BusEnv: A Multi-agent Reinforcement Learning Environment and Benchmark for Urban Public Transportation

Wesley da Silva e Silva
Federal University of Bahia
Salvador, Brazil
silvaw@ufba.br

Ricardo Araújo Rios
Federal University of Bahia
Salvador, Brazil
ricardoar@ufba.br

Rafael da Costa Fonseca
Federal University of Bahia
Salvador, Brazil
r.fonseca@ufba.br

Sabarikirishwaran Ponnambalam
Griffith University
Brisbane, Queensland, Australia
sabarikirishwaran.ponnambalam@griffithuni.edu.au

Léa Cassé
University of Waikato École Polytechnique
Hamilton & Palaiseau, New Zealand & France
casse.lea@gmail.com

Marcos Vinícius dos Santos Ferreira
Federal University of Bahia
Salvador, Brazil
marcosvsf@ufba.br

Albert Bifet
University of Waikato
Hamilton, New Zealand
albert.bifet@waikato.ac.nz

Tatiane Nogueira Rios
Federal University of Bahia
Salvador, Brazil
tatiane.nogueira@ufba.br

ABSTRACT

Reinforcement learning (RL) offers a powerful paradigm for managing complex, dynamic transportation systems where autonomous agents must adapt to uncertain and rapidly changing environments. We present BusEnv, a benchmark environment grounded in real-world data from the Salvador Urban Transportation Network, encompassing approximately 700,000 passengers, 2,000 vehicles, 400 lines, and 3,000 stops, collected between March 2024 and March 2025 at sub-minute resolution. BusEnv simulates realistic bus operations with stochastic passenger demand, route-specific travel times, and traffic-dependent variability, enabling controlled experimentation under partially observable, high-dimensional conditions. The reward function integrates multiple objectives, such as passenger service quality, operational efficiency, maintenance adherence, and sustainability, allowing the assessment of how different RL algorithms balance these competing factors. We evaluate nine baseline methods implemented in MARLlib, analyzing their convergence, robustness, and environmental impact when deployed under independent-learning conditions. Results show that PPO-based approaches achieve the highest stability and lowest energy waste, linking algorithmic robustness to sustainability performance. By combining data realism with reproducibility and extensibility, BusEnv establishes a foundation for systematic research on learning-based transport management and provides a scalable testbed for future studies on cooperative, sustainability-aware reinforcement learning.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/CYHA2042

KEYWORDS

Multi-agent environment; Reinforcement Learning; Public Transportation

ACM Reference Format:
Wesley da Silva e Silva, Ricardo Araújo Rios, Rafael da Costa Fonseca, Sabarikirishwaran Ponnambalam, Léa Cassé, Marcos Vinícius dos Santos Ferreira, Albert Bifet, and Tatiane Nogueira Rios. 2026. BusEnv: A Multi-agent Reinforcement Learning Environment and Benchmark for Urban Public Transportation. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/CYHA2042

1 INTRODUCTION

Most multiagent reinforcement learning (MARL) benchmarks center on synthetic or game-based domains, for example the StarCraft Multi-Agent Challenge (SMAC) [9, 16, 27] and Google Research Football (GRF) [18, 31, 32], which have become standard testbeds for evaluating cooperative and competitive strategies. Similarly, frameworks such as EPyMARL [26] and PyMARLzoo+ [25] expand benchmark diversity through domains from PettingZoo [33]. While useful for methodological progress and reproducibility, these domains offer limited realism for applied decision-making, with short horizons, handcrafted interactions, and dynamics far removed from large-scale real-world operations.

As a result, policies that perform well in these controlled settings may not generalize reliably to domains where agents must act under noisy, high-dimensional observations, dynamic constraints, and human-driven uncertainty. The abstraction that makes synthetic benchmarks tractable also limits their ecological validity, hindering the translation of algorithmic advances into applied systems such as transportation, energy, or public infrastructure. Addressing this limitation requires environments that preserve experimental

rigor while capturing real-world stochasticity, scale, and heterogeneity, conditions that more closely mirror those faced in complex operational networks.

To bridge this gap, we introduce BusEnv, a benchmark environment that combines the methodological consistency of existing MARL frameworks with the realism of an urban public transportation system. BusEnv models a large-scale bus network where multiple autonomous agents, representing vehicles, operate over realistic routes, passenger flows, and traffic patterns derived from the Salvador Urban Transportation System (SUNT). Each agent optimizes a local policy under partial observability, interacting with a shared, stochastic environment defined by route occupancy, passenger demand, and travel conditions. The reward function integrates four key objectives (service quality, operational efficiency, maintenance adherence, and traffic regularity), capturing the multi-objective nature of real-world transit control.

This formulation highlights key challenges of large-scale cooperative decision-making: partial observability, decentralized optimization, and the trade-off between local autonomy and global coordination. By modeling stochastic demand, resource constraints, and heterogeneous spatial interactions, BusEnv enables evaluation of MARL under realistic uncertainty. Unlike game-inspired benchmarks, it provides interpretable transportation metrics, such as headway regularity, occupancy balance, and energy consumption—linking algorithmic behavior to real urban outcomes.

The analysis of BusEnv was conducted through an extensive empirical study, evaluating nine state-of-the-art MARL algorithms implemented in MARLlib [14, 15], including IPPO [8, 30], IA2C [15, 23], ITRPO [15, 28], MAA2C [26], COMA [10], MAPPO [37], MATRPO [20], HAPPO [17], and HATRPO [17]. Results reveal distinct coordination regimes, where PPO-based centralized architectures consistently outperform trust-region and purely decentralized methods in both convergence and stability. Beyond classical reward-based analysis, we incorporate sustainability indicators, including total CO_2 emissions and energy efficiency, to quantify the environmental footprint of each algorithm, demonstrating that policy instability directly increases carbon intensity.

This work makes three main contributions:

(1) We present BusEnv, a data-driven MARL benchmark for urban public transportation that integrates real-world demand, network topology, and stochastic passenger behavior at city scale;
(2) We design a multi-objective reward function that operationalizes service quality, efficiency, maintenance, and traffic regularity, enabling interpretable evaluation of system-level trade-offs;
(3) We perform a comprehensive empirical study comparing nine MARL algorithms, highlighting the robustness of PPO-based centralized optimization and revealing the connection between coordination quality and environmental sustainability.

By coupling algorithmic evaluation with environmental impact analysis, BusEnv advances MARL research beyond synthetic or game-based scenarios, offering a reproducible and extensible testbed for studying sustainable, data-driven public transportation management.

2 RELATED WORK

Recent studies have increasingly sought to bridge MARL with real-world operational challenges. In the domain of infrastructure systems, IMP-MARL [19] addresses large-scale cooperative maintenance planning, emphasizing scalability and coordination across extensive agent populations. In the energy sector, Liu et al. [22] propose a reinforcement learning framework that combines Deep Deterministic Policy Gradient (DDPG) with action masking and MILP-based policy guidance to manage continuous control in vehicle-to-building (V2B) energy management. Using real-world data from an electric vehicle manufacturer, their approach achieves significant cost reductions while maintaining reliable energy supply, outperforming both heuristic and deep RL baselines. Similarly, EnEnv 1.0 [3] introduces a benchmark for smart grid operations, enabling the evaluation of multi-agent coordination and demand-response strategies under realistic operational constraints.

Beyond energy systems, MARL has been applied to large-scale distributed decision-making. Holder et al. [13] study sequential assignment in domains such as satellite constellations and power grids, learning assignment values from a greedy solver and integrating them into distributed optimal assignment with theoretical guarantees. In embodied intelligence, Assistax [12] provides a hardware-accelerated benchmark for human–robot coordination using JAX [4]. In collective robotics, CraftEnv [38], implemented in PyBullet [5], enables cooperative construction tasks and supports sim-to-real policy transfer, highlighting MARL's potential for swarm intelligence.

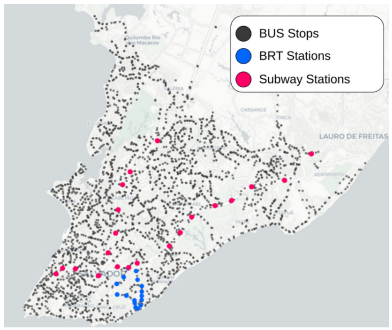
Despite these advances, a major gap remains: most MARL benchmarks operate on synthetic or simplified environments, or approximate real-world systems through stylized simulations. They rarely integrate large-scale, high-resolution data streams that capture the stochasticity, heterogeneity, and spatial–temporal complexity of urban mobility systems. BusEnv addresses this gap by providing a benchmark grounded in real passenger demand, traffic conditions, and vehicle operations from a major metropolitan network. By combining reproducibility with real-world data realism, it enables the systematic evaluation of MARL algorithms in a domain that is both practically relevant and methodologically rigorous for intelligent public transportation management.

Beyond advancing MARL research, BusEnv can be used to simulate real-world operational strategies such as headway-based holding [2, 7], skip-stop service [11], and schedule-based dispatch [1], which aim to improve reliability and reduce passenger waiting times.

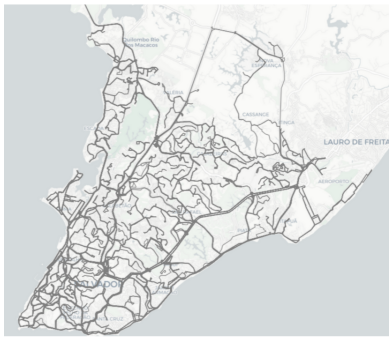
3 URBAN PUBLIC TRANSPORTATION BENCHMARK

3.1 Preliminaries

The dataset used in our environment was collected in Salvador, one of the largest cities in Brazil, from March 2024 to March 2025 at subminute resolution. It covers about 700,000 passengers and approximately 2,000 vehicles operating across nearly 400 lines, connecting almost 3,000 stops and stations. As illustrated in Figure 1(a), the dataset includes all available transportation modes: regular buses, subway, and BRT, a bus system operating in dedicated lanes.



(a)



(b)

Figure 1: Visualization of the integrated public transportation network of the metropolitan region of Salvador. In (a), bus stops (black), BRT stations (blue), and subway stations (pink) are shown. In (b), the road network covered by public transportation is displayed.

Unlike other public transportation studies, our dataset provides comprehensive details on city dynamics, including each passenger's origin and destination, the number of passengers traveling along routes connecting stops and stations (Figure 1(b)), boarding and alighting counts at each stop or station, and the time and speed of each vehicle between stops and stations.

Formally, the dataset is represented as a spatial-temporal graph $G = \{G_1, G_2, ..., G_T\}, \forall t = \{1, ..., T\}$, in which $G_t = (V, E)$ stands for an attributed and directed graph at time t, where $V = \{v_1, v_2, ..., v_N\}$ is the set of N vertices corresponding to the bus stops and stations, and E is the set of edges corresponding to feasible routes. A directed edge $(v_i, v_j) \in E$ connects vertices $v_i, v_j \in V$ if, and only if, there is a feasible route for the bus traffic from the corresponding station v_i to v_j in the network. G_t is a fixed graph structure since sets V and E do not change over time.

In our world representation, each passenger $p_m \in P$ is associated with a set of trips $\mathcal{T}_{mk} = (o_{mk}, d_{mk})$, where $o_{mk}, d_{mk} \in V$ denote origin and destination vertices on the graph in their k-th trip. Each vehicle $b_n \in B$ carries, at trip l, a passenger subset $p_n(l) \subseteq P$ and, for each traversed edge $(i, j) \in E$, records attributes including distance $d_{ij}(l)$, mean velocity $\bar{v}_{ij}(l)$, travel time $\tau_{ij}(l)$, and passenger count $|p_n(l)|$.

3.2 Environment formulation

Based on the previously described system, we formulate the public transportation network as a multi-agent reinforcement learning (MARL) environment, where buses are modeled as autonomous agents capable of independent decision-making within a shared urban network. Although the full formulation supports coordination and inter-agent dependencies, in this study we evaluate the environment under independent-learning conditions, where each agent optimizes its policy based on local information and shared rewards.

A multi-agent perspective remains essential in this domain, since in realistic operations each vehicle indirectly influences others through shared traffic conditions and passenger demand. BusEnv therefore provides the structure necessary to study both independent and coordinated control strategies, depending on the chosen training configuration. We formalize the bus network control problem as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP) [24], defined by the tuple

$$\langle S, Z, U, n, O, R, P, \gamma \rangle,$$

where n agents (buses) simultaneously choose actions at every time step t.

State space (S). The global state $s_t \in S$ encodes the configuration of the transportation system at time t. It is derived from the spatio-temporal graph $G_t = (V, E)$ introduced in the preliminaries, and includes: (i) vehicle positions and operational status, (ii) passenger demand across stops and stations, (iii) current occupancy levels, and (iv) traffic conditions such as travel times and velocities along edges.

Observation space (Z) and observation function (O). Each bus $a \in \{1, .., n\}$ receives a local observation $o_t^a \in Z$ obtained through the observation function $O : S \times \{1, .., n\} \rightarrow Z$. Since the underlying system is represented as a spatio-temporal graph $G_t = (V, E)$, observations are extracted from attributes associated with the vehicle b_a and its current edge traversal $(v_i, v_j) \in E$. The observation vector includes:

- $avg\_travel\_time_{AB}$: estimated travel time between reference stops A and B, computed from $\tau_{ij}(l)$ and $\bar{v}_{ij}(l)$ along traversed edges;
- $future\_demand_B$: predicted number of passengers expected to board at stop B, inferred from historical origin–destination pairs $\mathcal{T}_{mk}$ and boarding/alighting counts;
- $occupancy\_rate$: proportion of seats in use, obtained from the current passenger subset $p_a(l) \subseteq P$ carried by bus b_a;
- $uptime\_normalized$: normalized availability of the bus, derived from operational attributes such as fuel level and maintenance status.

These localized signals are consistent with the graph-based representation of the network: vertices provide access to demand information, edges capture travel dynamics, and vehicles record passenger and operational states. Together, they reflect the fact that agents perceive only partial information about the global system state.

Action space (U). Each agent a has a discrete action space $U^a = \{\texttt{WAIT}, \texttt{MOVE}, \texttt{SERVICE\_CENTER}\}$. The joint action space is $U = U^1 \times ... \times U^n$. Actions are defined as:

- `WAIT`: hold position before departing to the next stop, which may improve headway regularity and reduce bus clustering;
- `MOVE`: proceed to the next stop along the predefined route;
- `SERVICE_CENTER`: divert to maintenance facilities in case of low fuel or mechanical issues.

Transition function (P). Given a state s_t and joint action $u_t \in U$, the transition function $P(s_{t+1}|s_t, u_t)$ models the stochastic evolution of the system, accounting for traffic conditions, passenger flows, and vehicle interactions.

Reward Function Design (R). After each transition, the environment provides a shared global reward signal, integrating four objectives that reflect system-level performance:

- *Passenger service quality*: reduced waiting times, satisfied demand, and regular service;
- *Operational efficiency*: balanced occupancy and avoidance of overcrowding or underutilization;
- *Maintenance and fuel management*: penalties for ignoring operational constraints such as low fuel or technical issues;
- *Traffic flow and regularity*: penalties for bus bunching, idle times, and inefficient integration with road traffic.

Formally, for agent a at time t, we define:

$$r_t^a = \alpha_1 q_t^a + \alpha_2 e_t^a + \alpha_3 m_t^a + \alpha_4 c_t^a,$$

where q_t^a denotes service quality, e_t^a efficiency, m_t^a maintenance adherence, and c_t^a represents traffic regularity and network-level stability with α_k representing tunable weights.

Reinforcement Learning Training. Each agent a follows a policy $\pi^a(u_t^a|\tau_t^a, o_t^a)$ that maps its history $\tau_t^a \in (Z \times U)^{t-1}$ and current observation o_t^a into a distribution over actions u_t^a. The joint policy is $\pi = (\pi^1, ..., \pi^n)$. The cumulative discounted reward is:

$$R_t = \sum_{k=0}^{T-1} \gamma^k r_{t+k}, \quad \gamma \in [0, 1).$$

The goal of the agents is to learn the optimal joint policy

$$\pi^* = \arg\max_{\pi} \mathbb{E}[R_0 \mid \pi],$$

which maximizes the expected cumulative return across an episode.

In this formulation, reinforcement learning training aims to acquire policies that maximize long-term cumulative rewards over the course of an episode, rather than optimizing for immediate gains. In the present study, we adopt an independent-learning baseline, where each policy π^a is trained without parameter sharing or explicit communication among agents. This configuration isolates algorithmic stability and robustness under stochastic conditions, serving as a controlled foundation for future experiments that will incorporate explicit coordination and shared learning mechanisms. The following section details the experimental setup and the algorithms evaluated within this benchmark.

4 EXPERIMENTS

4.1 Computational Setup

BusEnv integrates seamlessly with established MARL ecosystems, providing standardized APIs and wrappers for interoperability. It adopts Gymnasium [35, 36] as the base API, extends compatibility through PettingZoo [33] for multi-agent interactions, and leverages SuperSuit [34] for preprocessing and environment wrappers. On top of this foundation, BusEnv supports MARLlib [14, 15], a high-level multi-agent reinforcement learning library built over Ray RLlib [21], enabling distributed and scalable training of diverse MARL algorithms.

Evaluation protocol. In our experimental configuration, bus agents were deployed within the environment without explicit policy sharing or inter-agent communication. Each bus acted as an independent learner, receiving local observations and optimizing its policy based solely on individual experience, while contributing to a shared team reward signal. This setup isolates algorithmic stability and robustness under stochastic transport conditions, allowing a controlled comparison of learning dynamics before introducing explicit coordination mechanisms. Although this configuration does not yet capture full cooperative behavior, it provides an essential baseline for understanding how independent learners adapt to partially observable, high-variance environments such as urban bus networks. Each episode corresponds to one operational day derived from real SUNT data, including route topology, stop sequences, passenger demand, and traffic patterns. We evaluate several MARL algorithms from MARLlib under this independent-learning regime, maintaining consistent hyperparameters and reward design across agents. The system scales to hundreds of agents in parallel through Ray's distributed architecture, ensuring reproducible large-scale experiments.

Hyperparameters. The hyperparameter configurations were grouped according to shared design characteristics. IA2C, COMA, and MAA2C adopt identical settings with Generalized Advantage Estimation (`use_gae = TRUE`), `lambda = 1.0`, `vf_loss_coeff = 1.0`, and batches of ten episodes using the *truncate_episodes* mode. Their learning rate is `lr = 0.0005` with `entropy_coeff = 0.01`, forming a minimal setup without clipping or Kullback–Leibler (KL) regularization. IPPO and MAPPO extend this baseline by introducing divergence control (`kl_coeff = 0.2`), multiple gradient updates (`num_sgd_iter = 5`), and policy clipping (`clip_param = 0.3`, `vf_clip_param = 10.0`) while preserving the same learning rate and entropy coefficient, promoting stability and exploration balance.

HATRPO, ITRPO, and MATRPO focus on *trust-region* optimization, sharing parameters such as `gamma = 0.99`, `kl_coeff = 0.2`, and conservative constraints (`kl_threshold = 0.00001`, `accept_ratio = 0.5`) with a smaller critic learning rate of `0.00005`. HAPPO combines both proximal and trust-region principles, using similar core parameters but with a smaller global learning rate (`lr = 0.0000005`), higher critic rate (`critic_lr = 0.05`), and additional regularization (`gain = 0.01`, `min_lr_schedule = 1e-11`). Overall, the configuration space evolves from simple independent actor–critic setups to more complex and stable schemes emphasizing KL control and cautious policy updates.

Code and data repository: https://github.com/LabIA-UFBA/BusEnv.

4.2 Baseline Methods for Benchmark Validation

To validate the proposed benchmark, we evaluate a diverse set of baseline algorithms implemented in MARLlib [14, 15]. Because the environment is modeled as a Decentralized Partially Observable Markov Decision Process (Dec-POMDP), each agent observes

only local information about the global state, intensifying the non-stationarity typical of independent learners. This property directly affects algorithms such as IPPO [8, 30], IA2C [15, 23], and ITRPO [15, 28], which optimize policies individually without centralized coordination. In contrast, methods based on centralized training with decentralized execution (CTDE), including MAA2C [26], COMA [10], MAPPO [37], MATRPO [20], HAPPO, and HATRPO [17], leverage a shared critic during training to stabilize learning and mitigate non-stationarity, while preserving decentralized decision-making at execution time.

This distinction between independent and CTDE paradigms motivates our choice of baselines, enabling a systematic comparison of their learning dynamics and robustness under the same partially observable and stochastic conditions of BusEnv. Although some of these methods, such as MAPPO and COMA, are designed for centralized training, in this study all algorithms were evaluated under the same independent-learning configuration described in Section 4.1, without explicit information sharing or policy coupling among agents.

Table 1 summarizes the evaluated algorithms by training paradigm, optimization family, and policy scope, while Table 2 introduces unified notation to enable consistent comparison of objectives, critics, and update rules. All baselines are policy-gradient methods, differing mainly in critic design, advantage estimation, and stability mechanisms (e.g., clipping or trust regions). We briefly describe each baseline and its key characteristics next.

Table 1: Baseline algorithms evaluated for validating the proposed real-world multi-agent benchmark.

Algorithm	Paradigm	Family	Scope
IPPO	Independent	PPO	Homogeneous
IA2C	Independent	Actor Critic	Homogeneous
ITRPO	Independent	TRPO	Homogeneous
MAA2C	CTDE	Actor Critic	Homogeneous
COMA	CTDE	Actor Critic	Homogeneous
MAPPO	CTDE	PPO	Homogeneous
MATRPO	CTDE	TRPO	Homogeneous
HAPPO	CTDE	PPO	Heterogeneous
HATRPO	CTDE	TRPO	Heterogeneous

Independent Proximal Policy Optimization (IPPO) extends Proximal Policy Optimization (PPO) [30] to multi-agent settings by training each agent independently under partial observability. Each agent a updates its policy $\pi_\theta^a(u_t^a|o_t^a)$ using local observations o_t^a and actions u_t^a.

The critic $V_\phi(o_t^a)$ is trained to approximate the expected return by minimizing:

$$\mathcal{L}_V(\phi) = \mathbb{E}_\tau\Big[\big(V_\phi(o_t^a) - R_t\big)^2\Big], \quad R_t = \sum_{k=0}^{T-1} \gamma^k r_{t+k}. \tag{1}$$

To evaluate how good an action u_t^a is compared to the critic's baseline, we estimate the *advantage* A_t. Intuitively, A_t measures the relative benefit of taking action u_t^a at observation o_t^a, compared to the average action suggested by the critic. A positive advantage means the action performed better than expected; a negative one means it performed worse.

Table 2: Notation used in baseline algorithms.

Symbol	Definition
$a \in \{1, \dots, n\}$	Agent index
o_t^a	Local observation of agent a at time t
u_t^a	Action of agent a at time t
s_t	Global state at time t
u_t	Joint action of all agents
r_t	Reward at time t
R_t	Discounted return, $\sum_{k=0}^{T-1} \gamma^k r_{t+k}$
γ	Discount factor
τ	Trajectory of (o_t^a, u_t^a, r_t)
π_θ^a	Policy of agent a, parameters θ
$\pi_{\theta_{\text{old}}}^a$	Previous policy
$V_\phi(o_t^a)$	Local value function (critic)
$V_\phi(s_t, u_t)$	Centralized value function (CTDE)
$Q(s_t, u_t)$	Centralized joint action–value
A_t	Advantage estimate (Eq. (2))
A_t^a	Counterfactual advantage (Eq. (7))
λ	GAE parameter (bias–variance trade-off)
$\rho_t(\theta)$	Importance ratio
ϵ	PPO clipping threshold
D_{KL}	Kullback–Leibler divergence
δ	Trust region threshold (TRPO)

Generalized Advantage Estimation (GAE) [29] is commonly used:

$$A_t = \sum_{l=0}^{\infty} (\gamma\lambda)^l \, \delta_{t+l}, \quad \delta_t = r_t + \gamma V_\phi(o_{t+1}^a) - V_\phi(o_t^a), \tag{2}$$

where λ balances bias and variance in the estimation.

The policy is optimized via the clipped surrogate objective:

$$\mathcal{L}_\pi(\theta) = \mathbb{E}_\tau\Big[\min\big(\rho_t(\theta)A_t,\ \text{clip}(\rho_t(\theta), 1-\epsilon, 1+\epsilon)A_t\big)\Big], \tag{3}$$

where $\rho_t(\theta) = \frac{\pi_\theta^a(u_t^a|o_t^a)}{\pi_{\theta_{\text{old}}}^a(u_t^a|o_t^a)}$ is the importance ratio.

Independent Advantage Actor–Critic (IA2C) is the independent extension of A2C [23]. The critic follows the temporal-difference (TD) loss:

$$\mathcal{L}_V(\phi) = \mathbb{E}_\tau\Big[\big(V_\phi(o_t^a) - (r_t + \gamma V_\phi(o_{t+1}^a))\big)^2\Big], \tag{4}$$

while the policy is optimized with the gradient:

$$\nabla_\theta J(\pi_\theta^a) = \mathbb{E}_\tau\Big[\nabla_\theta \log \pi_\theta^a(u_t^a|o_t^a)\, A_t\Big]. \tag{5}$$

Independent Trust Region Policy Optimization (ITRPO) adapts TRPO [28] to independent training. The critic update follows Eq. (4), while the policy update is formulated as:

$$\max_\theta \ \mathbb{E}_\tau\big[\rho_t(\theta)A_t\big], \quad \text{s.t. } \mathbb{E}_\tau\big[D_{\text{KL}}(\pi_{\theta_{\text{old}}}^a \,\|\, \pi_\theta^a)\big] \le \delta. \tag{6}$$

Multi-Agent Advantage Actor–Critic (MAA2C) extends A2C to CTDE. Policies remain decentralized, but a centralized critic $V_\phi(s_t, u_t)$ replaces the local one in Eq. (5), improving coordination.

Counterfactual Multi-Agent Policy Gradients (COMA) introduces a counterfactual baseline for credit assignment. The centralized critic $Q(s_t, u_t)$ estimates joint values, and the counterfactual advantage is:

$$A_t^a = Q(s_t, u_t) - \sum_{u'^a} \pi^a(u'^a|o_t^a)\, Q(s_t, (u'^a, u_t^{-a})). \tag{7}$$

The first term $Q(s_t, u_t)$ reflects the joint value of the executed action profile, while the second term is a baseline that averages over all possible actions u'^a of agent a, weighted by its current policy $\pi^a(u'^a|o_t^a)$, with the other agents' actions u_t^{-a} held fixed. This formulation measures the marginal contribution of agent a's actual action compared to what would have happened had it sampled according to its policy. By isolating each agent's causal impact on the team outcome, COMA improves gradient estimates, reduces variance, and provides a principled solution to the credit assignment problem in cooperative MARL.

Multi-Agent Proximal Policy Optimization (MAPPO) extends PPO to CTDE by leveraging a centralized critic $V_\phi(s_t, u_t)$ for advantage estimation, while policies remain decentralized. The surrogate objective follows Eq. (3), but the use of centralized information in the critic stabilizes training in cooperative settings and improves coordination across agents.

Multi-Agent Trust Region Policy Optimization (MATRPO) extends TRPO to CTDE, combining the trust region update of Eq. (6) with a centralized critic. Unlike MAPPO, which heuristically constrains updates via clipping, MATRPO enforces a strict Kullback–Leibler divergence constraint on the policy step, yielding more conservative updates.

Heterogeneous-Agent PPO (HAPPO) generalizes MAPPO to heterogeneous agents, where each agent maintains a distinct policy π_θ^a. While critics remain centralized, the surrogate objective in Eq. (3) is applied independently to each agent's policy, allowing specialization of behaviors. This makes HAPPO suitable for environments where agents have asymmetric roles or action spaces, while still benefiting from centralized training signals.

Heterogeneous-Agent TRPO (HATRPO) extends MATRPO to the heterogeneous setting, combining distinct policies per agent with the trust region constraint of Eq. (6). Each agent's policy update is restricted by its own Kullback–Leibler divergence bound, while the centralized critic provides consistent advantage estimates across agents. This combination preserves the monotonic improvement guarantees of TRPO, while enabling agents with different roles or dynamics to learn coordinated yet specialized behaviors.

4.3 Results

Figures 2 and 3, together with Table 3, jointly summarize the performance of nine MARL algorithms in the BusEnv benchmark. Each algorithm was trained under identical environmental conditions using the same reward structure and agent configuration previously described.

Learning dynamics. As illustrated in Figure 2, algorithms such as IA2C, IPPO, ITRPO, MAA2C, MAPPO, and MATRPO exhibit smooth and monotonic convergence, reaching normalized rewards near 1.0 within fewer than 2×10^5 training steps. This behavior indicates that, even under independent-learning conditions, these methods can efficiently adapt to the stochastic fluctuations of the transport environment and achieve stable policy updates. In contrast, COMA and HAPPO display unstable or stagnant learning patterns: COMA initially improves but later oscillates, while HAPPO remains nearly flat throughout training, failing to accumulate meaningful reward. These patterns highlight the sensitivity of algorithms that rely on centralized critics or trust-region constraints when explicit coordination signals are absent.

Figure 3 complements this view by revealing consistent patterns in the compiled metrics of energy distribution and power balance, showing that algorithms with smoother convergence also yield more stable operational profiles at the system level. Thus, the observed differences primarily reflect each algorithm's ability to maintain robustness and efficiency under partially observable, non-stationary dynamics rather than explicit inter-agent cooperation.

Quantitative comparison. Each MARL algorithm was trained for 400,000 timesteps over 50 independent runs. The final average reward was obtained by averaging the episodic rewards from the last evaluation window, while the Area Under the Curve (AUC) was computed from the normalized reward curve across training. These metrics jointly capture converged performance, learning efficiency, and training stability, with results reported as mean ± standard deviation across runs.

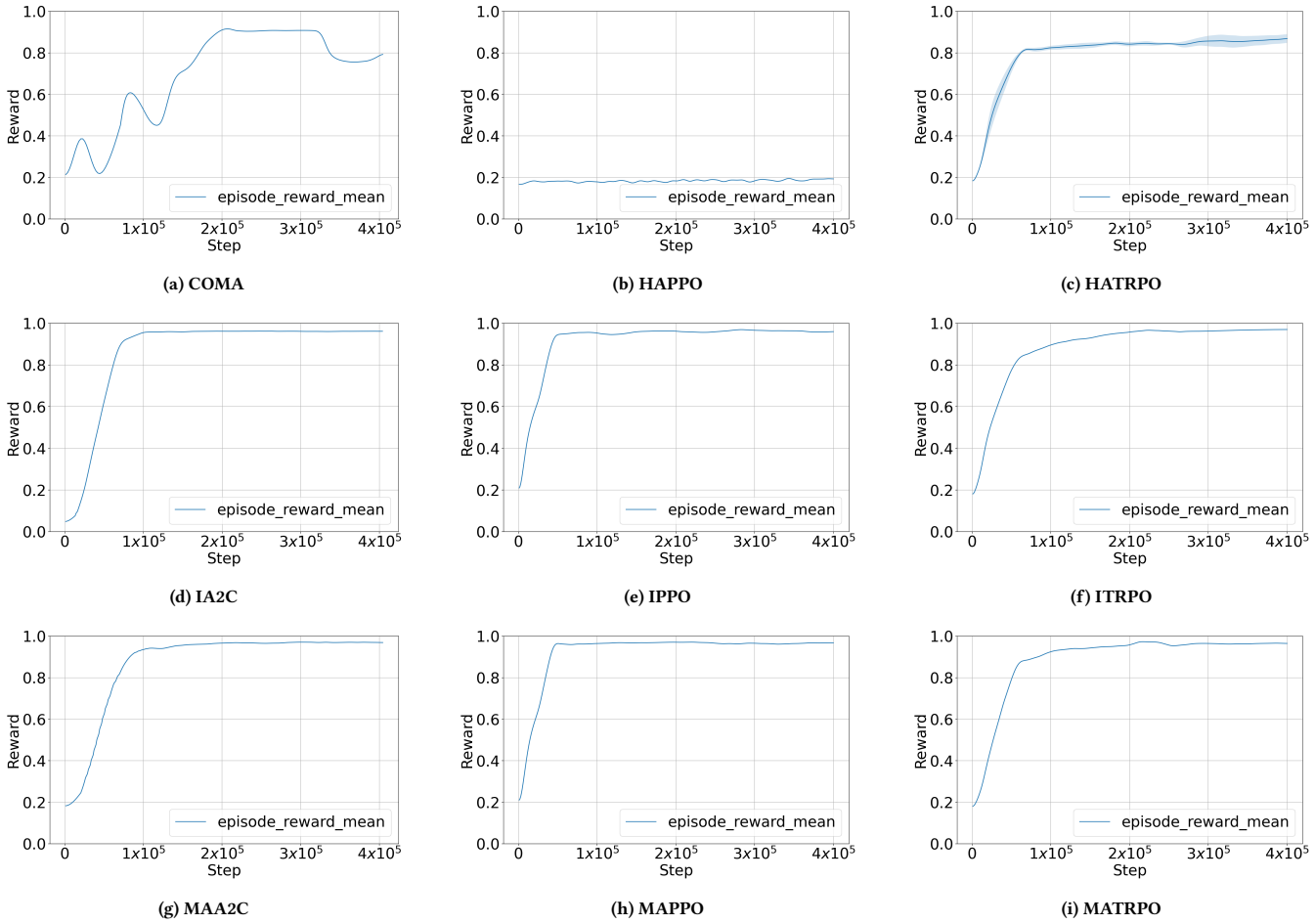
The numerical results (Table 3) reinforce the visual trends. MAPPO achieved the highest final reward (75.31 ± 0.00) and the largest AUC (0.924), confirming its superior stability and consistent gradient updates under stochastic conditions. IPPO follows closely with an AUC of 0.917, though its slightly lower final reward suggests faster early adaptation but a more limited asymptotic performance.

Algorithms such as IA2C, MAA2C, ITRPO, and MATRPO cluster tightly around rewards ≈ 73–77 and AUC ≈ 0.87–0.89, reflecting stable yet slightly slower convergence across runs. By contrast, HATRPO exhibits large variance (±15.43) and an intermediate AUC (0.79), while COMA and especially HAPPO yield negative rewards and very low AUC values (0.70 and 0.18, respectively), indicating severe training instability and failure to adapt reliably under independent-learning conditions.

Table 3: Final average reward and AUC across runs for each MARL algorithm.

Algorithm	Final Reward	AUC
COMA	-73.69 ± 2.17	0.704
HAPPO	-506.84 ± 1.25	0.183
HATRPO	-2.53 ± 15.43	0.794
IA2C	71.28 ± 0.15	0.871
IPPO	69.45 ± 0.00	0.917
ITRPO	77.16 ± 0.00	0.883
MAA2C	77.41 ± 0.24	0.879
MAPPO	75.31 ± 0.00	0.924
MATRPO	73.95 ± 0.00	0.887

Sustainability Footprint. Figure 3(a–c) shows that unstable learning dynamics substantially increase CO_2 emissions and energy consumption, leading to non-robust, sub-optimal control policies misaligned with Green AI principles. High-variance algorithms such as COMA and HAPPO exhibit the largest environmental costs,



(a) COMA (b) HAPPO (c) HATRPO

(d) IA2C (e) IPPO (f) ITRPO

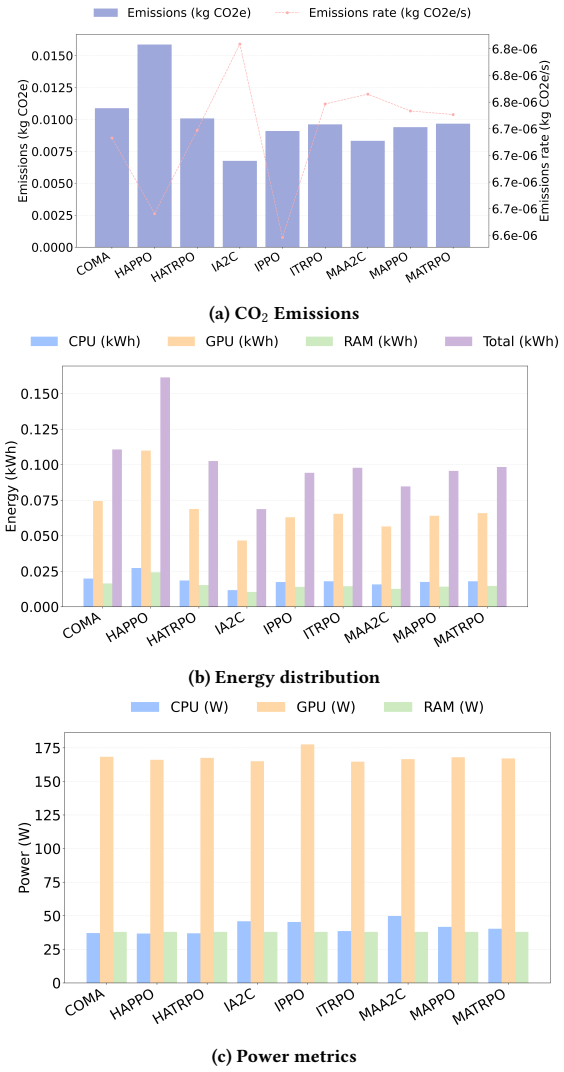
(g) MAA2C (h) MAPPO (i) MATRPO

Figure 2: Average episode reward across different MARL algorithms in the BusEnv benchmark. Each subplot shows the *episode_reward_mean* curve during training, highlighting convergence trends and stability.

indicating that instability in policy optimization directly translates into operational energy waste and higher carbon intensity, underscoring robustness as a key requirement for sustainable performance. Such results were obtained using the CodeCarbon [6] tool.

General Discussion. These findings highlight the robustness of PPO-based architectures in multi-agent reinforcement learning, even under independent-learning conditions. Algorithms such as MAPPO outperform other policy-gradient and trust-region methods by effectively stabilizing updates and maintaining consistent gradient flow under partial observability. Similarly, actor-critic variants such as IA2C and MAA2C remain competitive, indicating that their value-based guidance mechanisms can sustain learning stability in stochastic, high-dimensional environments. In contrast, algorithms relying on heavy trust-region constraints, such as HATRPO and HAPPO, appear overly conservative and struggle to adapt to the non-stationary dynamics of BusEnv. The combined evidence from learning curves and summary statistics suggests that PPO-based optimization provides the best compromise between convergence speed, final performance, and resilience to environmental stochasticity. Moreover, the consistent energy and emission profiles observed in Figure 3 indicate that stable policies not only maximize cumulative reward but also promote sustainable operational behavior, reducing unnecessary variability and energy waste. This relationship between algorithmic stability and environmental efficiency reinforces the role of robustness as a key driver for sustainability-aware reinforcement learning in real-world transport systems.

Interpretation in transportation context. The differences observed in Table 3 can be directly explained by the multi-objective structure of the reward function R. Since R combines positive incentives (service quality and operational efficiency) with negative penalties (maintenance violations, congestion, and excessive idling), the cumulative return of each algorithm reflects its ability to balance these competing factors over time. Algorithms that consistently minimize penalty components while maximizing service-related terms (e.g., MAPPO, IA2C, and MAA2C) achieve high and stable positive rewards, indicating robust adaptation to the stochastic conditions of the transport environment. Their training dynamics



(a) CO_2 Emissions

(b) Energy distribution

(c) Power metrics

Figure 3: Comparison of the environmental and energetic outcomes across the evaluated MARL algorithms, emphasizing the impact of policy instability.

promote smoother gradient updates and better value estimation, allowing each independent agent to sustain efficient operational behavior without repeatedly violating the environment's operational constraints.

In contrast, algorithms such as COMA, HATRPO, and HAPPO accumulate substantially negative rewards. Despite their strong theoretical underpinnings, these methods tend to amplify the penalty terms embedded in the reward function when applied under independent-learning conditions. Instabilities in value estimation (in COMA) or excessive conservatism and imprecise updates within trust-region frameworks (in HATRPO and HAPPO) lead to erratic or inefficient control patterns. In a transit setting, this behavior can manifest as undesirable operational actions such as simultaneous departures, prolonged idling, or route-level congestion, conditions that trigger large penalties and degrade overall system efficiency. Consequently, negative cumulative rewards correspond to severe inefficiencies analogous to bus bunching, missed service opportunities, and unnecessary energy waste.

This outcome reinforces that effective algorithms must go beyond optimizing convergence speed or variance reduction: they must balance the intrinsic trade-offs encoded in R to remain stable under stochastic disturbances. Achieving positive final rewards thus reflects the capacity to internalize operational objectives such as maintaining service regularity, balancing occupancy, and respecting maintenance limits, even in the absence of explicit inter-agent communication. Conversely, persistent negative rewards reveal instability in policy optimization and poor resilience to the non-stationary dynamics that characterize real-world urban transport systems.

5 CONCLUSION

The diversity of learning behaviors observed across algorithms validates BusEnv as a comprehensive benchmark for evaluating robustness and adaptability in reinforcement learning for public transport systems. Its detailed spatio-temporal representation and stochastic passenger flows expose the strengths and limitations of MARL methods when applied to dynamic, data-driven environments. By combining real operational data with reproducible experimental settings, BusEnv enables systematic analysis of algorithmic stability, sensitivity to stochasticity, and overall resilience in decision-making. Furthermore, the environment was designed to support extensibility, facilitating the future integration of explicit inter-agent communication, shared critics, and hybrid control strategies such as transfer-point reinforcement, adaptive energy allocation, and demand-responsive dispatching. This flexibility establishes BusEnv not only as a benchmark for algorithmic comparison but also as a testbed for innovation in data-driven urban mobility planning.

The stable convergence of algorithms such as MAPPO, together with the interpretability of their learned dynamics and the efficiency of their computational behavior, provides strong evidence that reinforcement learning can contribute to the broader goals of Green AI. Within the BusEnv framework, the derived policies demonstrate that learning processes can be optimized not only for performance but also for energy-efficient and resource-aware operation. This connection between algorithmic stability and environmental efficiency reinforces the value of BusEnv as a platform for studying how sustainability-oriented principles can be systematically integrated into large-scale reinforcement learning research. Future work will extend these findings by incorporating cooperative settings and collective optimization, enabling a deeper exploration of emergent coordination and its impact on sustainable urban transport systems.

ACKNOWLEDGMENTS

This work was supported by CNPq (Brazilian National Council for Scientific and Technological Development) grants [404771/2024-6, 406354/2023-5, 312755/2023-6, 313053/2023-5], UFBA/CNPq 68/2022 - MAI/DAI, Maria Emilia Foundation grant to 01/2023, INCITE FAPESB (Bahia Research Foundation) grant TO PIE0002/2022, CAPES (Coordination for the Improvement of Higher Education Personnel – Brazil), and FAPESB grant [1589/2021].

REFERENCES

[1] 2003. Real-time control of buses for schedule coordination at a terminal. *Transportation Research Part A: Policy and Practice* 37, 2 (2003), 145–164. https://doi.org/10.1016/S0965-8564(02)00010-1

[2] John J. Bartholdi and Donald D. Eisenstein. 2012. A self-coordinating bus route to resist bus bunching. *Transportation Research Part B: Methodological* 46, 4 (2012), 481–491. https://doi.org/10.1016/j.trb.2011.11.001

[3] Dominik Jacek Bogucki, Łukasz Lepak, Sonam Parashar, Bartłomiej Błachowski, and Paweł Wawrzyński. 2025. EnEnv 1.0: Energy Grid Environment for Multi-Agent Reinforcement Learning Benchmarking. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025)*. 361–370.

[4] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. 2018. *JAX: composable transformations of Python+NumPy programs.* http://github.com/jax-ml/jax

[5] Erwin Coumans and Yunfei Bai. 2021. PyBullet quickstart guide. *ed: PyBullet Quickstart Guide. https://docs. google. com/document/u/1/d* (2021).

[6] Benoit Courty, Victor Schmidt, Sasha Luccioni, Goyal-Kamal, MarionCoutarel, Boris Feld, Jérémy Lecourt, LiamConnell, Amine Saboni, Inimaz, supatomic, Mathilde Léval, Luis Blanche, Alexis Cruveiller, ouminasara, Franklin Zhao, Aditya Joshi, Alexis Bogroff, Hugues de Lavoreille, Niko Laskaris, Edoardo Abati, Douglas Blank, Ziyao Wang, Armin Catovic, Marc Alencon, Michał Stęchły, Christian Bauer, Lucas Otávio N. de Araújo, JPW, and MinervaBooks. 2024. *mlco2/codecarbon: v2.4.1.* https://doi.org/10.5281/zenodo.11171501

[7] Carlos F. Daganzo. 2009. A headway-based approach to eliminate bus bunching: Systematic analysis and comparisons. *Transportation Research Part B: Methodological* 43, 10 (2009), 913–921. https://doi.org/10.1016/j.trb.2009.04.002

[8] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. 2020. Is independent learning all you need in the starcraft multi-agent challenge? *arXiv preprint arXiv:2011.09533* (2020).

[9] Benjamin Ellis, Jonathan Cook, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob Foerster, and Shimon Whiteson. 2023. Smacv2: An improved benchmark for cooperative multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 36 (2023), 37567–37593.

[10] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. 2018. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[11] K. Gkiotsalitis and O. Cats. 2021. At-stop control measures in public transport: Literature review and research agenda. *Transportation Research Part E: Logistics and Transportation Review* 145 (2021), 102176. https://doi.org/10.1016/j.tre.2020.102176

[12] Leonard Hinckeldey, Elliot Fosong, Elle Miller, Rimvydas Rubavicius, Trevor McInroe, Patricia Wollstadt, Christiane B. Wiebel-Herboth, Subramanian Ramamoorthy, and Stefano V. Albrecht. 2025. Assistax: A Hardware-Accelerated Reinforcement Learning Benchmark for Assistive Robotics. In *RLC 2025 Workshop on Coordination and Cooperation in Multi-Agent Reinforcement Learning*.

[13] Joshua Holder, Natasha Jaques, and Mehran Mesbahi. 2025. Multi agent reinforcement learning for sequential satellite assignment problems *(AAAI'25/IAAI'25/EAAI'25)*. AAAI Press, Article 2954, 9 pages.

[14] Siyi Hu, Yifan Zhong, Minquan Gao, Weixun Wang, Hao Dong, Zhihui Li, Xiaodan Liang, Yaodong Yang, and Xiaojun Chang. 2022. Marllib: Extending rllib for multi-agent reinforcement learning. (2022).

[15] Siyi Hu, Yifan Zhong, Minquan Gao, Weixun Wang, Hao Dong, Xiaodan Liang, Zhihui Li, Xiaojun Chang, and Yaodong Yang. 2023. Marllib: A scalable and efficient multi-agent reinforcement learning library. *Journal of Machine Learning Research* 24, 315 (2023), 1–23.

[16] Mingyu Kim, Jihwan Oh, Yongsik Lee, Joonkee Kim, Seonghwan Kim, Song Chong, and Seyoung Yun. 2023. The StarCraft multi-agent exploration challenges: Learning multi-stage tasks and environmental factors without precise reward functions. *IEEE Access* 11 (2023), 37854–37868.

[17] Jakub Grudzien Kuba, Ruiqing Chen, Muning Wen, Ying Wen, Fanglei Sun, Jun Wang, and Yaodong Yang. 2022. Trust Region Policy Optimisation in Multi-Agent Reinforcement Learning. In *International Conference on Learning Representations*. https://openreview.net/forum?id=EcGGFkNTxdJ

[18] Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, et al. 2020. Google research football: A novel reinforcement learning environment. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 4501–4510.

[19] Pascal Leroy, Pablo G. Morato, Jonathan Pisane, Athanasios Kolios, and Damien Ernst. 2023. IMP-MARL: a suite of environments for large-scale infrastructure management planning via MARL. In *Proceedings of the 37th International Conference on Neural Information Processing Systems(NeurIPS)*. Article 2329.

[20] Hepeng Li and Haibo He. 2023. Multiagent trust region policy optimization. *IEEE Transactions on Neural Networks and Learning Systems* 35, 9 (2023), 12873–12887.

[21] Eric Liang, Richard Liaw, Robert Nishihara, Philipp Moritz, Roy Fox, Ken Goldberg, Joseph Gonzalez, Michael Jordan, and Ion Stoica. 2018. RLlib: Abstractions for distributed reinforcement learning. In *International conference on machine learning*. PMLR, 3053–3062.

[22] Fangqi Liu, Rishav Sen, Jose Paolo Talusan, Ava Pettet, Aaron Kandel, Yoshinori Suzue, Ayan Mukhopadhyay, and Abhishek Dubey. 2025. Reinforcement Learning-based Approach for Vehicle-to-Building Charging with Heterogeneous Agents and Long Term Rewards. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 1345–1358.

[23] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. 2016. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*. PmLR, 1928–1937.

[24] Frans A Oliehoek, Christopher Amato, et al. 2016. *A concise introduction to decentralized POMDPs*. Vol. 1. Springer.

[25] George Papadopoulos, Andreas Kontogiannis, Foteini Papadopoulou, Chaido Poulianou, Ioannis Koumentis, and George Vouros. 2025. An Extended Benchmarking of Multi-Agent Reinforcement Learning Algorithms in Complex Fully Cooperative Tasks. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (Detroit, MI, USA) *(AAMAS '25)*. 1613–1622.

[26] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2021. Benchmarking multi-agent deep reinforcement learning algorithms in cooperative tasks. In *Proceedings of the 35th Conference on Neural Information Processing Systems (NeurIPS)*.

[27] Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The StarCraft Multi-Agent Challenge. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS)*. 2186–2188.

[28] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *International conference on machine learning*. PMLR, 1889–1897.

[29] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* (2015).

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* (2017).

[31] Yan Song, He Jiang, Zheng Tian, Haifeng Zhang, Yingping Zhang, Jiangcheng Zhu, Zonghong Dai, Weinan Zhang, and Jun Wang. 2024. An empirical study on google research football multi-agent scenarios. *Machine Intelligence Research* 21, 3 (2024), 549–570.

[32] Yan Song, He Jiang, Haifeng Zhang, Zheng Tian, Weinan Zhang, and Jun Wang. 2024. Boosting Studies of Multi-Agent Reinforcement Learning on Google Research Football Environment: The Past, Present, and Future. In *Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (Auckland, New Zealand) *(AAMAS '24)*. 1772–1781.

[33] Jordan Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. 2021. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems* 34 (2021), 15032–15043.

[34] J. K Terry, Benjamin Black, and Ananth Hari. 2020. SuperSuit: Simple Microwrappers for Reinforcement Learning Environments. *arXiv preprint arXiv:2008.08932* (2020).

[35] Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. 2024. Gymnasium: A Standard Interface for Reinforcement Learning Environments. *arXiv preprint arXiv:2407.17032* (2024).

[36] Mark Towers, Jordan K. Terry, Ariel Kwiatkowski, John U. Balis, Gianluca de Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Arjun KG, Markus Krimmel, Rodrigo PerezVicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Andrew Tan Jin Shen, and Omar G. Younis. 2023. Gymnasium. https://github.com/Farama-Foundation/Gymnasium. Farama Foundation.

[37] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The surprising effectiveness of ppo in cooperative multi-agent games. *Advances in neural information processing systems* 35 (2022), 24611–24624.

[38] Rui Zhao, Xu Liu, Yizheng Zhang, Minghao Li, Cheng Zhou, Shuai Li, and Lei Han. 2023. CraftEnv: A Flexible Collective Robotic Construction Environment for Multi-Agent Reinforcement Learning. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023)*. 1164–1172.