. The same overall trend extends beyond energy management: ALLOTS achieves the highest average reward at every churn intensity in both Modified MPE and MultiDrone, showing that its advantages persist across different entity semantics and physical dynamics. Complete results are reported in Appendix C.3.

Ablation study. We ablate paired state-action tokens (No-PS), TD-consistent bootstrapping over the realized next active set (No-TD), and softmax-weighted entity-level credit assignment (No-EC). Detailed variant definitions are provided in Appendix C.4; the results are reported in Table 2.

Table 2: Ablation study of ALLOTS on FOgym environments.

Method	FOgym-4M			FOgym-10M		
	Low Churn	Mid Churn	High Churn	Low Churn	Mid Churn	High Churn
ALLOTS No-PS	59.38 ± 3.37	56.83 ± 3.41	52.65 ± 4.89	56.16 ± 4.85	50.23 ± 5.33	53.46 ± 6.09
ALLOTS No-TD	60.63 ± 2.96	38.98 ± 4.27	46.07 ± 5.84	47.53 ± 5.09	55.17 ± 4.21	55.16 ± 7.86
ALLOTS No-EC	57.40 ± 1.94	53.22 ± 5.15	57.74 ± 3.94	54.52 ± 3.87	56.65 ± 6.41	50.95 ± 6.79
ALLOTS	**60.98 ± 2.89**	**59.35 ± 3.12**	**58.84 ± 4.53**	**57.41 ± 4.43**	**58.80 ± 5.19**	**60.96 ± 5.87**

All three ablations reduce performance across every FOgym setting, confirming that each component contributes to ALLOTS. No-TD causes the largest single degradation (34.32%) on FOgym-4M under Mid Churn, while No-PS losses increase from 2.18%–2.62% at Low Churn to 10.52%–12.30% at High Churn, and No-EC reaches a 16.42% loss on FOgym-10M under High Churn. The same ranking (No-PS $>$ No-TD $>$ No-EC) holds in Modified MPE and MultiDrone, where Pair-Set removal increases the average penalty by 14.1% and 21.8%, respectively, indicating greater component importance under more complex dynamics. Detailed analyses are provided in Appendix C.4 and Tables 7 and 8, respectively.

A hyperparameter analysis of τ_w and H yields testbed-specific settings (τ_w=1.0, H=128 for FOgym-4M; τ_w=2.0, H=256 for FOgym-10M), reflecting the need for smoother credit assignment and greater capacity as the portfolio grows (Appendix C.5).

6 Conclusion and Future Work

We studied MARL under variable observation and action spaces caused by dynamic entities, and proposed ALLOTS, an adaptive actor-critic framework combining a permutation-equivariant set-to-set actor with a set-structured critic that pairs state-action tokens, bootstraps TD targets on the realized next active set, and assigns entity-level credit via softmax-weighted advantages. Experiments on FOgym, Modified MPE, and MultiDrone under three churn intensities show that ALLOTS consistently outperforms MARL baselines, and ablations on all three environments confirm the contribution of each component. Future work includes making ALLOTS robust to state perturbations.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used ChatGPT and Claude to improve the manuscript's grammar, clarity, readability, and language fluency. The figures, references, paper structure, core content, and methodological design were created by the authors. All AI-assisted text edits were reviewed and adapted by the authors, who take responsibility for the final content of this work.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

This work uses simulated multi-agent environments and does not involve human or animal subjects, user studies, personal data, or sensitive information. We are not aware of any specific ethical concerns arising from the methods, experiments, or applications presented in this paper.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

To support reproducibility, we provide an anonymous GitHub repository linked in the paper containing the code. The main paper describes the ALLOTS method and experimental protocol, while Appendix C documents the environments, churn configurations, evaluation protocol, and training budgets. Appendix C.2 provides the detailed training procedures, and Appendix C.5.1 lists the hyperparameters used for ALLOTS and the baseline methods. The environments and external software used in our experiments are identified with citations and source links in the appendix. All experiments were conducted on NVIDIA A40 GPU.

REFERENCES

Johannes Ackermann, Volker Gabler, Takayuki Osa, and Masashi Sugiyama. Reducing overestimation bias in multi-agent domains using double centralized critics. *Deep RL Workshop at Advances in Neural Information Processing Systems*, 2019.

Craig Boutilier, Alon Cohen, Avinatan Hassidim, Yishay Mansour, Ofer Meshi, Martin Mladenov, and Dale Schuurmans. Planning and learning with stochastic action sets. In *International Joint Conference on Artificial Intelligence*, pp. 4674–4682, 2018.

Yash Chandak, Georgios Theocharous, Chris Nota, and Philip Thomas. Lifelong learning with a changing action set. In *AAAI Conference on Artificial Intelligence*, volume 34, pp. 3373–3380, 2020.

Che Wei Chou, Wei Cheng Chiu, and Yu Teng Hsu. Multiagent reinforcement learning-based dispatching model for overhead hoist transfer in automated material handling system. *Computers & Industrial Engineering*, 204:111109, 2025.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pp. 1126–1135, 2017.

Jake Grigsby, Justin Sasek, Samyak Parajuli, Daniel Adebi, Amy Zhang, and Yuke Zhu. AMAGO-2: Breaking the multi-task barrier in meta-reinforcement learning with transformers. *Advances in Neural Information Processing Systems*, 37:87473–87508, 2024.

Pengjie Gu, Mengchen Zhao, Jianye Hao, and Bo An. Online ad hoc teamwork under partial observability. In *International conference on learning representations*, 2022.

Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Tushar Chandra, and Craig Boutilier. SLATEQ: A tractable decomposition for reinforcement learning with recommendation sets. In *International Joint Conference on Artificial Intelligence*, pp. 2592–2599, 2019.

Shariq Iqbal and Fei Sha. Actor-attention-critic for multi-agent reinforcement learning. In *International Conference on Machine Learning*, volume 97, pp. 2961–2970, 2019.

Ayush Jain, Norio Kosaka, Kyung Min Kim, and Joseph J Lim. Know your action set: Learning action relations for reinforcement learning. In *International Conference on Learning Representations*, 2021.

Christos Kaplanis, Murray Shanahan, and Claudia Clopath. Policy consolidation for continual reinforcement learning. In *International Conference on Machine Learning*, volume 97, pp. 3242–3251, 2019.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, volume 97, pp. 3744–3753, 2019.

Bo Liu, Qiang Liu, Peter Stone, Animesh Garg, Yuke Zhu, and Anima Anandkumar. Coach-player multi-agent reinforcement learning for dynamic team composition. In *International Conference on Machine Learning*, volume 139, pp. 6860–6870, 2021.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020.

Ryan Lowe, Yi I Wu, Aviv Tamar, Jean Harb, OpenAI Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. *Advances in Neural Information Processing Systems*, 30, 2017.

Torben Bach Pedersen, Laurynas Šikšnys, and Bijay Neupane. Modeling and managing energy flexibility using flexoffers. In *IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pp. 1–7, 2018.

Johan Peralez, Aurélien Delage, Jacopo Castellini, Rafael F. Cunha, and Jilles Steeve Dibangoye. Optimally solving simultaneous-move Dec-POMDPs: The sequential central planning approach. In *AAAI Conference on Artificial Intelligence*, volume 39, pp. 23276–23285, 2025.

Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, volume 97, pp. 5331–5340, 2019.

David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Laurynas Šikšnys, Torben Bach Pedersen, Muhammad Aftab, and Bijay Neupane. Flexibility modeling, management, and trading in bottom-up cellular energy systems. In *ACM International Conference on Future Energy Systems*, pp. 170–180, 2019.

Jun Jet Tai, Jim Wong, Mauro Innocente, Nadjim Horri, James Brusey, and Swee King Phang. PyFlyt–UAV simulation environments for reinforcement learning research. *arXiv preprint arXiv:2304.01305*, 2023.

Yujin Tang and David Ha. The sensory neuron as a transformer: Permutation-invariant neural networks for reinforcement learning. *Advances in Neural Information Processing Systems*, 34: 22574–22587, 2021.

J. K. Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S. Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, Niall Williams, Yashas Lokesh, and Praveen Ravi. PettingZoo: Gym for multi-agent reinforcement learning. 34: 15032–15043, 2021.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley Q-Value: A local reward approach to solve global reward games. In *AAAI conference on artificial intelligence*, volume 34, pp. 7285–7292, 2020.

Jiachen Xu, Torben Bach Pedersen, Zhongming Yao, Tianyi Li, and Yushuai Li. FOgym: A bottom-up home energy management framework based on flexoffers and multi-agent reinforcement learning. In *ACM International Conference on Future and Sustainable Energy Systems*, pp. 303–320, 2026.

Yaodong Yang, Rui Luo, Minne Li, Ming Zhou, Weinan Zhang, and Jun Wang. Mean field multi-agent reinforcement learning. In *International Conference on Machine Learning*, volume 80, pp. 5571–5580, 2018.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in Neural Information Processing Systems*, 30, 2017.

Miaomiao Zhang, Wei Tong, Guangyu Zhu, Xin Xu, and Edmond Q. Wu. SQIX: QMIX algorithm activated by general softmax operator for cooperative multiagent reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 54(11):6550–6560, 2024.

Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: Variational Bayes-Adaptive deep RL via meta-learning. *Journal of Machine Learning Research*, 22(289):1–39, 2021.