Transactions on Computational and Scientific Methods | Vo. 4, No. 8, 2024
ISSN: 2998-8780
https://pspress.org/index.php/tcsm
Pinnacle Science Press

Strategic Cache Allocation via Game-Aware Multi-Agent Reinforcement Learning

Yaokun Ren
Northeastern University, Seattle, USA
renyaokun0907@gmail.com

Abstract: This paper addresses the problems of low cache allocation efficiency and unstable strategies caused by multi-tenant resource competition in edge computing environments. A game-driven resource allocation mechanism based on multi-agent reinforcement learning is proposed. The mechanism consists of two core modules: the Game-aware Adaptive Policy Optimization (GAPO) framework and the State-aware Decentralized Agent Network (SADAN). GAPO introduces a local incentive adjustment function that guides agents to make more reasonable resource allocation decisions in dynamic competitive environments. It helps avoid convergence to suboptimal game equilibria. SADAN combines neighborhood state interaction with structured state encoding. It enables agents to capture system dynamics under partial observability and enhances policy coordination and learning efficiency. The cache resource allocation problem is modeled as a multi-agent game process. The proposed learning framework is applied to an edge caching system and evaluated using real-world datasets and a constructed simulation environment. Experimental results show that the proposed method outperforms existing approaches in key metrics such as cache hit rate, response delay, and policy convergence speed. Moreover, the method demonstrates strong robustness and stable system performance under varying conditions. These include multi-tenant scaling, reduced observation completeness, and changing resource constraints. The results effectively validate the adaptability and superiority of the proposed mechanism in edge cache resource allocation tasks.

Keywords: Edge caching, multi-agent systems, reinforcement learning, resource games

1. Introduction

With the rapid development of 5G, the Internet of Things, and big data applications, the number of network terminal devices is growing explosively[1]. Users are demanding services with lower latency and higher bandwidth. Traditional centralized cloud computing architectures struggle to meet the real-time response requirements of emerging applications due to physical distance and resource bottlenecks. As a result, edge computing has been proposed as an important complement to cloud computing. It moves computing and storage capabilities closer to the data source, significantly reducing communication latency and alleviating the load on core networks. In this architecture, edge caching plays a key role. By pre-storing popular content near users, effectively improves service response time and enhances user experience. However, edge nodes have limited resources and restricted caching space. How to allocate cache resources efficiently under resource constraints to meet diverse and dynamic user demands has become a critical issue in edge computing systems[2].

The problem of edge cache resource allocation is inherently competitive and non-cooperative in nature. It involves multiple users or service requests competing for limited caching resources. This problem depends not only on static information such as user preferences and content popularity, but also on dynamic factors including network topology, service latency, and bandwidth conditions[3]. Traditional static optimization methods often assume a stable or predictable environment, which makes them unsuitable for real-world scenarios where network conditions change frequently and participant strategies vary. In addition, cache conflicts among different services, uneven load distribution across edge nodes, and the spatiotemporal evolution of content popularity further increase the complexity of resource allocation. Therefore, relying solely on heuristic or centralized control strategies is insufficient to address the challenges of edge cache management in large-scale distributed environments[4].

In recent years, game theory has shown strong potential in modeling the interactions among multiple agents in distributed systems. It provides a theoretical foundation for analyzing strategic behaviors among users competing for cache resources. Game models help reveal equilibrium states under different strategy combinations, enabling the design of fair and efficient allocation mechanisms[5]. However, traditional game-theoretic approaches assume complete information and rational behavior, which are difficult to guarantee in edge computing environments. These environments often involve incomplete information and irrational decisions. Moreover, in highly dynamic settings, metrics such as cache hit rate and quality of service change with user behavior and network conditions. This leads to a massive and evolving strategy space that traditional game models cannot handle in real time[6].

To overcome these challenges, intelligent decision-making mechanisms have become a key research focus. Multi-agent reinforcement learning, which integrates game modeling and machine learning, has gained wide attention for solving complex decision-making problems. In edge computing scenarios, each edge node or user can be regarded as a learning agent. Through interaction with the environment, each agent continuously adjusts its strategy to maximize long-term rewards. This approach does not require prior knowledge of the global system. It learns adaptive allocation strategies through trial and error, making it especially suitable for distributed systems with high-dimensional state spaces, nonlinear feedback, and dynamic changes[7]. Combined with game-theoretic modeling, multi-agent reinforcement learning can address non-cooperative decision-making in competitive environments. It can also enhance system-wide efficiency and fairness through coordinated strategies. This makes the approach highly flexible and robust.

In conclusion, under the constraints of limited edge resources and distributed structures, adopting multi-agent reinforcement learning to drive game-based edge cache allocation has significant research value and practical relevance. This approach integrates strategic interactions among agents with feedback from the environment. It enhances the intelligence and automation of edge cache management under complex network conditions. Furthermore, it promotes the development of edge computing platforms toward higher efficiency, adaptability, and intelligence, laying a solid foundation for ubiquitous computing and real-time data services in the future.

2. Related work

2.1 Multi-agent reinforcement learning

As a natural extension of reinforcement learning, multi-agent reinforcement learning (MARL) has shown broad adaptability and potential in solving distributed decision-making problems in recent years. Under this paradigm, multiple agents learn strategies independently or collaboratively in a shared environment. They obtain feedback through continuous interaction and adjust their behavior to maximize long-term rewards for themselves or the entire system[8,9]. Compared with the single-agent setting, the key challenge in multi-agent systems lies in the non-stationarity of the environment. Each agent's behavior not only affects its feedback but also dynamically alters the learning environment of others. This significantly increases the complexity of the system's strategy space. To address this dynamic coupling, various methods have been proposed, including centralized training with decentralized execution, policy-sharing mechanisms, and joint action modeling. These approaches aim to reduce instability caused by policy updates among agents while preserving the distributed nature of the system[10].

In edge computing and network resource management scenarios, MARL is particularly suitable for modeling the behavioral evolution of multiple service nodes or users under shared resources. Edge nodes are typically widely distributed, resource-heterogeneous, and dynamically changing in operational state. Traditional centralized control methods often struggle to respond in real-time or face communication bottlenecks[11]. In contrast, multi-agent approaches allow each node or service to learn autonomously based on local information. This provides good scalability and robustness. Especially in problems like resource competition, cache allocation, and task offloading, there are naturally cooperative and conflicting relationships among entities. MARL is well suited to handle such complex game structures. With this method, efficient local optimal strategies can be achieved without relying on global information. Additionally, coordinated strategies can improve overall system performance while balancing efficiency and fairness[12].

It is worth noting that the performance of MARL in-game environments is closely related to algorithm design. Strategic interactions among agents may lead the system to fall into suboptimal equilibria or even cause strategy oscillations and convergence failures. Designing learning mechanisms that promote stable game convergence has become a key research focus. Recent studies have introduced attention mechanisms, value function decomposition, and policy projection techniques to improve model stability and convergence efficiency. At the same time, more efficient information-sharing frameworks are being developed under partially observable conditions to reduce learning bias caused by information asymmetry. These advances provide a solid theoretical and algorithmic foundation for the deeper application of MARL in edge caching, task scheduling, and intelligent communication. They also offer strong support for addressing game-driven resource allocation problems in this study[13].

2.2 Research on intelligent game

As the complexity of computing systems continues to grow, traditional game theory approaches that rely on precise modeling and rational behavior assumptions face increasing challenges in real-world applications[14]. To address the uncertainty of environments, bounded rationality of participants, and locality of information in multi-agent systems, the concept of intelligent game theory has emerged. Intelligent games integrate machine learning and reinforcement learning mechanisms[15]. This allows game agents to learn optimal strategies through continuous interaction, even in unknown or partially known environments, and to model and adapt to the behavior of others[16]. Compared with traditional static or complete-information dynamic games, intelligent games emphasize strategy evolution and learning ability. They better capture the complexity of strategic interdependence and information coupling in real systems. As a result, they are widely applied in fields such as network resource allocation, communication cooperation, and coordination of unmanned systems[17].

In edge computing and cache resource management scenarios, resource scarcity and diverse user demands lead to competitive game relationships in the system. Multiple edge nodes or service providers compete for limited cache space. Their strategic choices directly affect system performance and service quality. Traditional game methods are often based on static analysis or centralized control[18]. These approaches lack adaptability and real-time responsiveness, making them unsuitable for dynamic system requirements. Intelligent game methods, in contrast, leverage the learning ability of agents. They enable each participant to dynamically perceive environmental changes and adjust strategies based on local or historical information[19]. This supports strategy optimization and system coordination in complex interactions. Especially in non-cooperative environments, where services or users may have conflicting goals, intelligent games guide the system toward stable and efficient equilibrium through strategy evolution without requiring forced intervention. This gives the approach high practical value.

In recent years, intelligent game research has increasingly integrated advanced techniques such as reinforcement learning, multi-agent modeling, and graph-based optimization. These developments have led to significant improvements in modeling capability and computational efficiency. For example, reinforcement learning frameworks allow participants to explore unknown strategy spaces autonomously. Partially observable game models support reasonable decision-making under incomplete information[20]. Graph-based game methods leverage the topological relationships among nodes to introduce structure-aware strategies, enhancing game efficiency. These advances provide theoretical and technical foundations for building resource allocation mechanisms with learning ability, adaptability, and autonomy. In edge cache management, incorporating intelligent game concepts helps capture subtle strategy interactions among agents. It also offers effective paths for distributed, dynamic, and game-driven optimization in large-scale systems. This is of great significance for improving the overall intelligence of edge computing systems.

3. Method

This study proposes a game-driven edge cache resource allocation mechanism based on Multi-Agent Reinforcement Learning (MARL). It aims to address the challenge of dynamic coordination among multiple agents caused by competition for cache resources in edge computing environments. Compared with existing approaches, this method presents two key innovations. First, a learning framework is designed that integrates strategy evolution with resource games. Cache resource allocation is modeled as a repeated game among multiple agents. A local incentive mechanism is introduced to enable adaptive policy optimization. This enhances the system's distributed coordination capability. The framework is referred to as Game-aware Adaptive Policy Optimization (GAPO). Second, an agent architecture is developed that combines state awareness with neighborhood interaction. Each agent can capture strategy dependencies among agents using only local observations. This improves the model's stability and generalization in complex environments. The architecture is called a State-Aware Decentralized Agent Network (SADAN). These two innovations work together to advance edge cache allocation from rule-driven to learning-driven mechanisms. They lay a methodological foundation for building efficient, intelligent, and autonomous edge systems. The architecture of the overall model is illustrated in Figure 1.

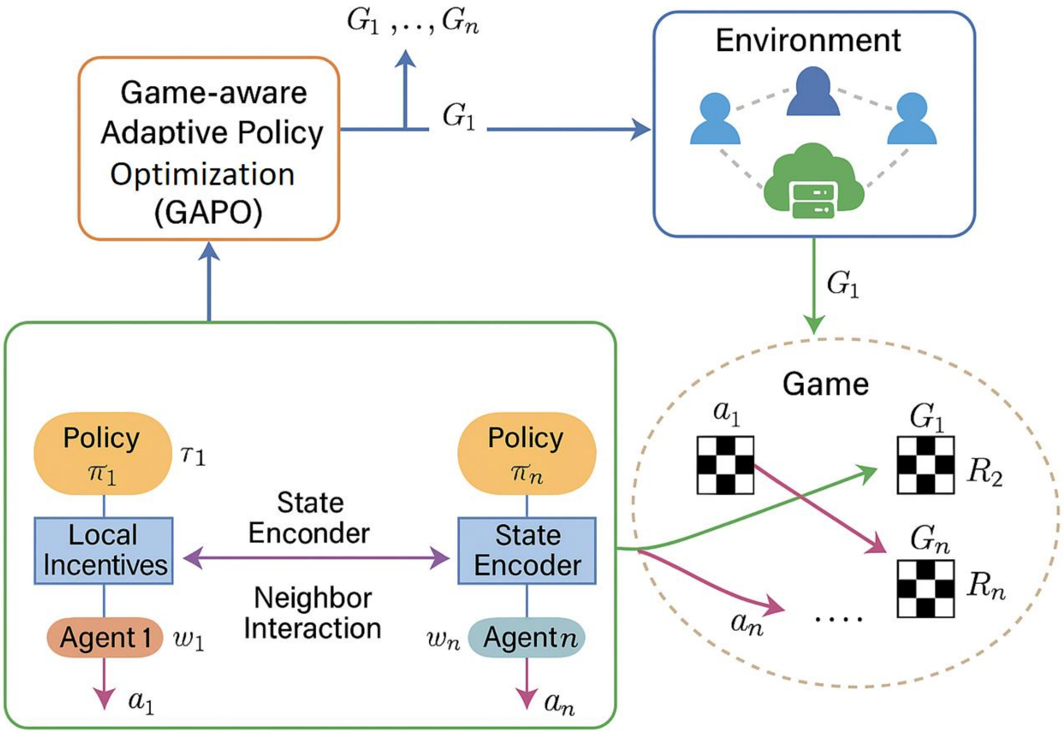

Figure 1. Overall model architecture diagram

3.1 Game-aware Adaptive Policy Optimization

To address the problem of cache resource competition among multiple agents in edge computing environments, this paper presents a game-aware multi-agent reinforcement learning approach called Game-aware Adaptive Policy Optimization (GAPO). This method is designed to capture the dynamic and competitive nature of distributed edge systems, where multiple autonomous agents—each representing an edge node or a service requester — must make independent yet interrelated decisions about resource allocation. The entire system is modeled as a multi-agent game environment in which each agent interacts with others through repeated strategic decisions. Within this framework, agents continuously adjust and evolve their policies in response to changing network conditions, user demands, and the behavior of neighboring agents. The objective of each agent is to maximize its long-term cache-related benefits while adapting to competition and resource constraints in a decentralized setting. GAPO integrates reinforcement learning with game-theoretic principles by introducing adaptive mechanisms that align individual agent incentives with system-level performance. This strategic learning process encourages more efficient and coordinated resource usage over time. The detailed architecture of the GAPO module is illustrated in Figure 2.

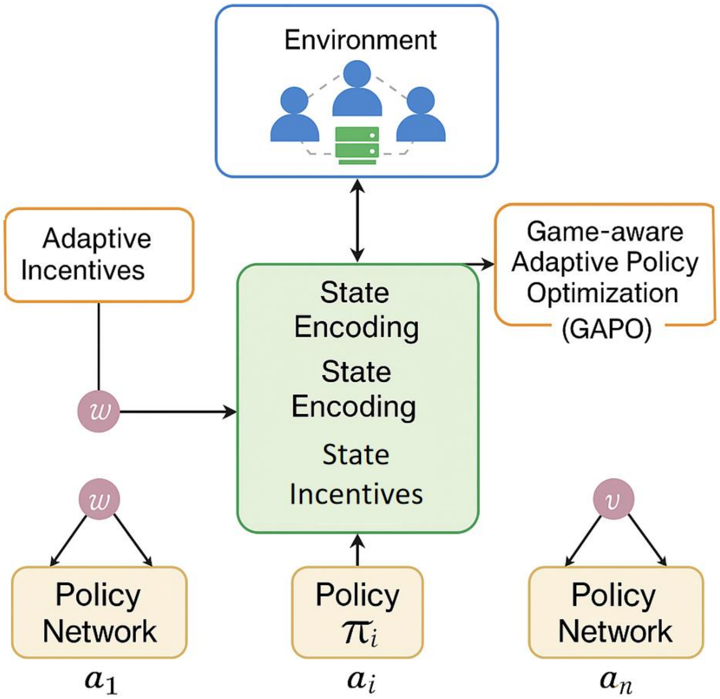

Figure 2. GAPO module architecture

The interaction between the agent and the environment is formalized as a partially observable Markov game (POMG), defined as a six-tuple:

$$G =< S, A, P, R, O, \gamma >$$

Among them, S represents the global state space, $A = \{A_1,..., A_n\}$ is the action space of each agent, P is the state transition probability function, $R = \{R_1,..., R_n\}$ is the local reward function of each agent, $O = \{O_1,..., O_n\}$ is the observation function, and $\gamma \in (0,1)$ is the discount factor.

In order to model the strategic interaction between agents, we introduce a graph-structured state perception mechanism that enables each agent to capture the behavior changes of other agents through neighborhood interactions. Let w_i represent the local observation input of agent i and N_i represent the set of its neighboring agents, then its state is expressed as:

$$h_i = Encoder(w_i, \{w_j\}_{j\in N_i})$$

After obtaining the state representation, the agent uses the policy network $\pi_{\theta_i}(a_i \mid h_i)$ to decide action a_i. The policy is optimized by maximizing the following expected return objective:

$$J(\theta_i) = E_{\tau\sim\pi_\theta}[\sum_{t=0}^{\infty} \gamma^t R_i^t]$$

Where τ represents the state-action sequence sampled from the policy trajectory. To ensure the stability of the policy update, the policy gradient is approximated as:

$$\nabla_{\theta_i} J(\theta_i) \approx E[\nabla_{\theta_i} \log \pi_{\theta_i}(a_i \mid h_i) \cdot \widehat{A}_i]$$

Where $\widehat{A}_i$ is the advantage function estimate, which measures the performance of the current action relative to the average behavior.

Furthermore, considering the heterogeneity and local incentive differences of different agents in the game process, GAPO introduces an adaptive incentive function φ_i to adjust the individual's strategic tendency and the direction of game equilibrium. This incentive function is combined with the original reward to form a new optimization goal:

$$\widehat{R}_i = R_i + \lambda \cdot \varphi_i(h_i, \{h_j\}_{j\in N_i})$$

λ is a regulatory factor used to balance the impact between environmental feedback and strategic games. This mechanism enables the agent to dynamically adjust its behavior strategy in both adversarial and collaborative game environments, thus improving the overall efficiency and stability of the system.

In summary, GAPO introduces graph structure neighborhood awareness at the state modeling level and integrates local game incentives at the policy optimization level, thus achieving policy autonomy and game awareness in resource allocation. This method has good scalability and generalization capabilities and is suitable for cache allocation scenarios with high requirements for policy dependence and resource heterogeneity in complex multi-agent environments. Through structured state encoding and policy decoupling optimization mechanisms, GAPO provides an intelligent solution with game awareness for edge cache resource management.

3.2 State-Aware Decentralized Agent Network

In a multi-agent edge computing system, environmental information is typically local, dynamic, and only partially observable due to the distributed nature of edge nodes and the inherent constraints in communication and sensing. To address these challenges and enhance the decision-making capabilities of individual agents, this paper introduces a state-aware decentralized agent network (SADAN). SADAN is specifically designed to improve the responsiveness of each agent to ongoing changes in its immediate surroundings as well as the evolving strategies of neighboring agents. The architecture allows each agent to operate autonomously, using its own locally observed data in combination with encoded state information from nearby agents. This enables agents to form a more comprehensive and context-sensitive understanding of their operational environment without requiring centralized coordination. By integrating neighborhood information into the local decision process, SADAN establishes a weakly coupled decentralized decision-making structure that balances autonomy and coordination. This design enhances the overall flexibility and scalability of the system, making it better suited for dynamic and resource-constrained edge computing environments. The detailed module architecture of SADAN is illustrated in Figure 3.

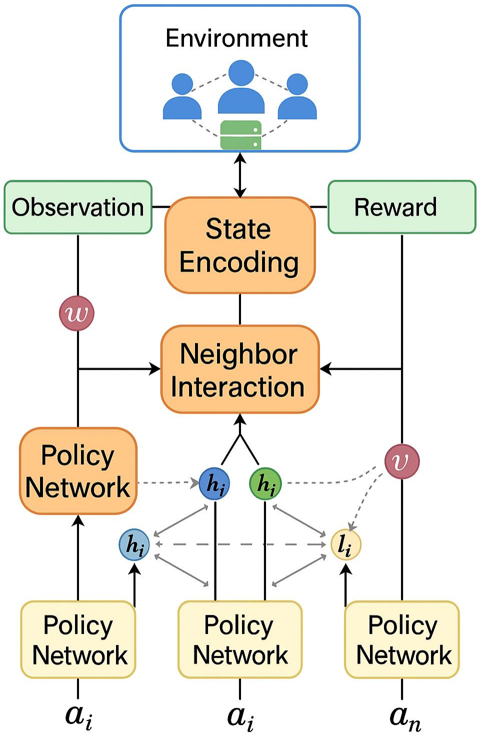

Figure 3. SADAN module architecture

Assume that the local observation of agent i at time step t is w_i^t, and its state representation is obtained through the state encoder:

$$h_i^t = f_{enc}(w_i^t, \{w_j^t\}_{j\in N_i})$$

Where N_i represents the set of neighbors that interact with agent i, and the encoding function $f_{enc}(\cdot)$ can be implemented based on the attention mechanism, graph neural network, or other structures.

After state encoding, the agent makes decisions based on the policy network π_{θ_i}, and its action sampling process is defined as:

$$a_i^t \sim \pi_{\theta_i}(a_i \mid h_i^t)$$

Each agent takes actions based on the local state, and the execution results affect the overall environment state and feedback on the local reward R_i^t. Since each agent strategy is learned independently, the overall strategy combination of the system is presented as a joint strategy space $\prod \pi_1 \times ... \times \pi_n$. Under this architecture, all agents optimize their strategy objective functions in parallel:

$$J(\theta_i) = E_\pi[\sum_{t=0}^{\infty} \gamma^t R_i^t]$$

Where γ is the discount factor, which is used to balance long-term and immediate returns.

In order to achieve the effectiveness of neighborhood interaction, SADAN introduces an adjacency-sensitive incentive propagation mechanism. By constructing an information interaction graph $G = (V, \varepsilon)$ between agents, each agent not only obtains local incentives but also combines the strategy evolution information of neighbor states to improve strategy coordination. This mechanism adjusts local incentive terms through structure-aware functions:

$$\widetilde{R}_i^t = R_i^t + \lambda \sum_{j \in N_i} \phi(h_i^t, h_j^t)$$

Where λ is the adjustment factor, and function $\phi(\cdot,\cdot)$ measures the similarity or coordination between neighborhood states, thereby encouraging local consistency of strategies.

In addition, in order to improve the stability of policy learning, the agent policy update uses the advantage function estimation method to perform policy gradient optimization. The update rule is as follows:

$$\nabla_{\theta_i} J(\theta_i) = E[\nabla_{\theta_i} \log \pi_{\theta_i}(a_i^t \mid h_i^t) \cdot \widehat{A}_i^t]$$

Where $\widehat{A}_i^t$ is the estimated value of the advantage function, which is usually approximated by the temporal difference (TD) method or the generalized advantage estimation (GAE). By tightly coupling the neighborhood interaction information with the strategy update process, the SADAN architecture not only realizes the autonomous learning and local optimization of the agent but also realizes the co-evolution of multi-agent strategies without the need for global information, providing structural support for game-driven edge cache resource management.

4. Experimental Results

4.1 Dataset

This study uses the EdgeDroid dataset as the foundation for experiments and validation. EdgeDroid is an open-source dataset specifically designed for edge computing and intelligent device behavior modeling. It contains real interaction data from a large number of mobile devices under various network environments and edge nodes. The dataset is highly representative and features rich data dimensions. It includes key indicators such as task offloading records, service request traces, latency feedback, bandwidth status, and cache hit rates. These features support the modeling needs of cache resource allocation and scheduling strategies in edge systems.

EdgeDroid collects data across several typical edge scenarios, including campus networks, urban hotspots, and home networks. In each scenario, end devices interact with nearby edge nodes, forming different computation request patterns and content access distributions. These data provide a realistic observation basis for simulating multi-agent game behaviors. They also offer high temporal resolution and spatial distribution, making them suitable for building state-action-reward structures used in reinforcement learning and game learning frameworks.

In addition, the EdgeDroid dataset includes configuration details of edge nodes, such as computational capacity, cache size, and task response time distributions. These parameters offer environmental constraints for resource allocation mechanisms. With appropriate preprocessing and feature extraction, the dataset can be effectively used to train state-aware multi-agent models. It also helps evaluate the generalization and stability of strategies in real-world scenarios. Its multidimensional and realistic characteristics make it an ideal choice for validating learning-based resource scheduling mechanisms in edge computing environments.

4.2 Experimental setup

To validate the effectiveness of the proposed method in practical scenarios, this study constructs an edge computing environment based on a simulation platform, using the real-world EdgeDroid dataset for modeling and training. The experimental platform consists of multiple simulated edge nodes and mobile users. The nodes have heterogeneous cache capacities and service processing capabilities. All agents perform parallel policy optimization under a unified training framework. Task requests, bandwidth states, and content access distributions in the environment are drawn from observational records in the dataset. The policy network adopts a shared structure and learns optimal resource allocation strategies in a discrete action space. A consistent discount factor and learning rate are used throughout training to ensure repeatability and fairness in the evaluation process.

During simulation, a unified hardware and software configuration is adopted. Several key parameters are set to control model complexity and learning efficiency. Table 1 presents the configuration details of the main experimental parameters, including the number of agents, number of edge nodes, cache capacity, observation dimensions, and policy update frequency. These parameters are determined based on the characteristics of the dataset and the task requirements. This ensures the realism of the experimental scenario and the controllability of algorithm execution. The detailed settings are shown in Table 1.

Table 1: Experimental Configuration Parameters

Parameter	**Value**
Number of Agents	10
Number of Edge Nodes	5
Cache Capacity per Node	100 units
Observation Dimension	64
Action Space	10 discrete actions
Discount Factor (γ)	0.95
Learning Rate	0.0001
Policy Update Frequency	Every 5 steps
Training Episodes	5000

4.3 Experimental Results

1) *Comparative experimental results*

This paper first gives the comparative experimental results, as shown in Table 2.

Table2: Comparative Results

Method	**Cache Hit Rate (%)**	**Avg. Latency (ms)**	**Convergence Steps**
Ours	91.2	42.7	3400
MAAC[20]	85.6	57.3	4900
QMIX[21]	83.1	60.5	5200
MAPPO[22]	88.0	51.2	4100

The comparative experimental results shown in Table 2 indicate that the proposed GAPO method outperforms existing multi-agent reinforcement learning models across several key performance metrics. In particular, GAPO achieves a cache hit rate of 91.2 percent, which is significantly higher than MAAC (85.6 percent), QMIX (83.1 percent), and MAPPO (88.0 percent). This demonstrates that the game-aware policy optimization mechanism is more effective in matching content demand between edge nodes and users. It leads to improved resource utilization efficiency. The result confirms that in dynamic game environments, combining local incentive adjustments with state-aware mechanisms offers clear advantages in cache resource allocation.

In terms of average task delay, GAPO records the lowest delay at only 42.7 milliseconds, reducing latency by at least 8.5 milliseconds compared to other models. This indicates that the proposed State-Aware Decentralized Agent Network can more effectively detect changes in neighborhood states. Allocating cache resources appropriately reduces task request transmission and waiting time. In real-world edge computing environments, low latency is critical. It meets the need for real-time responses from intelligent devices, especially in time-sensitive scenarios such as edge video services and intelligent transportation systems.

Regarding policy convergence efficiency, GAPO also shows faster training convergence. It stabilizes in only 3400 steps, compared to 5200 steps for QMIX and 4900 steps for MAAC. This improvement is due to the game-driven incentive adjustment mechanism introduced during policy optimization. It allows agents to consider not only environmental rewards but also the coordination of neighborhood behaviors. This reduces policy oscillation and enhances learning stability and efficiency.

In summary, the comparative experiments validate the proposed method's capability in resource allocation and policy adaptation under game-based environments. By introducing state-aware architectures and game-theoretic incentive modulation, GAPO outperforms mainstream multi-agent algorithms in several core metrics. These results demonstrate effective modeling and problem-solving from both system and algorithmic perspectives in edge cache management. They also lay the groundwork for deploying this method in real-world multi-tenant edge systems.

2) *Ablation Experiment Results*

This paper also further gives the results of the ablation experiment, and the experimental results are shown in Table 3.

Table 3: Ablation Experiment Results

Method	**Cache Hit Rate (%)**	**Avg. Latency (ms)**	**Convergence Steps**
Baseline	84.3	61.5	5400
+GAPO	88.1	49.6	4300
+SADAN	86.9	52.8	4700
Ours	91.2	42.7	3400

The ablation study results shown in Table 3 indicate that the two core components proposed in this study, GAPO and SADAN, both contribute significantly to the overall performance of the model. Compared with the baseline model, introducing only GAPO increases the cache hit rate from 84.3 percent to 88.1 percent. This shows that the game-aware policy optimization mechanism effectively guides agents to make cache decisions that better match content requests. The improvement is due to GAPO's dynamic adjustment of local incentives during multi-agent interactions, enabling more reasonable resource configuration in competitive environments and reducing content redundancy and resource conflicts.

The introduction of SADAN also brings a notable performance boost. The cache hit rate increases to 86.9 percent, and the average delay decreases from 61.5 milliseconds to 52.8 milliseconds. SADAN incorporates neighborhood state encoding into the agents' observation process. This allows agents to perceive not only their state but also infer the strategic tendencies of nearby nodes. As a result, the coordination among local strategies is enhanced. This state-aware mechanism addresses the policy fragmentation problem often seen in traditional decentralized methods and improves the regional rationality of resource allocation.

In terms of convergence efficiency, both GAPO and SADAN accelerate the training process. With GAPO, the number of convergence steps decreases from 5400 to 4300. With SADAN, convergence stabilizes at 4700 steps. This suggests that the two mechanisms improve learning effectiveness from different angles: GAPO through optimization target design and SADAN through state modeling. When combined, convergence further improves to 3400 steps. This shows that the two components complement each other and jointly accelerate policy stabilization during game-based learning. Overall, the complete method proposed in this paper outperforms any individual component across all three key metrics. This demonstrates the synergistic effect of the combined design of GAPO and SADAN in edge cache resource management. GAPO enhances policy adaptability through game-awareness, while SADAN improves environmental modeling through structure-aware mechanisms. Together, they significantly improve the intelligence of resource scheduling in multi-agent systems under dynamic and distributed conditions.

3) Comparative experiment on strategy convergence under different numbers of agents

This paper also gives the experimental results of strategy convergence comparison under different numbers of intelligent agents, as shown in Figure 4.

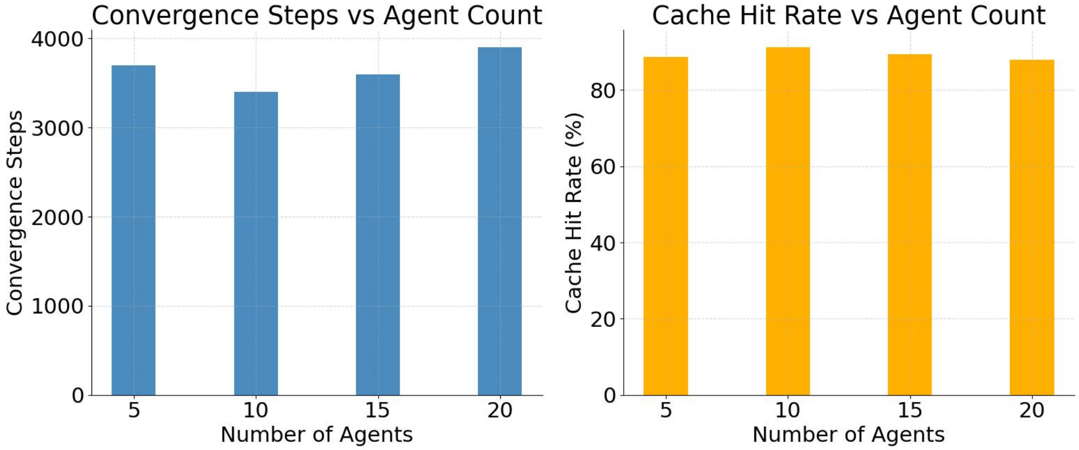

Figure 4. Comparative experiment on strategy convergence under different numbers of agents

As shown in Figure 4, the proposed method demonstrates consistent patterns in policy convergence efficiency and cache hit rate under different numbers of agents. When the number of agents increases from 5 to 10, the number of convergence steps decreases significantly. This indicates that moderately increasing the agent population enhances information exchange and learning efficiency across the system, which accelerates overall convergence. This result highlights the effectiveness of the proposed decentralized state-aware mechanism in multi-agent interactions. It also confirms the scalability of combining local observation with game-driven optimization.

As the number of agents continues to grow to 15 and 20, the convergence steps slightly increase. This suggests that in high-density agent environments, convergence may be affected by additional factors such as increased neighborhood state fluctuations and more complex game spaces. This observation supports the existence of a nonlinear relationship between agent population and learning dynamics in-game environments. It also emphasizes the importance of designing more efficient interaction mechanisms and incentive structures for large-scale systems.

In terms of cache hit rate, the model maintains a high overall performance, consistently exceeding 87 percent. This shows that the proposed strategy can stably identify and adapt to task content distributions even as the number of agents changes. When there are 10 agents, the cache hit rate reaches its highest value of 91.2 percent. This aligns with the best convergence efficiency, further confirming that the game-aware and statefusion mechanism achieves optimal synergy at moderate scales. Overall, the experimental results demonstrate the stability and adaptability of the proposed method during the scaling process of multi-agent systems. By building a framework that couples state awareness with strategic learning, the model achieves strong learning efficiency and maintains effective resource management across different levels of complexity in edge computing environments. This validates the robustness and practicality of the design in game-based scenarios.

4) Sensitivity analysis of multi-tenant competition intensity on game learning mechanism

This paper also gives a sensitivity analysis of the multi-tenant competition intensity to the game learning mechanism, and the experimental results are shown in Figure 5.

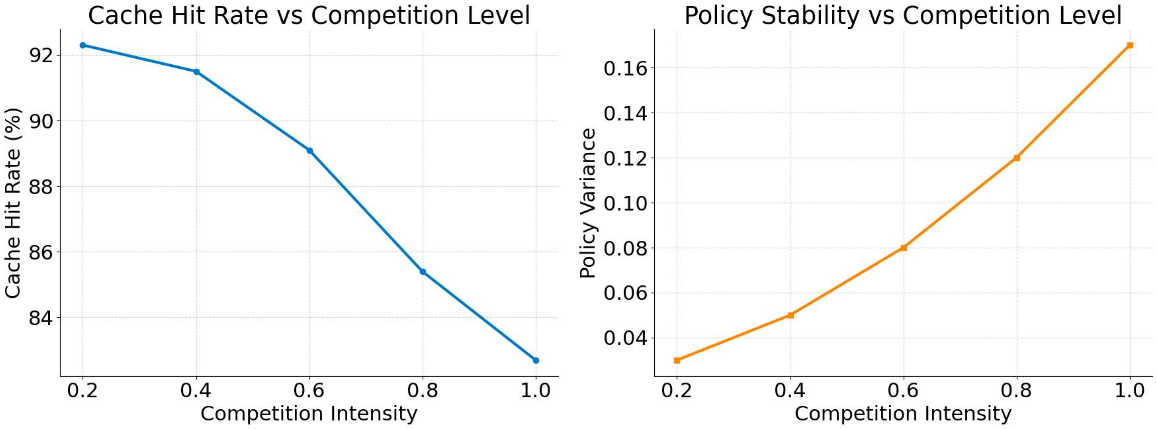

Figure 5. Sensitivity analysis of multi-tenant competition intensity on game learning mechanism

As shown in the experimental results of Figure 5, the cache hit rate decreases steadily as multi-tenant competition intensity increases. This trend indicates that when multiple tenants compete for limited edge cache resources, the overall caching performance of the system is significantly impacted. In particular, when the competition intensity exceeds 0.6, the decline in the hit rate becomes more rapid. This suggests that resource conflicts become more frequent, and agents face increasingly complex decisions in resource allocation. It highlights the challenges that high-competition game environments pose to learning mechanisms.

On the other hand, policy stability shows a clear upward trend with increasing competition intensity, as reflected in the continuous rise of the policy fluctuation coefficient. This result reveals that in highly competitive environments, strategy evolution among agents becomes more intense, leading to more frequent variations in decision-making. It suggests that learning mechanisms based solely on local feedback may struggle to converge to stable strategies under strong game dynamics. This emphasizes the importance of designing mechanisms to ensure stability in game-based learning.

Despite the observed fluctuations, the proposed mechanism maintains a relatively high cache hit rate and moderate policy stability under medium competition levels, such as between 0.4 and 0.6. This demonstrates a certain degree of robustness. The result confirms that the game-aware incentive mechanism introduced in this study can, within a reasonable range, alleviate resource conflicts and suppress strategy oscillations. This helps maintain baseline performance in multi-tenant environments. Overall, this experiment reveals the influence of multi-tenant competition on game-based learning mechanisms from two key dimensions. It provides empirical evidence for further optimizing state encoding structures, incentive function design, and policy coordination mechanisms. The findings also confirm that the proposed model retains a level of adaptability under resource-constrained conditions, offering strong support for practical deployment in multi-tenant edge computing environments.

5) *The impact of state observation completeness on multi-agent learning performance*

This paper also investigates the influence of state observation integrity on the performance of multi-agent learning within the proposed framework. In complex and dynamic edge computing environments, agents often operate under partial observability due to limitations in sensing capabilities, communication constraints, or privacy restrictions. As a result, the availability and completeness of state information can vary significantly across different agents and time steps. Understanding how these variations in observation quality affect the learning dynamics, coordination efficiency, and policy adaptation is essential for designing robust and scalable multi-agent systems. To explore this aspect, the study incorporates a series of controlled experiments that systematically adjust the level of state observation integrity and monitor its effects on the learning process. Figure 6 presents the corresponding results, which provide insight into the relationship between information completeness and agent behavior under game-driven resource allocation scenarios.

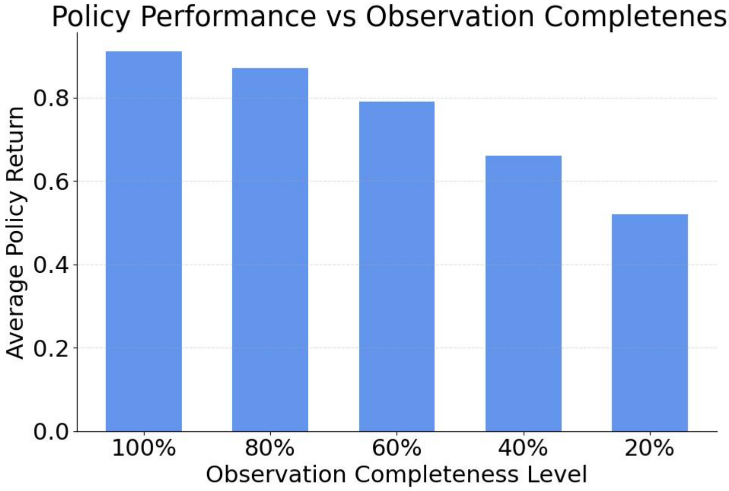

Figure 6. The impact of state observation completeness on multi-agent learning performance

Figure 6 shows the average policy return of the proposed method under different levels of state observation completeness. The figure indicates that as observation completeness decreases, the learning performance of agents declines. The average policy return drops from 0.91 under full observation to 0.52 under partial observation. This demonstrates that missing state information significantly weakens the agents' ability to perceive the environment, which in turn reduces the accuracy of policy decisions and resource allocation.

When observation completeness remains above 80 percent, the model performance remains relatively stable. The policy return stays at a high level. This indicates that the proposed state encoding and policy optimization mechanisms have a certain degree of robustness and can adapt to environments with minor information loss. However, when the observation level drops below 60 percent, the return decreases sharply. This suggests that missing information begins to disrupt neighborhood interactions and policy coordination. The policy network becomes less effective at capturing environmental dynamics, which undermines the overall performance of game-based optimization.

These results highlight the importance of introducing state-aware structures such as SADAN. Under incomplete observation, agents cannot accurately model the behavior of their neighbors. This increases the risk of local policy conflicts and resource waste. Enhancing state encoding and neighborhood information aggregation helps reduce decision errors caused by missing observations. It also improves system stability during the learning process in-game environments.

In conclusion, observation completeness has a critical impact on learning performance in multi-agent systems. In edge caching scenarios, incomplete state information directly leads to lower resource allocation efficiency.

The experiment further emphasizes the need to build structured state representations and robust decision-making mechanisms in multi-agent game settings. This provides a foundation for ensuring model stability in future real-world deployments.

5. Conclusion

This paper addresses the issue of resource competition in edge computing environments and proposes a game-driven cache resource allocation mechanism based on multi-agent reinforcement learning. The goal is to enhance intelligent decision-making under complex interaction scenarios. The mechanism integrates game modeling with reinforcement learning-based policy optimization. By introducing Game-aware Adaptive Policy Optimization (GAPO) and the State-aware Decentralized Agent Network (SADAN), agents are able to adaptively learn resource scheduling strategies in partially observable and multi-tenant competitive environments. This work provides a systematic exploration of the integration between multi-agent learning and edge resource management. It contributes to the research landscape of intelligent decision-making at the edge. Experimental results validate the effectiveness and robustness of the proposed method from multiple perspectives. The method outperforms existing mainstream models in cache hit rate, average response delay, and policy convergence speed. It also maintains strong stability under different conditions such as ablation settings, system scaling, and observation completeness perturbations. These results not only confirm the method's practical applicability but also highlight the impact of state modeling quality, game mechanism design, and local incentive structures on multi-agent learning performance. By leveraging the structural features of the environment and dynamic game interactions among agents, this study presents a learning framework that is both interpretable and adaptive for edge cache resource allocation. The proposed method is generalizable and extensible. It can be applied to various edge intelligence scenarios, including content delivery networks, intelligent transportation systems, industrial Internet of Things, and mobile cloud services. These applications often involve high task volumes, limited resources, and significant environmental uncertainty. They require efficient multi-agent collaborative learning mechanisms for resource scheduling and policy optimization. The method presented in this paper not only provides technical support for specific edge caching problems but also offers a modeling paradigm and algorithmic framework for broader distributed intelligent decision systems.

6. Future work

Future work may explore more generalizable agent architectures to improve policy transfer across different task distributions and network topologies. The integration of privacy-preserving mechanisms, asynchronous collaboration models, or federated game learning techniques could further enhance deployability and fairness in real-world multi-tenant environments. As computing continues to move closer to the edge and intelligent infrastructure matures, the findings of this study are expected to be applicable in large-scale edge systems. This will support the deployment and evolution of intelligent resource management technologies on emerging computing platforms.

References

[1] Zhou G, Tian W, Buyya R, et al. Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions[J]. Artificial Intelligence Review, 2024, 57(5): 124.

[2] Chen G, Qi J, Sun Y, et al. A collaborative scheduling method for cloud computing heterogeneous workflows based on deep reinforcement learning[J]. Future Generation Computer Systems, 2023, 141: 284-297.

[3] Song C, Han G, Zeng P. Cloud computing based demand response management using deep reinforcement learning[J]. IEEE transactions on cloud computing, 2021, 10(1): 72-81.

[4] Islam M T, Karunasekera S, Buyya R. Performance and cost-efficient spark job scheduling based on deep reinforcement learning in cloud computing environments[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 33(7): 1695-1710.

[5] Siddesha K, Jayaramaiah G V, Singh C. A novel deep reinforcement learning scheme for task scheduling in cloud computing[J]. Cluster computing, 2022, 25(6): 4171-4188.

[6] Sudhakar R V, Dastagiraiah C, Pattem S, et al. Multi-Objective Reinforcement Learning Based Algorithm for Dynamic Workflow Scheduling in Cloud Computing[J]. Indonesian Journal of Electrical Engineering and Informatics (IJEEI), 2024, 12(3): 640-649.

[7] Mangalampalli S, Karri G R, Kumar M, et al. DRLBTSA: Deep reinforcement learning based task-scheduling algorithm in cloud computing[J]. Multimedia tools and applications, 2024, 83(3): 8359-8387.

[8] Canese L, Cardarilli G C, Di Nunzio L, et al. Multi-agent reinforcement learning: A review of challenges and applications[J]. Applied Sciences, 2021, 11(11): 4948.

[9] Li T, Zhu K, Luong N C, et al. Applications of multi-agent reinforcement learning in future internet: A comprehensive survey[J]. IEEE Communications Surveys & Tutorials, 2022, 24(2): 1240-1279.

[10] Wen M, Kuba J, Lin R, et al. Multi-agent reinforcement learning is a sequence modeling problem[J]. Advances in Neural Information Processing Systems, 2022, 35: 16509-16521.

[11] Zhang K, Yang Z, Başar T. Multi-agent reinforcement learning: A selective overview of theories and algorithms[J]. Handbook of reinforcement learning and control, 2021: 321-384.

[12] Gu S, Kuba J G, Chen Y, et al. Safe multi-agent reinforcement learning for multi-robot control[J]. Artificial Intelligence, 2023, 319: 103905.

[13] Souchleris K, Sidiropoulos G K, Papakostas G A. Reinforcement learning in game industry—Review, prospects and challenges[J]. Applied Sciences, 2023, 13(4): 2443.

[14] Perolat J, De Vylder B, Hennes D, et al. Mastering the game of Stratego with model-free multiagent reinforcement learning[J]. Science, 2022, 378(6623): 990-996.

[15] Xu Z, Yu C, Fang F, et al. Language agents with reinforcement learning for strategic play in the werewolf game[J]. arXiv preprint arXiv:2310.18940, 2023.

[16] Zhang R, Zong Q, Zhang X, et al. Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2022, 34(10): 7900-7909.

[17] Bakhtin A, Wu D J, Lerer A, et al. Mastering the game of no-press diplomacy via human-regularized reinforcement learning and planning[J]. arXiv preprint arXiv:2210.05492, 2022.

[18] Qian T, Shao C, Li X, et al. Multi-agent deep reinforcement learning method for EV charging station game[J]. IEEE Transactions on Power Systems, 2021, 37(3): 1682-1694.

[19] Yuan M, Shan J, Mi K. Deep reinforcement learning based game-theoretic decision-making for autonomous vehicles[J]. IEEE Robotics and Automation Letters, 2021, 7(2): 818-825.

[20] Zhu L, Zhang S, Xie L. IUS-MAAC: Multi-Agent Attention-Critic for Task Offloading in Integrated User-Server Mobile Edge Computing[C]//2024 China Automation Congress (CAC). IEEE, 2024: 2019-2024.

[21] Rashid T, Farquhar G, Peng B, et al. Weighted qmix: Expanding monotonic value function factorisation for deep multi-agent reinforcement learning[J]. Advances in neural information processing systems, 2020, 33: 10199-10210.

[22] Lohse O, Pütz N, Hörmann K. Implementing an online scheduling approach for production with multi agent proximal policy optimization (MAPPO)[C]//Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems: IFIP WG 5.7 International Conference, APMS 2021, Nantes, France, September 5–9, 2021, Proceedings, Part V. Springer International Publishing, 2021: 586-595.