PRISM: EVENT-DRIVEN MULTI-AGENT RESOURCE COLLECTION WITH ADAPTIVE PREFERENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-agent resource collection, exemplified by the Multi-agent Traveling Officer Problem (MTOP), requires mobile agents with limited capacity to capture spatially distributed resources whose availability evolves over time. Existing methods optimize only collected volume and rely on uniform-step formulations, which introduce uninformative no-op transitions and ignore the coupling among agents dispatched at the same time. Real-world dispatch, however, must also balance timeliness and fairness. We propose **PRISM**, which formulates MTOP as an event-driven asynchronous multi-agent semi-Markov decision process in which decisions are triggered only by agent availability. A state-aware context encoder models interactions among agents, resources, and dispatch history; a semi-autoregressive decoder generates joint assignments efficiently while preserving inter-assignment dependencies; and an adaptive preference head infers agent-specific weights over Resource Volume, Timeliness, and Fairness, letting agents specialize to their local opportunities and team needs. On four public datasets spanning crime, emergency medical services, and traffic, PRISM outperforms the strongest multi-objective baselines by 16.98%, 6.19%, and 4.81% on Volume, Timeliness, and Fairness, respectively. Code: https://anonymous.4open.science/r/PRISM-C5BA.

1 INTRODUCTION

Many urban sensing and service systems generate dynamic spatio-temporal graph data in which mobile agents must collect, inspect, or serve resources that evolve over a spatial network (Gong et al., 2018; Wei & Zheng, 2020; Zheng et al., 2014; Zanella et al., 2014), as in urban patrol, mobile inspection, and emergency response (Fu et al., 2020; Haghani et al., 2004; Wang et al., 2024). In urban patrol, for instance, crimes emerge at different locations and persist only for a limited period, and police officers (the agents) must travel through the road network to respond to them (the resources) before they disappear. Resources in such problems are therefore not static targets that can be visited at arbitrary times: agents must decide where to move under travel-time constraints and limited service capacity (Psaraftis et al., 2016; Ritzinger et al., 2016), and decision quality depends not only on which resources are valuable but also on whether an agent can reach them at the right time (Tang et al., 2019). The Traveling Officer Problem (TOP) (Shao et al., 2017) captures this setting as stochastic resource collection (Schmoll & Schubert, 2021; Strauss et al., 2022), and we use TOP in this generalized sense throughout the paper. Unlike classical routing problems such as the Traveling Salesman Problem (Lawler, 1985), TOP involves time-sensitive and uncertain resources rather than a fixed set of static targets.

The Multi-agent Traveling Officer Problem (MTOP) extends TOP to a fleet of agents operating in a shared environment. It is not a simple parallelization of single-agent TOP: agents must be coordinated to avoid redundant dispatches, exploit spatial complementarity, and allocate limited service capacity across changing resources. Prior MTOP studies have proposed heuristic, meta-heuristic, and learning-based methods for assigning agents to targets (Qin et al., 2020; Strauss et al., 2022), but they leave four challenges unresolved. **(1) Formulation mismatch.** MTOP is naturally driven by dispatch events, i.e., the moments at which one or more agents become available. Step-based RL and MARL formulations (Yu et al., 2022) advance all agents on a uniform clock and emulate this asynchrony with virtual no-op steps for agents still in transit, while per-agent semi-Markov formulations (Schmoll & Schubert, 2021; Strauss et al., 2022) model each agent's timing separately.

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.

Neither treats the dispatch event itself as the unit of decision, which requires handling variable durations between events and a variable set of available agents at each event. **(2) Structured assignment coupling.** Within one event, assigning an agent to a resource changes the feasible choices and assignment quality of the agents assigned after it. Heuristic methods usually select targets per agent and learning-based methods largely generate each agent's action independently, so these inter-assignment dependencies are ignored. **(3) Volume-oriented optimization.** Existing TOP/MTOP methods mainly maximize captured resource volume, which can drive agents to compete for high-volume hotspot nodes, whereas realistic dispatch must also account for temporal responsiveness and service fairness, and the best choices for these objectives often differ from the highest-volume resources. **(4) Static objective tradeoffs.** Even when multiple objectives are considered, a fixed global tradeoff is restrictive, because agents face different local opportunities at the same event: one agent may be well placed for resource collection while another is better positioned for timeliness or fairness. Fixed preferences suppress this flexibility and collapse agents into homogeneous behaviors. Together, these challenges call for an event-driven, structured, and adaptive multi-objective dispatch framework.

We propose PRISM, an event-driven multi-agent resource collection framework with adaptive preferences. PRISM formulates MTOP as an asynchronous multi-agent semi-Markov decision process (SMDP) whose unit of decision is the system-level dispatch event rather than a uniform time step or an individual agent: an event is triggered when one or more agents become available, decisions are made only at events with no virtual steps in between, and returns are discounted by the actual elapsed time (formal definition in Appendix A). Within each event, PRISM produces a joint assignment of all available agents. A state-aware context encoder captures the interactions among agents, resources, and dispatch history, and a semi-autoregressive decoder scores all agent–resource pairs in a single batched pass and then commits assignments sequentially, so that each assignment is conditioned on those made before it. Beyond volume, PRISM optimizes three operational objectives: **Resource Volume**, the amount of dynamic resource captured upon arrival; **Timeliness**, which rewards arriving when a destination is near a recent local peak; and **Fairness**, which encourages serving locations that have remained unvisited for longer. An adaptive preference head computes separate sub-scores for these objectives and infers per-agent preferences from local constraints and global team context, so that agents adapt their tradeoffs to the current dispatch conditions.

Our contributions are summarized as follows:

- We formulate MTOP as an asynchronous multi-agent SMDP whose unit of decision is the system-level dispatch event. It contains no virtual no-op steps, treats each event as a single decision over all agents that become available at it, and is trained with an SMDP-aware advantage that discounts by the actual elapsed time between events.
- We propose PRISM, a structured sequential dispatch framework in which a state-aware context encoder models interactions among global signals, resource nodes, agents, and recent dispatch history, and a semi-autoregressive decoder captures the coupling among agents, resources, and previous assignments, enabling efficient joint dispatch while preserving inter-assignment dependencies.
- We introduce an adaptive preference head that infers per-agent tradeoffs over Resource Volume, Timeliness, and Fairness, allowing agents to specialize according to their local opportunities and team-level needs and moving multi-objective dispatch beyond hotspot chasing.
- Experiments on four public datasets spanning crime, emergency medical services, and traffic show that PRISM outperforms the strongest multi-objective baselines by 16.98%, 6.19%, and 4.81% on Volume, Timeliness, and Fairness, respectively.

2 Related Work

The Traveling Officer Problem (TOP) was originally introduced to model parking enforcement supported by sensor data, where a mobile officer is routed through an urban road network to capture violations whose value diminishes over time (Shao et al., 2017). Subsequent TOP studies have largely followed this parking enforcement benchmark, but this setting represents only one specific instance of the broader problem of stochastic resource collection, where spatially distributed resources may

change value over time, and agents must decide where to move under time constraints and uncertain resource dynamics. Therefore, although TOP is closely related to classical routing problems such as the Traveling Salesman Problem (TSP) (Lawler, 1985) and the Vehicle Routing Problem (VRP) (Dantzig & Ramser, 1959), it cannot be reduced to visiting a fixed set of static targets.

Early TOP methods mainly rely on probabilistic modeling and combinatorial optimization. Shao et al. (Shao et al., 2017) estimated vehicle leaving probabilities from spatio-temporal parking data and proposed greedy and ant colony optimization methods to improve patrol strategies. Subsequent studies incorporated feature learning into this pipeline by using LSTM auto-encoders and clustering to extract spatio-temporal features before applying optimization methods such as genetic algorithms and simulated annealing (Shao et al., 2020). Neural combinatorial optimization methods, including pointer networks and attention models, have been used to learn routing policies for TSP, VRP, and related problems (Vinyals et al., 2015; Kool et al., 2018; Bengio et al., 2021; Alanzi & Menai, 2025). Inspired by these advances, recent TOP methods have introduced neural architectures such as heterogeneous pointer networks (He et al., 2022), dynamic graph attention networks (Zhang et al., 2022), and spatial-aware reinforcement learning models (Strauß & Schubert, 2024). These models improve representation learning for dynamic routing, but all of them are designed for single-agent and single-objective settings.

For the multi-agent extension of TOP, namely MTOP, prior studies have explored population-based algorithms with specialized solution encoding schemes (Qin et al., 2020). These methods are effective in specific settings, but they typically rely on predefined single-objective optimization criteria and hand-crafted search procedures, making it difficult to learn flexible multi-agent dispatch policies from dynamic graph states. On the other hand, directly applying conventional step-based multi-agent reinforcement learning to MTOP is not fully aligned with its event-driven structure, because agents become available asynchronously after executing actions with different travel durations. Existing studies have further formulated multi-agent stochastic resource collection as a semi-Markov game to handle temporal abstraction and asynchronous actions (Strauss et al., 2022; Schmoll & Schubert, 2021). However, existing methods still do not fully model multi-agent decisions within the same dispatch event as a structured sequential assignment problem over available agents and candidate resources. Moreover, most methods focus only on captured resource volume, which may induce hotspot chasing while neglecting responsiveness and service fairness.

3 METHOD

Fig. 1 presents the overall architecture of PRISM. At each dispatch event, PRISM builds heterogeneous feature tokens from the current and historical spatio-temporal graph states. A state-aware context encoder then jointly encodes these tokens through self-attention and produces contextual representations for the global state, candidate nodes, and currently free agents. Then, the adaptive preference head infers agent-specific objective weights over Resource Volume, Timeliness, and Fairness with these contexts. The semi-autoregressive decoder computes separate sub-scores for each objective over agent and node pairs, fuses them with the learned preferences, and generates joint dispatch assignments through a constrained-first loop with feasibility mask.

3.1 STATE-AWARE CONTEXT ENCODER

The encoder is responsible for transforming the event-level dispatch state into contextual representations. Since MTOP involves heterogeneous information, including global control signals, node states, agent states, and recent dispatch history, directly treating the input as a flat feature vector would discard important structural distinctions. We therefore design a state-aware context encoder that converts the heterogeneous input into a unified token sequence and jointly encodes all token groups through self-attention.

Given the current dispatch event, the token builder constructs four groups of tokens: control tokens, node tokens, agent tokens, and history tokens. The control tokens represent event-level information, such as remaining episode time and team-level objective weights. Node tokens encode resource locations and their dynamic resource signals. Agent tokens describe agent states, including positions, availability, and dispatch-related status. History tokens summarize recent node resource volume and dispatch records, providing short-term temporal context. These tokens are concatenated in the order

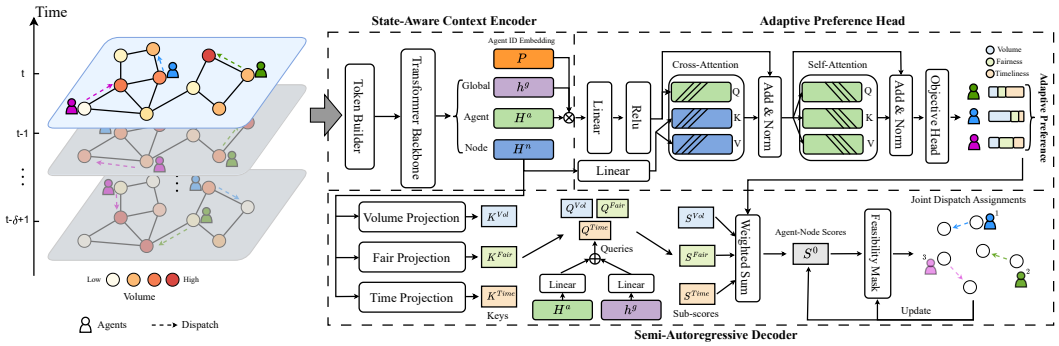


Figure 1: Framework of PRISM for MTOP resource collection. The context encoder transforms heterogeneous event-level inputs into global, node, and free agent contexts. The adaptive preference head infers agent-specific objective preference weights, while the semi-autoregressive decoder combines objective sub-scores with these preferences and resolves feasible joint assignments.

of control, nodes, agents, and history to form a single dispatch-state sequence. The Transformer backbone then maps the heterogeneous token sequence into a shared latent space. Each token feature is first projected into the latent dimension. To preserve group semantics, the model adds a group type and positional embedding, followed by layer normalization and dropout. The resulting sequence is processed by stacked Transformer encoder layers, where self-attention enables information exchange across all token types. This design allows the encoder to jointly model global control state, node state, agent state, and recent dispatch history.

Let $\mathbf{x} \in \mathbb{R}^{N_s \times d_{\text{in}}}$ denote the token feature matrix for the current event, where N_s is the token sequence length and d_{in} is the input dimension. Let $\mathbf{E}_{\text{type}}$ and $\mathbf{E}_{\text{pos}}$ denote the group type and positional embeddings, respectively. The input to the Transformer encoder is

$$\mathbf{z}^{(0)} = \text{LayerNorm}\left(\mathbf{x}\mathbf{W}_{\text{in}} + \mathbf{E}_{\text{type}} + \mathbf{E}_{\text{pos}}\right), \tag{1}$$

where $\mathbf{W}_{\text{in}} \in \mathbb{R}^{d_{\text{in}} \times d}$ projects raw token features into the latent dimension d. The encoded sequence is then obtained by

$$\mathbf{z} = \text{TransformerEncoder}(\mathbf{z}^{(0)}), \quad \mathbf{z} \in \mathbb{R}^{N_s \times d}. \tag{2}$$

This formulation keeps all heterogeneous dispatch tokens in a shared self-attention space while preserving their semantic group identities.

After encoding, the output sequence is split back into semantic groups: the global context, node contexts, and the free agent contexts:

$$(\mathbf{h}^g, \mathbf{H}^n, \mathbf{H}^a), \tag{3}$$

where $\mathbf{h}^g \in \mathbb{R}^d$ denotes the global context, $\mathbf{H}^n \in \mathbb{R}^{M \times d}$ denotes node contexts, and $\mathbf{H}^a \in \mathbb{R}^{N_f \times d}$ denotes the contexts of currently free agents; M is the number of nodes, and N_f is the number of free agents. Although history tokens and busy agent tokens are included in the input sequence, they are not retained as separate downstream contexts. Instead, their information is injected into the global, node, and free agent representations through self-attention during encoding.

3.2 Adaptive Preference Head

The adaptive preference head is designed to infer an objective tradeoff for each free agent before the decoder evaluates candidate resources. Instead of assigning all agents the same fixed preference, PRISM allows different agents to emphasize different objectives according to their own states and the states of other free agents. This design supports coordinated specialization: for example, one agent may focus more on resource volume, while another may prioritize timeliness or fairness depending on the current dispatch condition.

To preserve agent identity information, each free agent is associated with a learnable identity embedding. For each free agent, PRISM fuses three sources of information: agent context, global context,

and identity embedding. The global context is broadcast to all free agents and concatenated with the corresponding agent context and identity embedding. The concatenated vector is then projected through a nonlinear transformation to obtain an initial preference representation:

$$\mathbf{u}_i = \mathrm{ReLU}\left(\mathbf{W}_f[\mathbf{h}_i^a; \mathbf{h}^g; \mathbf{p}_i] + \mathbf{b}_f\right), \tag{4}$$

where $\mathbf{h}_i^a$ is the context of free agent i, $\mathbf{p}_i$ is its identity embedding, and $\mathbf{u}_i$ is the fused representation. This step makes the preference prediction both context-aware and agent-specific. The fused free agent representations then attend to node contexts through cross-attention. The node contexts are first linearly projected into hidden representations:

$$\mathbf{Z}^n = \mathbf{H}^n\mathbf{W}_n + \mathbf{b}_n, \quad \mathbf{Z}^n \in \mathbb{R}^{M\times d_h}, \tag{5}$$

where d_h is the hidden dimension. In the cross-attention layer, the fused free agent representations serve as queries, while the projected node representations serve as both keys and values:

$$\widetilde{\mathbf{U}} = \mathrm{LayerNorm}\left(\mathbf{U} + \mathrm{CrossAttn}(\mathbf{U}, \mathbf{Z}^n, \mathbf{Z}^n)\right), \tag{6}$$

where $\mathbf{U} \in \mathbb{R}^{N_f\times d_h}$ is obtained by stacking the fused representations of all free agents, and $\widetilde{\mathbf{U}}$ denotes the node-aware agent representations. This step helps the preference head understand which nodes are relevant to each free agent before deciding how much that agent should emphasize Resource Volume, Timeliness, or Fairness.

Next, PRISM applies self-attention among free agents. This step allows agents to exchange information with one another before their preferences are predicted:

$$\widehat{\mathbf{U}} = \mathrm{LayerNorm}\left(\widetilde{\mathbf{U}} + \mathrm{SelfAttn}(\widetilde{\mathbf{U}}, \widetilde{\mathbf{U}}, \widetilde{\mathbf{U}})\right). \tag{7}$$

The representation $\widehat{\mathbf{U}}$ contains both node-aware information and inter-agent coordination information. Without this interaction, preferences would be estimated independently for each agent, which may lead to uncoordinated objective tradeoffs.

Finally, each free agent representation is passed through an objective head to produce logits over the objective dimensions. A softmax is applied to obtain normalized preference weights:

$$\boldsymbol{\alpha}_i = \mathrm{softmax}\left(\mathrm{MLP}(\widehat{\mathbf{u}}_i)\right), \quad \boldsymbol{\alpha}_i \in \mathbb{R}^K, \tag{8}$$

where $\widehat{\mathbf{u}}_i$ is the i-th row of $\widehat{\mathbf{U}}$, denoting the final representation of free agent i, and K is the number of objectives. In our setting, there are three objectives ($K = 3$). The adaptive preference head controls how each agent values different objectives during node assignment. The resulting preference weights are used to combine the objective sub-scores into the final agent-node scores.

3.3 Semi-Autoregressive Decoder

The semi-autoregressive decoder generates a structured joint assignment for the currently free agents at each dispatch event. Unlike independent per-agent decision making, the decoder first computes a complete agent-node score matrix, and then sequentially selects assignments under feasibility constraints. An accumulated assignment context bias allows earlier assignments within the same event to adjust later agents' scores without score recomputation. This design combines the efficiency of parallel scoring with the flexibility of sequential assignment.

The decoder takes as input the global context $\mathbf{h}^g$, free agent contexts $\mathbf{H}^a$, node contexts $\mathbf{H}^n$, and the agent adaptive preference weights $\boldsymbol{\alpha}_i$ produced by the preference head. For each objective k, node contexts are projected into objective-specific key representations, and free agent contexts are projected into corresponding query representations. The global context is also projected and added as a shared dispatch bias to the agent queries. For objective k, the sub-score matrix is computed as:

$$\mathbf{S}^{(k)} = \frac{\mathbf{Q}^{(k)}(\mathbf{K}^{(k)})^\top}{\sqrt{d_h}}, \quad \mathbf{S}^{(k)} \in \mathbb{R}^{N_f\times M}, \tag{9}$$

where $\mathbf{Q}^{(k)} \in \mathbb{R}^{N_f\times d_h}$ denotes the objective-specific agent queries, $\mathbf{K}^{(k)} \in \mathbb{R}^{M\times d_h}$ denotes the objective-specific node keys. Each matrix $\mathbf{S}^{(k)}$ evaluates all free agent and node pairs under one objective, such as Volume, Timeliness, or Fairness. For each objective, normalization is performed

independently for each free agent over its currently feasible candidate nodes at each dispatch event. This ensures that the adaptive preference weights reflect the relative importance of different objectives within the agent's current decision space rather than being affected by numerical scale differences. The normalized objective scores are then combined using the agent adaptive preference weights. For each free agent i and node j, the preference weighted score is:

$$S^0_{i,j} = \sum_{k=1}^{K} \alpha_{i,k} \widehat{S}^{(k)}_{i,j}. \tag{10}$$

where $\widehat{S}^{(k)}$ denotes the normalized score matrix for objective k. This produces the base agent-node score matrix $\mathbf{S}^0 \in \mathbb{R}^{N_f \times M}$. The matrix is computed once before the assignment loop, which avoids repeatedly running the full neural decoder after each selected assignment.

After obtaining the base score matrix, the decoder constructs a feasibility mask from the candidate information. The initial candidate set for each agent includes all nodes. A feasibility mask is then applied to remove unreachable nodes, nodes that cannot be reached within the remaining episode time, and choices that would cause arrival-time collisions with other agents. The decoder then enters a constrained-first assignment loop. At each step, it selects the remaining free agent with the fewest feasible candidate nodes. This ordering gives priority to agents with more restricted choices. For the selected agent i, the decoder reads the corresponding row $\mathbf{S}^0_{i,:}$ from the base score matrix and adjusts it using an accumulated assignment context bias. This bias is initialized as zero and is updated whenever a node is selected. Intuitively, selected nodes leave a memory signal that affects subsequent agents' scores without recomputing the full score matrix.

Given the running assignment context $\mathbf{b} \in \mathbb{R}^{d_h}$, the decoder computes a node-wise bias for the current agent by comparing $\mathbf{b}$ with the specific objective node keys and weighting the result using the agent's preferences:

$$\Delta_{i,j} = \sum_{k=1}^{K} \alpha_{i,k} \frac{\mathbf{b}^\top \mathbf{k}^{(k)}_j}{\sqrt{d_h}}. \tag{11}$$

where $\mathbf{k}^{(k)}_j \in \mathbb{R}^{d_h}$ is the j-th row of $\mathbf{K}^{(k)}$, corresponding to node j under objective k. Thus, $\Delta_{i,j}$ is a scalar bias added to the base score of assigning agent i to node j. The current score for agent i is then:

$$\widetilde{S}_{i,j} = S^0_{i,j} + \Delta_{i,j}. \tag{12}$$

After applying feasibility mask, the decoder either samples a node during training or greedily selects the highest score node during evaluation. The selected assignment is added to the joint dispatch output. After a node j is selected, the decoder updates the running assignment context by accumulating a linear projection of the selected node representation:

$$\mathbf{b} \leftarrow \mathbf{b} + \mathbf{W}_b \mathbf{h}^n_j, \tag{13}$$

where $\mathbf{h}^n_j \in \mathbb{R}^d$ is the context representation of node j, and $\mathbf{W}_b \in \mathbb{R}^{d_h \times d}$ maps it into the decoder hidden space. When all free agents are processed, the decoder returns the joint dispatch assignment for the current event. The detailed training objective and optimization procedure are provided in Appendix B.

4 Experiment

4.1 Experiment Settings

We evaluate PRISM on four public spatio-temporal graph datasets: NYPD, EMS, PEMS08, and METR-LA. These datasets cover four representative urban service scenarios, including crime incidents, emergency medical services, traffic flow, and traffic speed. They differ in node scale, spatial geometry, graph connectivity, and temporal resource dynamics, enabling us to examine whether PRISM can generalize across heterogeneous spatio-temporal environments. These datasets are widely used in spatio-temporal data mining, forecasting, and urban computing tasks (Li et al., 2017; Guo et al., 2019; Shao et al., 2024). Detailed dataset statistics and implementation details of the experimental settings are provided in Appendix C.1.

We compare PRISM with representative TOP/MTOP baselines, including optimization-oriented baselines: PTOP (Shao et al., 2017), LAE (Shao et al., 2020), LERK-GA (Qin et al., 2020), and learning-based baselines: SDDQN (Schmoll & Schubert, 2021), PTR (He et al., 2022), DGAT (Zhang et al., 2022), SAIN (Strauss et al., 2022), SATOP (Strauß & Schubert, 2024). For single-agent baselines, we extend these methods by running multiple independent agents under the same event-driven protocol and feasibility mask rules as PRISM. We keep the original Volume objective for all baselines, but evaluate every method on Volume, Timeliness, and Fairness, with detailed metric definitions provided in Appendix C.2.

To further ensure a fair comparison under the multi-objective setting, we additionally construct multi-objective variants of representative baselines, including LERK-GA-MO, SDDQN-MO, PTR-MO, DGAT-MO, SAIN-MO, SATOP-MO, MAPPO-MO, and Hungarian-SC. These variants optimize the same three objectives as PRISM through a scalarized objective formulation, while preserving their original optimization or policy mechanisms. Detailed adaptation strategies are provided in Appendix C.3.

Additional experimental analyses, including efficiency analysis C.4, hyperparameter sensitivity analysis C.5 C.6, and case studies C.7, are provided in Appendix.

4.2 Comparison Experiment

Table 1 reports the comparison results on four datasets across Resource Volume, Fairness, and Timeliness. PRISM consistently achieves the best performance on all datasets, demonstrating strong robustness across heterogeneous spatio-temporal resource collection scenarios. Compared with the strongest baseline for each metric, PRISM improves Volume, Fairness, and Timeliness by 15.86%, 13.85%, and 26.64% on average, respectively. It can be observed that: (1) Although all baselines are designed to optimize volume only, some baselines exhibit severe strategy collapse on certain datasets, with Volume score dropping dramatically, highlighting their sensitivity to dataset characteristics. In contrast, PRISM maintains stable performance across all datasets, illustrating its reliability and generalization. (2) Even in cases where a baseline's Volume approaches that of PRISM, such as PTOP on the EMS dataset, its Fairness and Timeliness scores remain far below PRISM. This confirms that single-objective volume optimization induces hotspot-chasing behavior, neglecting service fairness and temporal responsiveness. PRISM, by contrast, achieves consistently high performance across all three objectives, indicating that its multi-objective design improves overall dispatch quality without sacrificing the primary collection goal. (3) Optimizing Volume does not necessarily yield strong Timeliness. Intuitively, pursuing high-volume resources would naturally lead agents to arrive near peak moments, the results show that methods with similar Volume can differ substantially in Timeliness. This indicates that Volume and Timeliness are related but not equivalent objectives, and Timeliness should be explicitly modeled.

To further validate the effectiveness of PRISM under a fair optimization setting, we compare it with representative multi-objective baselines, including multi-objective extensions of TOP/MTOP methods, MAPPO-MO, and Hungarian-SC. These baselines optimize the same three objectives as PRISM through their corresponding scalarized formulations. Table 2 reports the results. PRISM achieves the best performance on all four datasets and all three objectives. Compared with the strongest multi-objective baseline for each metric, PRISM improves Volume, Fairness, and Timeliness by 16.98%, 4.81%, and 6.19% on average, respectively. The improvements under the multi-objective setting demonstrate that the superiority of PRISM does not merely come from optimizing additional objectives. Even when baseline methods consider the same objective formulation, PRISM consistently achieves better performance, highlighting the effectiveness of its adaptive preference modeling and structured joint assignment mechanism. Specifically, MAPPO-MO uses the same multi-objective reward but generates agent decisions independently, while Hungarian-SC performs centralized assignment optimization based on immediate objectives. PRISM consistently outperforms both methods, indicating that explicit assignment coordination and adaptive decision mechanisms are important for achieving effective long-term multi-agent resource collection.

4.3 Ablation study

We evaluate the following ablated variants of PRISM:

Table 1: Comparison with representative TOP/MTOP baselines

Dataset	NYPD			EMS			PEMS08			METR-LA		
Metric	Volume	Fairness	Timeliness	Volume	Fairness	Timeliness	Volume	Fairness	Timeliness	Volume	Fairness	Timeliness
PTOP	3.85±0.72	3.36±0.19	7.32±0.76	4.44±0.66	3.21±0.17	8.85±0.77	6.91±2.65	2.01±0.23	12.27±0.61	13.59±1.40	3.20±0.28	13.93±1.35
LAE	2.61±0.31	4.04±0.08	7.69±0.57	1.56±0.27	5.51±0.34	9.27±0.74	6.50±2.46	2.49±0.18	16.76±0.51	13.79±1.48	4.06±0.24	15.66±0.90
LERK-GA	3.60±0.46	3.77±0.17	7.12±0.45	3.52±0.48	3.22±0.18	7.09±0.56	5.82±2.32	2.68±0.15	10.88±0.60	13.42±0.96	4.52±0.32	14.22±0.71
SDDQN	1.94±0.20	3.50±0.19	6.35±0.38	0.80±0.20	2.87±0.35	5.92±0.84	6.96±2.91	1.43±0.11	19.07±1.39	9.55±1.68	2.19±0.21	11.96±1.43
PTR	1.91±0.30	3.98±0.24	5.96±0.37	2.31±0.39	4.77±0.41	6.87±0.54	6.66±2.54	1.72±0.35	12.16±0.74	11.29±1.57	1.78±0.18	11.79±1.65
DGAT	1.35±0.16	1.94±0.17	5.99±0.72	0.77±0.16	5.07±0.45	5.28±0.67	7.71±3.61	1.56±0.24	17.12±1.39	14.05±0.89	2.88±0.38	14.34±0.99
SAIN	1.56±0.20	2.84±0.29	5.63±0.50	0.60±0.08	3.46±0.33	4.06±0.69	7.98±2.83	2.05±0.60	15.77±1.55	10.72±2.07	3.29±0.64	12.18±1.60
SATOP	2.30±0.36	2.25±0.11	7.04±0.58	0.67±0.13	2.27±0.21	5.90±0.49	6.49±2.90	1.88±0.13	18.26±1.12	12.20±1.42	2.57±0.21	14.00±1.02
PRISM	**4.69**±0.65	**4.79**±0.12	**11.75**±0.74	**4.47**±0.48	**5.69**±0.15	**11.78**±0.55	**9.56**±3.66	**3.01**±0.31	**20.02**±0.99	**17.02**±1.46	**5.48**±0.29	**19.06**±0.76

Table 2: Comparison with Multi-objective Baseline Variants

Dataset	NYPD			EMS			PEMS08			METR-LA		
Metric	Volume	Fairness	Timeliness	Volume	Fairness	Timeliness	Volume	Fairness	Timeliness	Volume	Fairness	Timeliness
LERK-GA-MO	3.40±0.53	4.15±0.18	6.89±0.62	3.17±0.38	3.64±0.15	6.73±0.39	5.40±2.17	2.95±0.30	10.39±0.75	12.58±1.06	4.85±0.34	13.45±0.87
SDDQN-MO	2.18±0.32	1.81±0.23	7.48±0.89	0.56±0.11	2.21±0.15	4.16±0.35	8.83±4.05	1.58±0.02	19.92±0.85	13.26±1.73	3.37±0.69	16.04±1.10
PTR-MO	2.18±0.34	3.50±0.09	6.05±0.51	2.11±0.41	4.24±0.29	6.59±0.61	7.63±2.14	2.10±0.44	16.87±0.59	14.72±0.55	4.19±0.15	15.44±0.43
DGAT-MO	2.35±0.41	4.68±0.21	5.13±0.37	3.26±0.39	2.68±0.23	5.86±0.39	8.78±4.08	1.98±0.18	18.80±1.17	11.58±0.52	4.51±0.17	12.55±0.68
SAIN-MO	1.52±0.28	2.49±0.22	5.40±0.73	0.38±0.10	3.41±0.31	3.65±0.65	8.45±3.26	1.74±0.23	16.76±1.74	12.13±0.42	4.34±0.33	12.99±0.46
SATOP-MO	1.80±0.26	1.98±0.08	6.21±0.50	1.79±0.19	2.04±0.12	8.94±0.37	5.36±2.06	1.91±0.27	17.11±0.93	9.57±1.91	4.52±0.32	11.80±0.63
MAPPO-MO	1.14±0.02	2.16±0.04	3.03±0.03	1.17±0.12	1.66±0.17	2.58±0.14	6.91±0.05	2.82±0.13	17.29±0.34	16.49±0.32	5.00±0.17	17.82±1.14
Hungarian-SC	3.93±0.29	4.45±0.12	10.71±0.32	3.14±0.48	5.15±0.20	10.95±0.44	7.14±2.06	2.74±0.21	18.87±0.73	15.02±0.50	5.25±0.19	14.70±0.49
PRISM	**4.69**±0.65	**4.79**±0.12	**11.75**±0.74	**4.47**±0.48	**5.69**±0.15	**11.78**±0.55	**9.56**±3.66	**3.01**±0.31	**20.02**±0.99	**17.02**±1.46	**5.48**±0.29	**19.06**±0.76

- **w/o event-driven formulation**: replaces event-driven formulation with step-based formulation, introducing no-op transitions between actual dispatch events.
- **w/o joint dispatch**: replaces structured joint dispatch with independent per-agent decisions and deterministic conflict handling.
- **w/o decomposed scorer**: replaces the separate decoder sub-score heads for the three objectives with a single scalar scoring head, while keeping the multi-objective reward unchanged.
- **w/o agent-specific preference**: uses one shared adaptive preference weight for all agents.
- **w/o adaptive preference**: replaces learned preferences with a fixed shared objective weight vector that assigns equal importance to all three objectives.
- **w/o context update**: removes the assignment context bias used to condition later decisions on previous assignments.

Table 3 reports the ablation results of PRISM. The full model achieves the best performance on all dataset-metric combinations, indicating that the proposed components jointly contribute to the final dispatch policy. Replacing the event-driven formulation with uniform-step decisions reduces the average performance by 11.0%, as the inserted no-op transitions introduce learning noise and complicate credit assignment, leading to performance degradation. Removing joint dispatch causes the largest degradation, with an average relative drop of 39.1% across all datasets and metrics. This confirms that MTOP cannot be effectively solved by independent per-agent decision making. Explicitly modeling joint dispatch is critical because multiple agents may become available at the same event, and their assignments are structurally coupled through shared candidate resources. Replacing the decoder sub-score heads with a single scalar scorer reduces the average performance by 22.2%, showing that a black-box score is insufficient to preserve the distinct semantics of Resource Volume, Timeliness, and Fairness. Removing agent-specific preferences causes a drop of 21.0%, indicating that per-agent adaptive preferences are necessary for coordinated specialization. Removing adaptive preference learning causes a comparable drop of 19.1%, especially on Fairness, which suggests that fixed shared objective weights cannot adapt to changing event contexts and local candidate constraints. Removing context update in the assignment loop leads to an average drop of 18.0%. This validates the role of the accumulated assignment context bias: although the base agent-node score matrix is computed in a batched manner, previous selections still need to influence later decisions during sequential assignment.

Table 3: Ablation study

Dataset	NYPD			EMS			PEMS08			METR-LA		
Metric	Volume	Fairness	Timeliness	Volume	Fairness	Timeliness	Volume	Fairness	Timeliness	Volume	Fairness	Timeliness
w/o event-driven formulation	4.12±0.50	4.34±0.11	10.79±0.73	2.81±0.41	5.00±0.13	8.72±0.46	8.53±3.22	2.82±0.11	19.43±0.61	16.40±1.33	5.46±0.21	18.44±0.63
w/o joint dispatch	2.83±0.41	1.94±0.14	7.40±0.43	0.99±0.22	2.74±0.06	4.08±0.28	6.15±2.38	2.96±0.05	15.95±0.44	15.08±1.61	2.23±0.04	17.19±0.64
w/o decomposed scorer	3.66±0.54	1.97±0.11	9.86±0.42	3.32±0.14	4.26±0.17	9.86±0.52	8.07±3.14	2.26±0.11	19.52±0.85	16.01±1.39	2.92±0.12	17.77±0.80
w/o agent-specific preference	3.14±0.48	2.07±0.08	8.85±0.69	3.19±0.34	4.51±0.15	10.07±0.41	9.11±3.23	2.03±0.12	19.75±0.56	16.96±1.80	3.72±0.21	18.63±0.76
w/o adaptive preference	3.77±0.46	1.81±0.08	10.65±0.52	3.01±0.40	4.18±0.10	10.16±0.71	7.94±2.87	2.42±0.06	18.98±0.64	16.87±1.37	2.41±0.23	19.05±0.69
w/o context update	3.47±0.44	3.74±0.18	8.13±0.51	2.34±0.38	4.57±0.20	7.45±0.35	9.35±3.80	2.39±0.38	19.27±1.37	16.97±1.45	5.27±0.18	18.67±0.81
full model	**4.69**±0.65	**4.79**±0.12	**11.75**±0.74	**4.47**±0.48	**5.69**±0.15	**11.78**±0.55	**9.56**±3.66	**3.01**±0.31	**20.02**±0.99	**17.02**±1.46	**5.48**±0.29	**19.06**±0.76

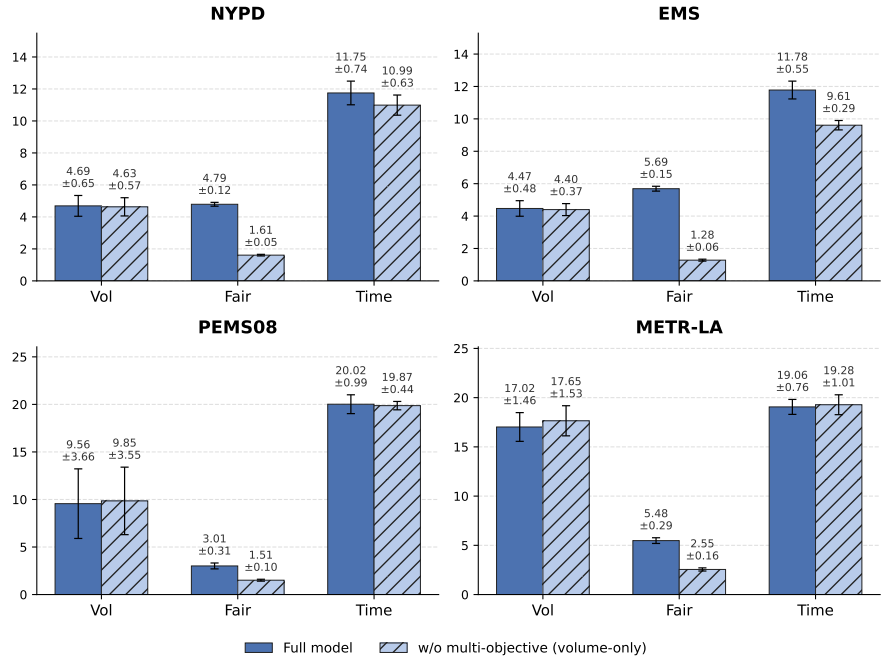


Figure 2: Result of the volume-only variant.

We additionally evaluate a volume-only variant, where both the reward and decoder scoring are reduced to a single Volume objective. As the result shown in Fig. 2, this variant can obtain comparable or slightly higher Volume, because it directly optimizes the Volume objective. However, this comes with a substantial loss in the other objectives, especially Fairness. These results show that volume-only optimization tends to produce hotspot-chasing behavior and neglects long unvisited nodes. In contrast, the full PRISM model preserves strong Volume while achieving much better Fairness and Timeliness, validating the necessity of the multi-objective formulation.

5 Conclusion

We proposed PRISM for the Multi-agent Traveling Officer Problem, formulated as an event-driven asynchronous multi-agent resource collection problem on dynamic spatio-temporal graphs. PRISM extends conventional volume-oriented settings to a multi-objective formulation with Resource Volume, Timeliness, and Fairness. It combines a state-aware context encoder, an adaptive preference head, and a semi-autoregressive decoder to generate structured joint assignments for currently available agents. By learning agent-specific objective preferences and using objective-aligned scoring, PRISM enables agents to dynamically specialize according to local state and global team context. The semi-autoregressive decoder further balances efficiency and coordination by computing agent-node scores in a batched manner while using assignment context updates to capture inter-assignment dependencies. Experiments on four public spatio-temporal datasets show that PRISM achieves the best performance across all datasets and metrics.

AI USE STATEMENT

In this work, we used generative AI tools only for improving the readability and language quality of the manuscript. We did not use generative AI tools for developing research ideas, designing methods, conducting experiments, implementing algorithms, analyzing results, or generating scientific content. All AI-assisted edits were carefully reviewed and verified by the authors. The authors take full responsibility for the final content of this paper.

REFERENCES

Entesar Alanzi and Mohamed El Bachir Menai. Solving the traveling salesman problem with machine learning: a review of recent advances and challenges. *Artificial Intelligence Review*, 58(9): 267, 2025.

Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: a methodological tour d'horizon. *European Journal of Operational Research*, 290(2): 405–421, 2021.

George B Dantzig and John H Ramser. The truck dispatching problem. *Management science*, 6(1): 80–91, 1959.

Yanyun Fu, Yiping Zeng, Deyong Wang, Hui Zhang, Yang Gao, and Yi Liu. Research on route optimization based on multiagent and genetic algorithm for community patrol. In *2020 International Conference on Urban Engineering and Management Science (ICUEMS)*, pp. 112–116. IEEE, 2020.

Wei Gong, Baoxian Zhang, and Cheng Li. Task assignment in mobile crowdsensing: Present and future directions. *IEEE network*, 32(4):100–107, 2018.

Shengnan Guo, Youfang Lin, Ning Feng, Chao Song, and Huaiyu Wan. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 922–929, 2019.

Ali Haghani, Qiang Tian, and Huijun Hu. Simulation model for real-time emergency vehicle dispatching and routing. *Transportation Research Record*, 1882(1):176–183, 2004.

Rongguang He, Xiao Xiao, Yufan Kang, Hongyu Zhao, and Wei Shao. Heterogeneous pointer network for travelling officer problem. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2022.

Wouter Kool, Herke Van Hoof, and Max Welling. Attention, learn to solve routing problems! *arXiv preprint arXiv:1803.08475*, 2018.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Eugene L Lawler. The traveling salesman problem: a guided tour of combinatorial optimization. *Wiley-Interscience Series in Discrete Mathematics*, 1985.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. *arXiv preprint arXiv:1707.01926*, 2017.

Harilaos N Psaraftis, Min Wen, and Christos A Kontovas. Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3–31, 2016.

Kyle K Qin, Wei Shao, Yongli Ren, Jeffrey Chan, and Flora D Salim. Solving multiple travelling officers problem with population-based optimization algorithms. *Neural Computing and Applications*, 32(16):12033–12059, 2020.

Ulrike Ritzinger, Jakob Puchinger, and Richard F Hartl. A survey on dynamic and stochastic vehicle routing problems. *International Journal of Production Research*, 54(1):215–231, 2016.

Sebastian Schmoll and Matthias Schubert. Semi-markov reinforcement learning for stochastic resource collection. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pp. 3349–3355, 2021.

Wei Shao, Flora D Salim, Tao Gu, Ngoc-Thanh Dinh, and Jeffrey Chan. Traveling officer problem: Managing car parking violations efficiently using sensor data. *IEEE Internet of Things Journal*, 5(2):802–810, 2017.

Wei Shao, Siyu Tan, Sichen Zhao, Kyle Kai Qin, Xinhong Hei, Jeffrey Chan, and Flora D Salim. Incorporating lstm auto-encoders in optimizations to solve parking officer patrolling problem. *ACM Transactions on Spatial Algorithms and Systems (TSAS)*, 6(3):1–21, 2020.

Wei Shao, Yufan Kang, Ziyan Peng, Xiao Xiao, Lei Wang, Yuhui Yang, and Flora D Salim. Stemo: Early spatio-temporal forecasting with multi-objective reinforcement learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2618–2627, 2024.

Niklas Strauß and Matthias Schubert. Spatial-aware deep reinforcement learning for the traveling officer problem. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*, pp. 869–877. SIAM, 2024.

Niklas Strauss, David Winkel, Max Berrendorf, and Matthias Schubert. Reinforcement learning for multi-agent stochastic resource collection. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 200–215. Springer, 2022.

Xiaocheng Tang, Zhiwei Qin, Fan Zhang, Zhaodong Wang, Zhe Xu, Yintai Ma, Hongtu Zhu, and Jieping Ye. A deep value-network based approach for multi-driver order dispatching. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1780–1790, 2019.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. *Advances in neural information processing systems*, 28, 2015.

Jingwei Wang, Qianyue Hao, Wenzhen Huang, Xiaochen Fan, Zhentao Tang, Bin Wang, Jianye Hao, and Yong Li. Dyps: Dynamic parameter sharing in multi-agent reinforcement learning for spatio-temporal resource allocation. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 3128–3139, 2024.

Yongyong Wei and Rong Zheng. Informative path planning for mobile sensing with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 864–873. IEEE, 2020.

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.

Andrea Zanella, Nicola Bui, Angelo Castellani, Lorenzo Vangelista, and Michele Zorzi. Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32, 2014.

Rongsheng Zhang, Cai Yang, and Xinxin Peng. Dynamic graph attention network for traveling officer problem. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2022.

Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):1–55, 2014.

A Problem Formulation

We formulate the Multi-agent Traveling Officer Problem as an event-driven asynchronous multi-agent resource collection problem on a dynamic spatio-temporal graph. Let $G = (V, E)$ denote the spatial network, where V is the set of nodes and E is the set of edges. Each node $v \in V$ is associated with a time-varying resource volume $x_v(t)$. A set of agents $\mathcal{A} = \{1, \dots, N\}$ moves over G within a finite episode horizon T. Each agent $i \in \mathcal{A}$ has a position $p_i(t) \in V$ at time t and may be either available for dispatch or traveling to a previously assigned destination. Different from step-based formulations, decisions in MTOP are made only at dispatch events. The l-th dispatch event is defined as

$$e_l = (t_l, F_l),$$

where t_l is the event time and $F_l \subseteq \mathcal{A}$ denotes the set of agents that become available at t_l. Multiple agents may become available at the same event. The event state s_l summarizes the current resource volumes, agent positions and statuses, recent resource histories, and node service histories.

At each event, the policy assigns available agents to feasible candidate resources. For each agent $i \in F_l$, let $\mathcal{C}_l^i \subseteq V$ denote its feasible candidate set. The event-level action is a structured joint assignment

$$a_l = \{a_l^i \mid i \in F_l\}, \qquad a_l^i \in \mathcal{C}_l^i \cup \{\varnothing\},$$

where $\varnothing$ is a fallback idle action used only when no feasible candidate is available. Since assignments at the same event are coupled, we represent a_l as a sequential assignment over agents in F_l. In this paper, agents are decoded in ascending order of their candidate set sizes, which means agents with fewer feasible choices are assigned earlier.

After agent i is assigned to destination a_l^i, it travels for a non-uniform duration

$$\tau_l^i = d(p_i(t_l), a_l^i),$$

where $d(\cdot, \cdot)$ is the travel time on the graph. The next dispatch event occurs when the next group of agents reaches their destinations. Therefore, MTOP naturally forms an event-driven asynchronous multi-agent SMDP.

We evaluate each event with a vector-valued reward $\mathbf{r}_l = \left(r_l^{\text{vol}}, r_l^{\text{time}}, r_l^{\text{fair}}\right)$, corresponding to Resource Volume, Timeliness, and Fairness. In training, the reward is scalarized as

$$r_l = \lambda_{\text{vol}} r_l^{\text{vol}} + \lambda_{\text{time}} r_l^{\text{time}} + \lambda_{\text{fair}} r_l^{\text{fair}}.$$

The goal is to learn a dispatch policy $\pi(a_l \mid s_l)$ that achieves strong expected performance over the multi-objective return

$$\mathbf{J}(\pi) = \mathbb{E}_\pi \left[\sum_{l=0}^{L} \gamma^{t_l} r_l \right],$$

where L is the number of dispatch events within the episode horizon, t_l is the elapsed time of the l-th dispatch event from the beginning of the episode. The term γ^{t_l} discounts rewards according to the actual elapsed time.

B Training

PRISM is trained with on-policy rollouts over event-driven episodes. Since consecutive dispatch events are separated by variable durations, we compute advantages using an SMDP-aware form of generalized advantage estimation. Let τ_t denote the duration between two consecutive dispatch events. The temporal-difference residual and advantage are computed as

$$\delta_t = r_t + \gamma^{\tau_t} V(s_{t+1}) - V(s_t), \tag{14}$$

$$A_t = \delta_t + \gamma^{\tau_t} \lambda A_{t+1}, \tag{15}$$

where s_t and s_{t+1} denote two consecutive event states, r_t is the reward obtained during the transition, $V(\cdot)$ is the value function, γ is the discount factor, and λ is the Generalized Advantage Estimation (GAE) parameter.

We optimize the policy with PPO. Since the semi-autoregressive decoder produces multiple assignment decisions within one event, PPO updates are applied at the assignment level: each selected

agent-node assignment has its own probability ratio, while all assignments in the same event share the global SMDP-aware advantage. A global critic estimates $V(s_t)$ from the encoded global context, and an entropy term is used to encourage exploration over feasible dispatch choices.

In addition to the PPO objective, PRISM uses two architecture auxiliary losses. First, we introduce a ranking loss for the decoder's sub-score heads. This loss aligns each sub-score head with its intended objective. For objective k, given a sampled node pair where one node has a better proxy signal than the other, the loss encourages the corresponding sub-score head to rank the better node higher:

$$\mathcal{L}_{\text{rank}} = \mathbb{E}\left[\max\left(0, \eta - \left(s^{(k)}_{\text{better}} - s^{(k)}_{\text{worse}}\right)\right)\right], \tag{16}$$

where η is the ranking margin. The proxy signals are current resource volume for Resource Volume, local temporal increase for Timeliness, and node age for Fairness. This loss helps each sub-score head preserve its intended objective semantics.

Second, we apply preference specialization regularization to the adaptive preference weights. Let $\boldsymbol{\alpha}_i \in \mathbb{R}^K$ denote the objective preference vector of free agent i. We combine an anti-collapse term and an inter-agent diversity term:

$$\mathcal{L}_{\text{pref}} = (1-\rho)\left(-\mathbb{E}_i \sum_{k=1}^{K} \log \alpha_{i,k}\right) - \rho \mathbb{E}_i \left\|\boldsymbol{\alpha}_i - \bar{\boldsymbol{\alpha}}\right\|_2^2, \tag{17}$$

where $\bar{\boldsymbol{\alpha}} = \frac{1}{N_f}\sum_{i=1}^{N_f} \boldsymbol{\alpha}_i$. The first term prevents degenerate preference vectors, while the second term encourages different agents to develop distinct objective emphases.

The final training objective is

$$\mathcal{L} = \mathcal{L}_{\text{policy}} + \beta_v \mathcal{L}_{\text{value}} - \beta_e \mathcal{H} + \beta_r \mathcal{L}_{\text{rank}} + \beta_p \mathcal{L}_{\text{pref}}, \tag{18}$$

where $\mathcal{L}_{\text{policy}}$ is the PPO policy loss, $\mathcal{L}_{\text{value}}$ is the value function loss, $\mathcal{H}$ denotes policy entropy, and β_v, β_e, β_r, and β_p are coefficients weights.

C Experiment

C.1 Implementation Details

Detailed dataset statistics are shown in Table 4. The environment advances in discrete time steps that correspond to the sample rate of each dataset. For example, for PEMS08, one time step corresponds to 5 minutes. Agent travel durations between nodes are also measured in time steps and are rounded up to the nearest integer. Specifically, the required travel time from node i to node j is calculated as $t_{ij} = \left\lceil \frac{d_{ij}}{v} \right\rceil$, where d_{ij} denotes the physical distance between the two nodes and v represents the moving speed of the agent. The ceiling operation ensures that the travel time is represented by an integer number of time steps, which is consistent with the discrete temporal resolution of the spatio-temporal datasets. For example, a travel time of 2.4 time steps is treated as 3 steps in the simulation. Each episode spans a fixed horizon of T time steps. Since different datasets provide distance adjacency matrices with different scales, directly applying the same moving speed would result in inconsistent mobility dynamics across datasets. Therefore, we set dataset-specific agent moving speeds according to the scale of the corresponding distance matrices. The detailed speed configurations for each dataset are provided in the released code.

At the beginning of an episode, each agent is initialized at a random starting position. After the episode ends, all agents return to their initial positions, and the next episode begins under the same protocol. For each dataset, we segment the original spatio-temporal sequence into a set of independent episodes for training and evaluation. The episodes are split into training and testing sets with a ratio of 8:2, where the former is used for model optimization and the latter is used to evaluate the generalization performance. All reported results are averaged over 10 independent runs, and the standard deviations are reported.

We use $N = 10$ agents and set the episode horizon to $T = 24$ time steps for all datasets. The reward weights $\lambda_{\text{vol}} = 1, \lambda_{\text{fair}} = 1.2, \lambda_{\text{time}} = 0.5$. In training, the entropy coefficient $\beta_e = 0.02$, value loss coefficient $\beta_v = 0.5$, objective ranking loss coefficient $\beta_r = 0.15$ and a margin $\eta = 0.1$,

Table 4: Statistics of datasets.

Datasets	Samples	Nodes	Sample rate	Domain
NYPD	4386	77	4 h	Crime
EMS	7992	145	1 h	Emergency
PEMS08	17856	170	5 min	Traffic Flow
METR-LA	34272	207	5 min	Traffic Speed

preference specialization regularization coefficient $\beta_p = 1.0$ and a diversity ratio $\rho = 0.98$. These hyperparameters were selected based on preliminary experiments and represent stable, effective settings without extensive manual tuning. For other specific parameters please refer to the code. All experiments are conducted on a PC with an Intel(R) Core(TM) i7-10700F CPU @ 2.90GHz, an NVIDIA GeForce RTX 3080 GPU.

C.2 METRICS

We evaluate all methods using three objectives: Resource Volume, Timeliness, and Fairness. Let $\mathcal{I}_l$ be the set of agents arriving during dispatch event l, N be the number of agents, T be the episode horizon, and let $\bar{x}_v(\tau)$ denote the max-normalized resource volume of node v at time τ.

Resource Volume measures the normalized resource demand captured upon arrival:

$$R_{\text{vol}}(l) = \frac{1}{N} \sum_{i \in \mathcal{I}_l} \bar{x}_{v_i}(\tau_i), \tag{19}$$

where v_i is the destination node of arriving agent i, and τ_i is its arrival time.

Timeliness measures whether agents arrive near recent local peaks. For each arriving agent i, let $p(i) = \max_{\tau \in [\tau_i - W, \tau_i]} \bar{x}_{v_i}(\tau)$, then

$$R_{\text{time}}(l) = \frac{1}{N} \sum_{i \in \mathcal{I}_l} \frac{\bar{x}_{v_i}(\tau_i)}{\max(p(i), \epsilon)}, \tag{20}$$

where $W = 3$ and ϵ is used for numerical stability.

Fairness measures the normalized time since the destination node was last visited:

$$R_{\text{fair}}(l) = \frac{1}{N} \sum_{i \in \mathcal{I}_l} \frac{\tau_i - \ell_{v_i}}{T}, \tag{21}$$

where ℓ_{v_i} is the previous visit time of node v_i. After computing the reward, ℓ_{v_i} is updated to τ_i.

For each episode, the reported metric is obtained by accumulating the corresponding dispatch event rewards over all events:

$$\mathcal{R}_m = \sum_{l \in \mathcal{L}_{\text{ep}}} R_m(l), \qquad m \in \{\text{vol}, \text{time}, \text{fair}\}, \tag{22}$$

where $\mathcal{L}_{\text{ep}}$ denotes all dispatch events within an episode.

C.3 BASELINES

The original TOP/MTOP baselines are mainly designed for volume-oriented decision making. To evaluate them in the multi-agent resource collection environment, we follow their original decision mechanisms while adapting the execution procedure to the shared dispatch environment of PRISM.

At each dispatch event, all available agents independently generate their decisions according to the original policies. Each agent receives its corresponding local observation and feasible candidate set, and the selected actions are executed under the same environmental constraints as PRISM, including destination feasibility and one-agent-one-destination assignment constraints. When multiple agents select the same destination, conflicts are resolved using a deterministic procedure to satisfy the assignment constraints. All baselines optimize their original objectives, where the single-objective

methods primarily maximize Resource Volume. Unlike PRISM, these baselines do not model explicit inter-agent coordination or assignment dependencies during decision generation.

To ensure a fair comparison, we extend representative baselines to the multi-objective setting by optimizing the same scalarized objective as PRISM:

$$R = \lambda_{\text{vol}} R_{\text{vol}} + \lambda_{\text{time}} R_{\text{time}} + \lambda_{\text{fair}} R_{\text{fair}}.$$

The following describes the adaptation details of each multi-objective baseline.

LERK-GA-MO. LERK-GA-MO extends the original LERK-GA from single-objective optimization to the multi-objective setting. Specifically, we replace the original genetic optimization objective with NSGA-II to optimize the three objectives considered in PRISM and retain the original solution representation and evolutionary operators. Since NSGA-II produces a set of Pareto-optimal assignments, the final assignment is selected according to the same scalarized objective used by PRISM for evaluation.

MAPPO-MO. MAPPO-MO adapts MAPPO (Yu et al., 2022) to the same multi-objective resource collection environment as PRISM. It uses the same state information, feasibility constraints, and scalarized reward formulation, while preserving the original MAPPO policy optimization framework. Unlike PRISM's structured joint assignment decoder with adaptive objective preferences, MAPPO-MO generates agent decisions through independent policy outputs and does not explicitly model assignment-level dependencies among agents.

Hungarian-SC. Hungarian-SC is a centralized assignment baseline that solves each dispatch event through combinatorial optimization. It constructs an assignment cost matrix by scalarizing the same three objectives as PRISM and applies the Hungarian algorithm (Kuhn, 1955) to obtain the assignment solution. All objective components are normalized before scalarization to avoid scale imbalance among different objectives. Unlike PRISM, Hungarian-SC optimizes assignment quality for the current dispatch event and does not learn adaptive policies or long-term decision strategies.

Other MO variants. For SDDQN, PTR, DGAT, SAIN, and SATOP, we extend their original implementations by replacing the single-objective Volume reward with the same scalarized multi-objective reward used by PRISM. The original policy architectures and decision mechanisms are preserved, while the training objective is modified to jointly consider Volume, Timeliness, and Fairness.

C.4 Efficiency Analysis

We compare the proposed semi-autoregressive decoder with a fully autoregressive decoder. The difference is that the fully autoregressive decoder recomputes node scores after each selected assignment, while the semi-autoregressive decoder computes the full agent-node base score matrix once through a batched forward pass. It then resolves assignments sequentially using feasibility masks and a lightweight assignment context bias. This avoids repeated neural score recomputation while still allowing previous assignments to affect later decisions.

Table 5 reports the average training time per episode in seconds. The semi-autoregressive decoder yields an 18.5% reduction in per-episode training time across all datasets. Since the model generally converges within about 20,000 episodes, the semi-autoregressive decoder saves roughly 1.67-3.83 hours of training time depending on the dataset, with an average saving of about 3.14 hours. Notably, even for the largest dataset in terms of graph scale, METR-LA, PRISM completes training in 15 hours. These results demonstrate that PRISM not only improves decoding efficiency but also enables fast training and deployment across diverse spatio-temporal environments, making it practical for large-scale real-world scenarios.

C.5 Effect of the Number of Agents

We evaluate PRISM with different numbers of agents, $N = 10, 15, 20, 25$, with the results presented in Table 6. Since PRISM aims to optimize the overall utility of a multi-agent team, total team rewards are used as the primary evaluation metric. The average reward per agent is additionally reported to analyze whether the performance gain comes from effective coordination rather than merely increasing the number of agents. Although average rewards per agent generally decrease

Table 5: Average per-episode training time (in seconds) for different decoder designs on four datasets.

Dataset	NYPD	EMS	PEMS08	METR-LA
autoregressive	2.29	2.72	3.15	3.06
semi-autoregressive	1.74	2.42	2.46	2.54

Table 6: Effect of the number of agents on PRISM. Each entry reports the total team reward, with the average reward per agent shown in parentheses: total (average).

Dataset	Agents	Volume	Fairness	Timeliness
NYPD	$N = 10$	46.89 (4.69)	47.88 (4.79)	117.50 (11.75)
	$N = 15$	49.22 (3.28)	50.52 (3.37)	134.60 (8.97)
	$N = 20$	62.51 (3.13)	61.22 (3.06)	174.27 (8.71)
	$N = 25$	70.55 (2.82)	69.87 (2.79)	210.56 (8.42)
EMS	$N = 10$	44.74 (4.47)	56.91 (5.69)	117.80 (11.78)
	$N = 15$	46.08 (3.07)	76.23 (5.08)	127.47 (8.50)
	$N = 20$	47.76 (2.39)	81.76 (4.09)	157.61 (7.88)
	$N = 25$	48.59 (1.94)	84.13 (3.37)	178.19 (7.13)
PEMS08	$N = 10$	95.58 (9.56)	30.09 (3.01)	200.22 (20.02)
	$N = 15$	137.06 (9.14)	35.44 (2.36)	292.23 (19.48)
	$N = 20$	161.64 (8.08)	50.11 (2.51)	368.25 (18.41)
	$N = 25$	176.22 (7.05)	62.67 (2.51)	413.84 (16.55)
METR-LA	$N = 10$	170.23 (17.02)	54.81 (5.48)	190.68 (19.07)
	$N = 15$	259.35 (17.29)	75.54 (5.04)	288.37 (19.22)
	$N = 20$	326.95 (16.35)	86.81 (4.34)	362.18 (18.11)
	$N = 25$	373.81 (14.95)	93.04 (3.72)	408.25 (16.33)

as the fleet size increases, this phenomenon is expected because the number of high-value service opportunities is limited by the underlying spatio-temporal distribution. With more agents competing for available opportunities, the marginal utility of each additional agent naturally decreases, which does not necessarily indicate degraded coordination performance.

As shown in Table 6, increasing the number of agents consistently improves the total rewards of all three objectives across different datasets, indicating that PRISM can effectively exploit additional service capacity while maintaining stable coordination under increasing fleet sizes. However, the improvement in total rewards gradually becomes saturated as the fleet size increases, while the average reward per agent decreases correspondingly. This indicates diminishing marginal gains when sufficient agents are available. For example, on EMS, increasing the number of agents from $N = 15$ to $N = 25$ only slightly improves the total Volume reward (46.08 to 48.59), while Fairness and Timeliness continue to increase notably (76.23 to 84.13 and 127.47 to 178.19, respectively). This indicates that when volume opportunities become saturated, PRISM can adaptively allocate additional agents toward improving other operational objectives rather than repeatedly pursuing limited volume gains. These results demonstrate that the adaptive preference mechanism enables PRISM to maintain effective multi-objective coordination under different agent budgets.

C.6 Effect of Multi-objective Preference Weights

To investigate the influence of global objective preferences on PRISM, we evaluate different settings of the preference weights λ_{vol}, λ_{fair}, and λ_{time}, which correspond to the weights assigned to Volume, Fairness, and Timeliness, respectively. In the main experiments, we use $\lambda_{\text{vol}} : \lambda_{\text{fair}} : \lambda_{\text{time}} = 1.0 : 1.2 : 0.5$. Here, we further compare three additional configurations: $p2 = (1.0 : 1.0 : 1.0)$, $p3 = (1.0 : 2.0 : 0.5)$, and $p4 = (1.0 : 0.5 : 0.5)$. The preference weights are used both as team-level objective weights signals in the State-Aware Context Encoder and as coefficients for computing the multi-objective reward.

Table 7: Effect of different multi-objective preference weights on PRISM.

Preference	NYPD			EMS			PEMS08			METR-LA		
	Vol.	Fair.	Time.	Vol.	Fair.	Time.	Vol.	Fair.	Time.	Vol.	Fair.	Time.
p1 (1.0:1.2:0.5)	4.69	4.79	11.75	4.47	5.69	11.78	9.56	3.01	20.02	17.02	5.48	19.06
p2 (1.0:1.0:1.0)	4.41	4.23	12.43	4.11	5.36	12.17	9.28	2.81	22.75	16.64	4.79	21.67
p3 (1.0:2.0:0.5)	4.20	5.26	10.99	3.06	6.17	10.10	8.95	3.78	18.94	15.58	6.07	18.48
p4 (1.0:0.5:0.5)	4.86	4.18	11.82	4.78	5.11	10.59	10.58	2.41	20.24	19.37	4.43	20.16

As shown in Table 7, different global preference settings lead to distinct behavioral tendencies of the agent team, demonstrating that PRISM can effectively adjust the trade-off among multiple objectives. The default setting $p1 = (1.0 : 1.2 : 0.5)$ provides a balanced preference among Volume, Fairness, and Timeliness, achieving stable performance across different datasets. When increasing the relative importance of Timeliness in $p2 = (1.0 : 1.0 : 1.0)$, PRISM tends to improve Timeliness rewards, especially on datasets with more dynamic spatial-temporal variations, while slightly reducing the emphasis on other objectives. Similarly, assigning a larger weight to Fairness in $p3 = (1.0 : 2.0 : 0.5)$ encourages more equitable resource allocation, resulting in higher Fairness performance but with reduced Volume and Timeliness rewards. In contrast, decreasing the Fairness weight in $p4 = (1.0 : 0.5 : 0.5)$ shifts the policy toward resource utilization, leading to higher Volume rewards while sacrificing part of the Fairness objective.

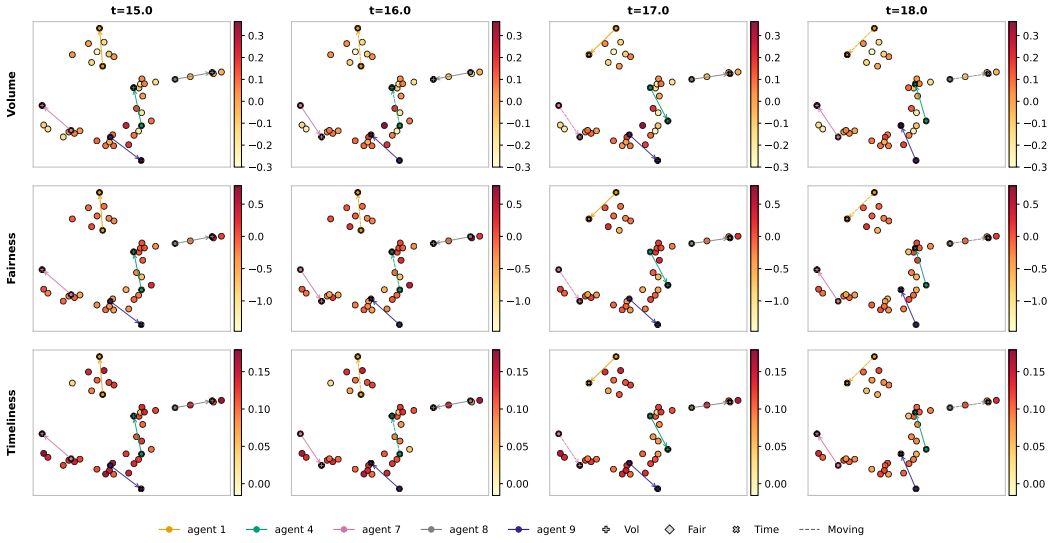
These results demonstrate that the global preference weights effectively control the objective trade-offs of PRISM. Instead of enforcing a fixed optimization priority, PRISM can adapt its team-level behavior according to different objective preferences while maintaining coordinated multi-objective decision making.

C.7 Case Study

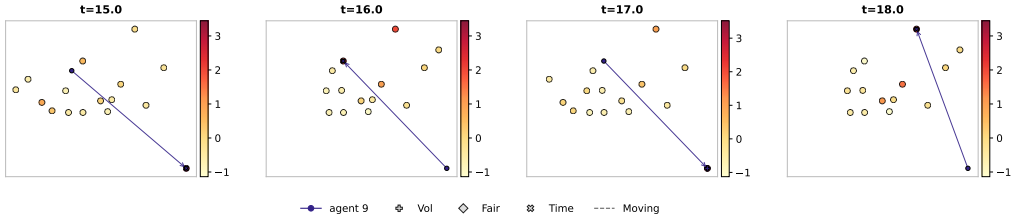
To investigate how PRISM performs adaptive multi-objective decision making, we visualize a representative dispatch episode on the METR-LA dataset. Figure 3a illustrates the joint assignments of multiple agents over four consecutive dispatch events. The node colors represent the raw objective-specific scores generated by the preference head. At each event, currently available agents are reassigned while agents in transit continue their previous routes. Different agents exhibit distinct dominant objectives under the same environment. For example, at $t = 15$, agent 1 mainly emphasizes Fairness, agents 4, 7, and 8 prioritize Volume, while agent 9 focuses on Timeliness. These preferences dynamically change with the evolving resource distribution and agent states. Meanwhile, selected destinations remain distinct at every dispatch event, demonstrating that adaptive preferences are integrated with the joint assignment mechanism to satisfy feasibility constraints.

While Figure 3a provides a multi-agent overview, the subsequent analysis focuses on an individual agent. This is because the final decision scores become agent-dependent after incorporating agent-specific preferences and assignment-context bias updates, preventing a unified visualization of final scores across multiple agents. To further examine the decision process of an individual agent, we analyze agent 9 in detail. Figure 3b presents the normalized objective-specific scores over its feasible candidate nodes at each selection moment. The preferred objective changes dynamically across decision events: Timeliness dominates at $t = 15$, $t = 16$, and $t = 18$, while Volume becomes dominant at $t = 17$. In each case, the selected node achieves the highest preference-weighted score among the feasible candidates. This indicates that the learned preference is not a fixed agent characteristic, but adapts according to the current dispatch condition.

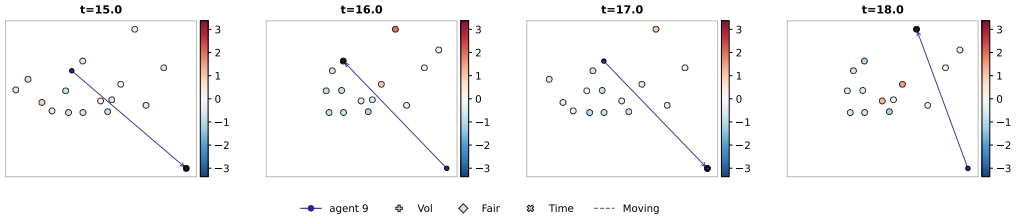
The contribution analysis in Figure 3c further explains how the adaptive preference mechanism affects the final assignment decision. After objective score normalization, the contribution of each objective is determined by both its normalized score and the corresponding preference weight. The results show that the objective with the highest preference consistently provides the dominant contribution to the final score. For instance, at $t = 15$, although the selected node receives competitive scores from all objectives, agent 9 assigns a preference weight of 0.978 to Timeliness, making Timeliness the primary contributor to the final decision. At $t = 17$, the preference shifts toward Volume, causing Volume to dominate the assignment score. Therefore, PRISM does not simply select nodes



(a)



(b)



(c)

Figure 3: Case study of adaptive multi-objective decision making in PRISM. (a) Multi-agent dispatch coordination under dynamic preferences. (b) Normalized objective-specific scores for an individual agent over feasible candidates. (c) Preference-weighted objective contributions explaining the final assignment decisions.

with universally high objective values; instead, it dynamically adjusts the importance of different objectives according to the current agent state and environment.

Overall, the case study demonstrates that PRISM achieves adaptive coordination at both system and agent levels. The joint assignment mechanism enables coordinated multi-agent dispatch, while objective normalization and preference weighting provide an interpretable mechanism for balancing different operational objectives under dynamic environments.