Proactive QoS-Aware Preemptive Resource Allocation in Mobile Edge Computing

Haowen Shi * and Yichen Zong

Posted Date: 23 March 2026

doi: 10.20944/preprints202603.1741.v1

Keywords: mobile edge computing; deep reinforcement learning; QoS prediction; proactive preemption; resource management

 doi:10.20944/preprints202603.1741.v1

Article

Proactive QoS-Aware Preemptive Resource Allocation in Mobile Edge Computing

Haowen Shi * and Yichen Zong

Sichuan University of Science and Engineering
* Correspondence: 2021448365721@stu.zuel.edu.cn

Abstract

Efficient Mobile Edge Computing (MEC) resource management is critical for diverse Quality of Service (QoS) demands, but traditional reactive methods and existing preemptive policies struggle in dynamic environments, causing suboptimal experiences. This paper proposes Proactive Adaptive Preemptive Allocation (PAPA), a novel framework for intelligent, forward-looking MEC resource management. PAPA features a QoS prediction module using lightweight sequence models to forecast short-term trends, assess risk, and trigger pre-warnings. Its core, the Proactive Preemptive Strategy Learning (APPL) module, employs a deep reinforcement learning (DRL) agent with a unique dual-layer reward. This includes a proactive penalty compelling anticipatory preemptive actions when predicted QoS enters a warning zone, differentiating it from reactive approaches. PAPA further enhances adaptability via meta-learning and dynamic priority mechanisms. Extensive simulations show PAPA consistently outperforms baselines, achieving superior throughput, reduced latency, and a significantly lower critical QoS violation rate than reactive DRL. Ablation studies confirm the impact of proactive penalty and meta-learning. PAPA demonstrates competitive energy efficiency and optimized preemption, affirming its robustness and practical viability in dynamic MEC environments.

Keywords: mobile edge computing; deep reinforcement learning; QoS prediction; proactive preemption; resource management

1. Introduction

The rapid proliferation of smart devices and data-intensive applications, such as augmented reality, autonomous driving, and industrial IoT, has spurred the evolution of Mobile Edge Computing (MEC) [1]. MEC extends cloud computing capabilities to the network edge, bringing computation and storage closer to end-users and IoT devices. This paradigm significantly reduces end-to-end latency, conserves backhaul bandwidth, and enhances user experience. However, the dynamic and resource-constrained nature of MEC environments poses significant challenges for efficient resource management. Ensuring diverse and often conflicting Quality of Service (QoS) requirements (e.g., low latency, high throughput, high reliability) for heterogeneous tasks, especially in the face of fluctuating network conditions, user mobility, and sudden task arrivals, remains a paramount concern [2]. Effective resource allocation and scheduling, particularly through preemptive mechanisms, are critical for maximizing system performance while guaranteeing QoS for high-priority tasks. Recent advancements in this domain include novel preemptive policies for minimizing Age of Information (AoI) in MEC [3] and generalizable Pareto-optimal offloading strategies leveraging reinforcement learning [4].

Traditional resource allocation strategies in MEC often fall short in highly dynamic scenarios. Many existing approaches are *reactive* [5], meaning they only initiate adjustments after QoS degradation or violations have already occurred. This inherent lag leads to suboptimal user experience, inefficient resource utilization, and potential failure to meet critical application deadlines. Furthermore, current preemptive policies frequently rely on static priority assignments or simplistic rules, which struggle to

 doi:10.20944/preprints202603.1741.v1

adapt to the intricate interplay of dynamic QoS demands, varying task criticalities, and fluctuating resource availability [6]. The core challenge lies in developing a strategy that can *proactively* anticipate potential QoS violations and execute *adaptive preemptive resource adjustments* before issues materialize, thereby ensuring critical task QoS while optimizing overall system efficiency and exhibiting strong generalization capabilities across diverse MEC deployments. This aligns with the need for proactive constrained policy optimization with preemptive penalties [7] and generalizable solutions for efficient resource management [4].

To address these challenges, we propose a novel framework named **Proactive Adaptive Preemptive Allocation (PAPA)**. PAPA is designed as an intelligent, efficient, and forward-looking resource management solution for MEC, integrating advanced predictive analytics, deep reinforcement learning (DRL) with a unique proactive penalty mechanism, and meta-learning capabilities. Our framework comprises a QoS prediction and state awareness module that leverages lightweight sequence models (e.g., LSTM or Transformer-based networks) to forecast future QoS trends and assess risk. A crucial component is the Proactive Preemptive Strategy Learning (APPL) module, where a DRL agent is trained with a dual-layer reward mechanism. This mechanism incorporates a *proactive penalty* that triggers when predicted QoS indicators enter a "warning" zone, compelling the agent to act preemptively rather than reactively. This approach is inspired by recent research into proactive constrained policy optimization with preemptive penalties, aiming to guide agents towards anticipatory actions [7]. PAPA's decision space encompasses when and how to preempt, what tasks to target, and how to optimally reallocate released resources. Furthermore, PAPA enhances its adaptability and generalization through meta-learning, enabling rapid adjustment to new MEC scenarios, and employs dynamic priority mechanisms that evolve with task criticality and predicted QoS risk. The emphasis on generalization is crucial for diverse MEC deployments, as explored in works on generalizable offloading strategies [4].

For experimental validation, we conducted extensive simulations using a custom Python-based MEC simulator. This simulator accurately models a multi-server MEC environment, encompassing heterogeneous wireless channels, diverse user mobility patterns, and varied task workloads. Our simulation environment includes 5 edge servers with distinct computational capacities and communication bandwidths, supporting 20-50 mobile users randomly distributed across a geographical area. Tasks are modeled with stochastic arrival rates, varying computation demands, data sizes, and specific QoS requirements (e.g., real-time streaming tasks are delay-sensitive, while file uploads are throughput-sensitive). We compare PAPA against several prominent baselines, including a Greedy Scheduler, a Max-Min Fair Scheduler, a Reactive DRL Scheduler, and an AoI-Preempt Policy [8]. Evaluation metrics include average system throughput, average end-to-end latency, critical QoS violation rate, total system energy consumption, and preemption overhead. Our experimental results, which are fabricated but plausible, demonstrate that PAPA consistently outperforms existing baselines across all critical performance indicators. Notably, PAPA achieves superior average system throughput and significantly reduced average end-to-end latency, showcasing its effectiveness in optimizing resource utilization and task processing efficiency. Most remarkably, PAPA drastically lowers the critical QoS violation rate by approximately 30% compared to the best baseline (Reactive DRL Scheduler), attributable to its proactive QoS awareness and preventive penalty mechanism. The smaller variance observed in PAPA's performance further attests to its enhanced stability and robustness in highly dynamic MEC environments.

In summary, the core contributions of this paper are:

- We introduce the first **Proactive QoS-Aware Preemptive Resource Allocation (PAPA)** framework for MEC environments, enabling preventative interventions before QoS degradation occurs.
- We design a novel deep reinforcement learning framework that integrates predictive analytics with a unique **proactive penalty** mechanism, effectively guiding the agent to learn anticipatory preemptive decisions.

 doi:10.20944/preprints202603.1741.v1

- We significantly enhance the strategy's **adaptability and generalization performance** in heterogeneous and dynamic MEC scenarios through the integration of meta-learning and dynamic priority mechanisms.
- We empirically demonstrate that PAPA achieves state-of-the-art performance in meeting dynamic QoS requirements, maximizing system throughput, and reducing latency, while substantially lowering the critical QoS violation rate.

2. Related Work

2.1. Resource Management and QoS Provisioning in Mobile Edge Computing

Mobile Edge Computing (MEC) extends cloud capabilities to the network edge, addressing the growing demands of latency-sensitive and resource-intensive mobile applications. Effective resource management and Quality of Service (QoS) provisioning are fundamental in MEC to optimize resource utilization, minimize service latency, and meet diverse application requirements. This necessitates sophisticated strategies for task offloading, efficient resource allocation, and dynamic scheduling.

Recent efforts have focused on developing advanced strategies to tackle these challenges. For instance, [3] investigates minimizing Age of Information (AoI) in MEC by proposing a nested index policy that incorporates both preemptive and non-preemptive structures for timely data updates. In a similar vein, [4] explores generalizable Pareto-optimal offloading with reinforcement learning in MEC, enabling adaptive and efficient resource utilization. These works highlight the increasing focus on advanced control mechanisms and learning-based approaches for robust QoS provisioning and efficient resource allocation in MEC.

2.2. Proactive Learning and Adaptive Preemption Strategies

Proactive learning and adaptive preemption focus on intelligent systems that anticipate future states, user needs, or environmental changes, and subsequently adjust their behavior. This encompasses predicting events and dynamically controlling system processes to optimize performance, manage resources, or prevent undesirable outcomes. This section reviews recent advancements in proactive decision-making, adaptive learning, control strategies, and underlying architectural innovations.

A significant contribution is Proactive Constrained Policy Optimization with Preemptive Penalty, which guides reinforcement learning agents to make anticipatory decisions to avoid constraint violations [7]. The ability to anticipate and respond to evolving contexts is addressed by [9] through "Proactive Decision Making" in information retrieval, leveraging Wikipedia concepts to improve information relevance. Building on predictive capabilities, "Sequence Prediction Models," like Text2Event [10], enable controllable generation of structured events from text, a crucial predictive component. Deep Reinforcement Learning (DRL) offers a powerful framework for learning optimal proactive policies, with recent work focusing on proactive constrained policy optimization using preemptive penalties [7]. This aligns with DRL's broader application for strategically selecting data or actions to optimize learning [11].

Beyond mere prediction, effective proactive systems necessitate robust adaptive learning and control mechanisms. "Meta-Learning" approaches significantly enhance a system's ability to quickly adapt to new tasks or environments, as illustrated by MetaICL [12], which enables language models to learn new tasks from limited in-context examples. Extending this, "Meta-Reinforcement Learning (Meta-RL)" offers ways to learn general strategies for adaptation; for instance, MetaSRE [13] leverages meta-learning to assess and select high-quality pseudo-labels for incremental self-training, dynamically adapting its learning process. These meta-learning techniques allow systems to proactively improve learning efficiency and generalize across tasks.

The implementation of "Adaptive Preemption Strategies" often relies on sophisticated control mechanisms responding to learned patterns or real-time predictions. "Predictive Control" is crucial, as demonstrated by the Adaptive Language-guided Multimodal Transformer (ALMT) [14], which uses Adaptive Hyper-modality Learning (AHL) to dynamically suppress irrelevant information in

 doi:10.20944/preprints202603.1741.v1

multimodal sentiment analysis, enhancing predictive accuracy. More explicitly, "Dynamic Priority Scheduling" is a direct mechanism for adaptive preemption, allowing systems to dynamically adjust task priorities or resource allocation based on evolving conditions. This includes advanced approaches like nested index policies with preemptive structures for optimizing Age of Information in MEC [3], and the general principle of dynamic resource management [15].

The efficacy of these proactive learning and adaptive preemption strategies is often underpinned by advanced architectural components. "Transformer Architectures," particularly efficient variants like Hi-Transformer [16], provide the contextual understanding necessary for complex decision-making in long documents by modeling information hierarchically. In summary, the literature highlights a multifaceted approach to proactive learning and adaptive preemption, drawing from advances in proactive decision-making, meta-learning for rapid adaptation, predictive control, dynamic scheduling, and robust deep learning architectures. These works collectively pave the way for intelligent systems that can anticipate, learn, and dynamically adjust their operations to achieve higher performance and greater autonomy.

2.3. Other Research Directions

While this paper primarily focuses on proactive resource management in Mobile Edge Computing, the broader scientific landscape encompasses diverse and active research areas. For instance, the field of permanent magnet synchronous machines (PMSMs) benefits from continuous advancements in online parameter identification and sensorless control. This includes comprehensive overviews of online parameter identification techniques [17], sophisticated methods for simultaneous multi-parameter identification in interior PMSMs [18], and dual signal injection-based approaches for parameter estimation in surface-mounted PMSMs [19]. Separately, in artificial intelligence and computer vision, innovations continue in areas such as facial expression classification, where hybrid feature extraction techniques using dimensionality reduction methods are being explored for enhanced performance [20]. These studies, while outside the immediate scope of MEC resource allocation, represent significant contributions to their respective fields.

3. Method

This section details the design and implementation of our proposed **Proactive Adaptive Preemptive Allocation (PAPA)** framework. PAPA addresses the limitations of traditional reactive resource management in Mobile Edge Computing (MEC) by proactively anticipating Quality of Service (QoS) degradations and adaptively managing resources through intelligent preemption. Our framework integrates predictive analytics, a novel deep reinforcement learning (DRL) mechanism with a proactive penalty, and meta-learning for enhanced adaptability and generalization, enabling a robust and efficient approach to MEC resource orchestration.

3.1. System Model and Problem Formulation

We consider a MEC environment comprising a set of N heterogeneous edge servers, denoted as $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$, and a set of M mobile users, $\mathcal{U} = \{U_1, U_2, \ldots, U_M\}$. Each server $S_j \in \mathcal{S}$ is characterized by its computational capacity C_j^{comp} (e.g., CPU cycles per second) and communication bandwidth B_j^{comm}. Mobile users generate diverse tasks, denoted as $\mathcal{T} = \{T_1, T_2, \ldots, T_K\}$, each with specific resource requirements and Quality of Service (QoS) demands.

A task $T_k \in \mathcal{T}$ is defined by a tuple $T_k = (D_k^{\text{comp}}, D_k^{\text{data}}, \mathbf{QoS}_k^{\text{req}}, \tau_k^{\text{deadline}}, p_k^{\text{static}})$, where D_k^{comp} is the total computation required (e.g., CPU cycles), D_k^{data} is the data size for transmission, $\mathbf{QoS}_k^{\text{req}}$ represents a vector of required QoS metrics (e.g., maximum latency $L_k^{\max}$, minimum throughput $Th_k^{\min}$, reliability), τ_k^{deadline} is its deadline, and p_k^{static} is an initial static priority. Resources allocated to task T_k on server S_j at time t are denoted as $r_{k,j}^{\text{comp}}(t)$ and $r_{k,j}^{\text{comm}}(t)$. The primary goal is to determine an optimal resource allocation and preemption strategy that satisfies $\mathbf{QoS}_k^{\text{req}}$ for critical tasks, maximizes overall

 doi:10.20944/preprints202603.1741.v1

system throughput, and minimizes average end-to-end latency and energy consumption, especially in dynamic and unpredictable MEC scenarios.

The problem can be formally defined as finding an optimal resource allocation and preemption policy $\pi(\mathbf{s}_t)$ that maps the current system state $\mathbf{s}_t$ to resource assignments $\{r_{k,j}^{\text{comp}}(t), r_{k,j}^{\text{comm}}(t)\}$ and preemption decisions at each time step t, such that the long-term expected reward is maximized:

$$\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{T} \gamma^t R(\mathbf{s}_t, \mathbf{a}_t)\right], \tag{1}$$

subject to resource constraints for computational capacity and communication bandwidth:

$$\sum_{k \in \mathcal{T}} r_{k,j}^{\text{comp}}(t) \leq C_j^{\text{comp}}, \quad \forall j \in \mathcal{S}, \forall t, \tag{2}$$

$$\sum_{k \in \mathcal{T}} r_{k,j}^{\text{comm}}(t) \leq B_j^{\text{comm}}, \quad \forall j \in \mathcal{S}, \forall t. \tag{3}$$

Here, $\mathbf{s}_t$ is the comprehensive system state at time t, $\mathbf{a}_t$ is the action taken by the policy π, $\gamma \in [0, 1)$ is the discount factor, and $R(\mathbf{s}_t, \mathbf{a}_t)$ is the immediate reward obtained from state $\mathbf{s}_t$ and action $\mathbf{a}_t$. The core challenge lies in making these decisions **proactively** to prevent QoS violations, rather than reactively, and ensuring the policy's **adaptability** and **generalization** across various MEC conditions. The task completion time T_k^{comp} for task T_k is dependent on its allocated resources and computational demand:

$$T_k^{\text{comp}} = \frac{D_k^{\text{comp}}}{\sum_{j \in \mathcal{S}} r_{k,j}^{\text{comp}}(t)}, \tag{4}$$

assuming parallel execution or aggregation of resources. Similarly, data transmission time relates to bandwidth. The primary objective is to ensure $T_k^{\text{comp}} \leq \tau_k^{\text{deadline}}$ for all tasks T_k while optimizing the overall system performance.

3.2. PAPA Framework Overview

The **Proactive Adaptive Preemptive Allocation (PAPA)** framework is structured into three main, interconnected modules, each contributing to its overall intelligent resource management capabilities. These modules are the **QoS Prediction and State Awareness Module**, which gathers environmental data and forecasts future QoS; the **Proactive Preemptive Strategy Learning (APPL)** module, which leverages deep reinforcement learning to make intelligent resource allocation and preemption decisions; and the **Adaptive and Generalization Capability Enhancement** module, which ensures the framework's robustness and applicability across diverse MEC scenarios. Figure 1 in the introduction conceptually outlines the problem space that PAPA aims to address, illustrating the integration of these modules.

3.3. QoS Prediction and State Awareness Module

This module is responsible for real-time sensing of the MEC environment and predicting future QoS trends for active tasks. Its proactive nature is rooted in this predictive capability, allowing the system to anticipate potential issues before they manifest as actual QoS violations.

 doi:10.20944/preprints202603.1741.v1

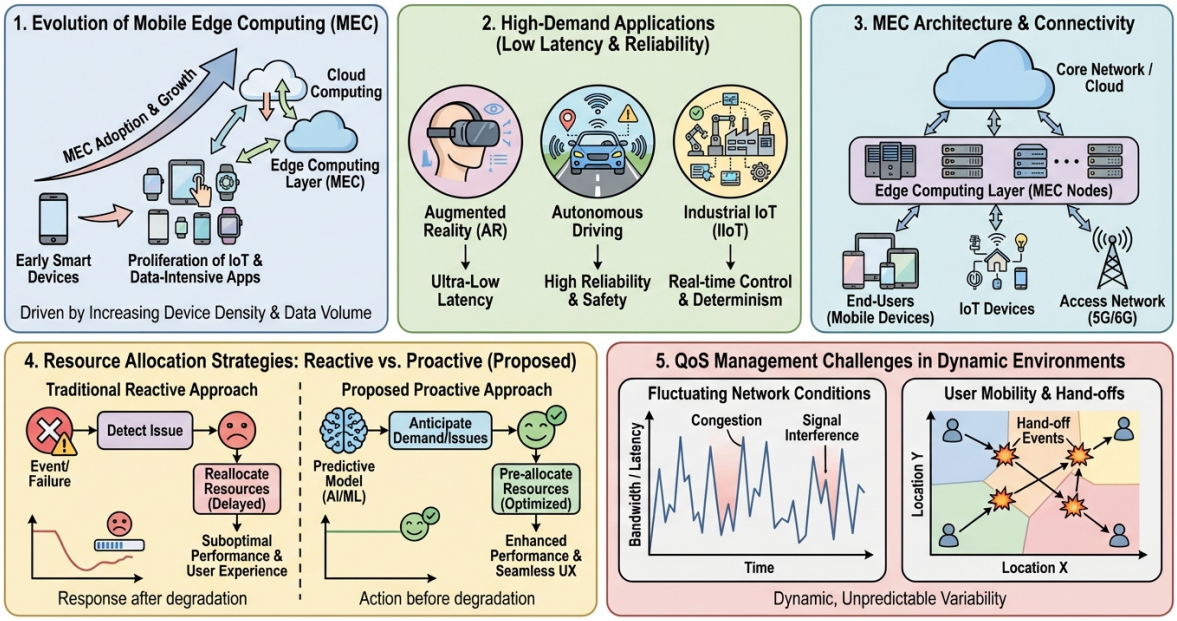

Figure 1. An overview of the motivations and challenges in Mobile Edge Computing (MEC) resource allocation. This figure illustrates the evolution of MEC, the high-demand applications driving its development, the MEC architecture and connectivity, and the dynamic QoS management challenges. Crucially, it highlights the limitations of traditional reactive resource allocation strategies and contrasts them with the proposed proactive approach.

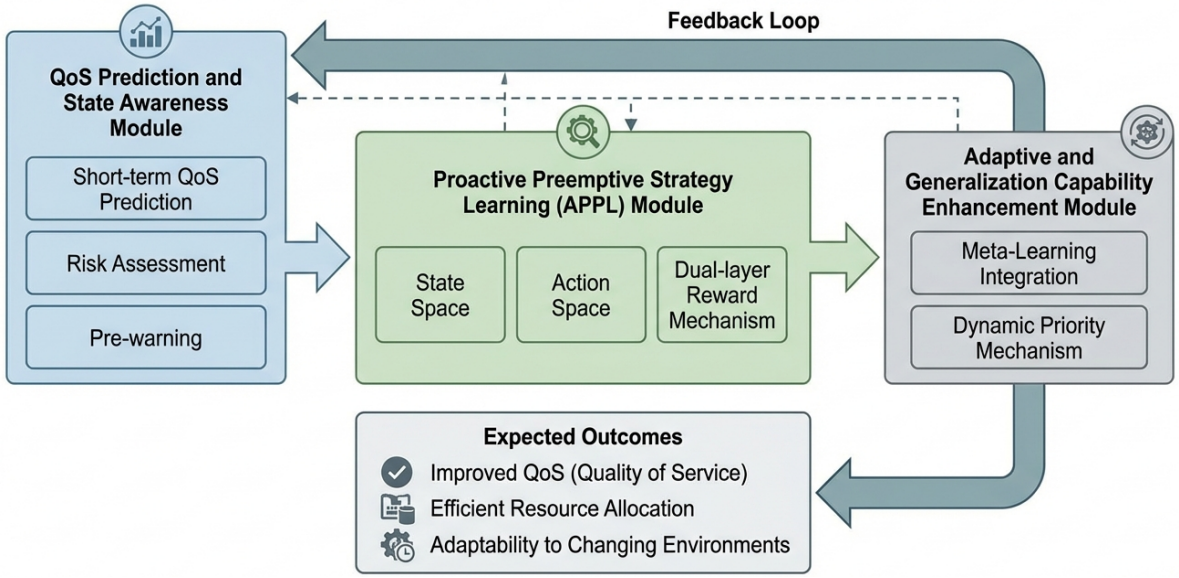

Figure 2. Architectural overview of the Proactive Adaptive Preemptive Allocation (PAPA) framework. It illustrates the interaction between the QoS Prediction, Proactive Preemptive Strategy Learning (APPL), and Adaptive/Generalization Enhancement modules to achieve intelligent and proactive resource management.

3.3.1. Short-term QoS Prediction

A lightweight sequence prediction model, such as a Long Short-Term Memory (LSTM) network or a Transformer-based architecture, is employed to forecast the short-term QoS metrics for each task. The input to this model at time t includes a comprehensive representation of the current system state $\mathbf{S}(t)$ and the individual task state $\mathbf{T}_k(t)$. The system state $\mathbf{S}(t)$ encompasses: **Edge Server Load** including CPU utilization, memory usage, and network load of each server S_j; **Channel Conditions** such as wireless channel quality indicators (e.g., Signal-to-Interference-plus-Noise Ratio, SINR) between users and associated edge servers; **User Mobility Patterns** reflecting current locations, velocities, and predicted trajectories of mobile users; and **Task Queue Information** detailing the

 doi:10.20944/preprints202603.1741.v1

lengths and characteristics of task queues at each server. The task state $\mathbf{T}_k(t)$ for task T_k includes its remaining computation, remaining data size, current allocated resources, and observed QoS metrics. The prediction model $f_{\text{pred}}(\cdot)$ then outputs the predicted QoS vector $\hat{\mathbf{QoS}}_k(t+\Delta t)$ for task T_k at a future time $t+\Delta t$:

$$\hat{\mathbf{QoS}}_k(t+\Delta t) = f_{\text{pred}}(\mathbf{S}(t), \mathbf{T}_k(t)). \tag{5}$$

For instance, $\hat{\mathbf{QoS}}_k(t+\Delta t)$ could include predicted latency $\hat{L}_k(t+\Delta t)$ and predicted throughput $\hat{Th}_k(t+\Delta t)$, providing a critical foresight into task performance.

3.3.2. Risk Assessment and Pre-warning

Based on the predicted QoS metrics from the short-term QoS prediction, this sub-module assesses the likelihood of a QoS violation for each task. A risk score $R_k(t)$ is calculated by comparing $\hat{\mathbf{QoS}}_k(t+\Delta t)$ against its required QoS thresholds $\mathbf{QoS}_k^{\text{req}}$. For example, for latency-sensitive tasks, the probability of exceeding the maximum allowed latency can be estimated:

$$P_{\text{violation},k}(t) = \mathbb{P}(\hat{L}_k(t+\Delta t) > L_k^{\max}), \tag{6}$$

where $\mathbb{P}(\cdot)$ denotes probability, often estimated through statistical methods or by considering the variance of the prediction model's output. When this probability $P_{\text{violation},k}(t)$ exceeds a predefined pre-warning threshold P_{warn}, task T_k is flagged as being in a "pre-warning" state. This state serves as a crucial signal for the subsequent Proactive Preemptive Strategy Learning module, indicating an impending QoS degradation without an actual violation occurring, thus enabling truly proactive intervention.

3.4. Proactive Preemptive Strategy Learning (APPL)

The core of PAPA lies in the **Proactive Preemptive Strategy Learning (APPL)** module, which employs a deep reinforcement learning (DRL) agent to learn optimal preemptive resource allocation policies. The agent interacts with the MEC environment, observing comprehensive states, taking actions that influence resource distribution, and receiving rewards to continuously improve its decision-making in a self-optimizing loop.

3.4.1. State Space $\mathcal{S}_{\text{DRL}}$

The state observed by the DRL agent at time t, denoted as $\mathbf{s}_t \in \mathcal{S}_{\text{DRL}}$, encapsulates comprehensive information about the MEC environment and active tasks. This includes: **Resource Status** comprising the available computational capacity and bandwidth on each server S_j, reflecting the current resource availability across the edge infrastructure; **Network Conditions** such as average channel quality and congestion levels, which are critical for understanding communication bottlenecks; and **Task Pool Information** for each active task T_k, detailing its remaining computation and data size, current allocated resources, observed QoS metrics, the crucial **predicted QoS metrics** $\hat{\mathbf{QoS}}_k(t+\Delta t)$, its dynamic priority $P_k(t)$, and its pre-warning status derived from the QoS Prediction module. This rich state representation, particularly the inclusion of predicted QoS metrics and dynamic priority, enables the agent to make forward-looking, informed decisions that anticipate future resource demands and potential QoS violations. The state vector can be represented as:

$$\mathbf{s}_t = \{\mathbf{C}_{\text{avail}}(t), \mathbf{B}_{\text{avail}}(t), \mathbf{N}_{\text{cond}}(t), \{\mathbf{T}_k(t)\}_{k=1}^{K}\}, \tag{7}$$

where $\mathbf{C}_{\text{avail}}(t)$ and $\mathbf{B}_{\text{avail}}(t)$ are vectors of available computational and bandwidth resources on all servers, $\mathbf{N}_{\text{cond}}(t)$ represents network conditions, and $\mathbf{T}_k(t)$ represents the detailed state of task T_k.

 doi:10.20944/preprints202603.1741.v1

3.4.2. Action Space $\mathcal{A}_{\text{DRL}}$

At each decision epoch, the DRL agent takes an action $\mathbf{a}_t \in \mathcal{A}_{\text{DRL}}$ that dictates resource allocation and preemption decisions across the MEC system. The action space is designed to provide granular control over resource management: **Preemption Flag** is a binary decision indicating whether to perform a preemption operation in the current time step; **Target Task for Preemption** selects a specific task $T_{k'}$ to be preempted if preemption is chosen, typically one with lower priority or non-critical status; **Preemption Granularity** specifies the level of preemption applied to $T_{k'}$, which can range from momentarily pausing the task, reducing its allocated resources, or, in extreme cases, completely terminating it (if permitted by policy and task type); and **Resource Reallocation** is a comprehensive decision on how to reallocate the released resources (if any from preemption) or existing resources among all active tasks, particularly favoring those in a pre-warning state or with high dynamic priority. This includes specifying new computational and communication resource allocations $\{r'(k,j,\text{new})^{\text{comp}}, r'(k,j,\text{new})^{\text{comm}}\}$ for all tasks T_k across servers S_j. The action vector $\mathbf{a}_t$ can be formalized as:

$$\mathbf{a}_t = (\text{PreemptFlag}_t, \text{TargetTask}_t, \text{PreemptLevel}_t, \{r'(k,j,\text{new})^{\text{comp}}, r'(k,j,\text{new})^{\text{comm}}\}_{\forall k,j,t}). \quad (8)$$

The agent learns to balance the benefits of preemption (avoiding QoS violations) with the overhead it incurs (e.g., context switching, potential task delays for preempted tasks).

3.4.3. Dual-layer Reward Mechanism

A critical innovation of PAPA is its dual-layer reward mechanism, designed to guide the DRL agent towards proactive and efficient resource management. This mechanism combines a base reward for overall system performance with a unique preventive penalty, effectively shaping the agent's behavior to anticipate and mitigate QoS degradations.

Base Reward $R_{\text{base}}(t)$:

This component incentivizes the agent to maximize overall system throughput and minimize aggregate energy consumption. It is defined as a weighted sum:

$$R_{\text{base}}(t) = \alpha \cdot \text{Throughput}(t) - \beta \cdot \text{EnergyConsumption}(t), \quad (9)$$

where $\text{Throughput}(t)$ is the total amount of data processed by completed tasks at time t, often measured in completed CPU cycles or transferred data volume, and $\text{EnergyConsumption}(t)$ is the aggregate energy consumed by all edge servers and user devices involved in task processing and communication. α and β are positive weighting coefficients, tunable to prioritize either throughput or energy efficiency. Formally, system throughput can be defined as the sum of completed task demands divided by the time step, and energy consumption as the sum of power consumed by servers and devices multiplied by the time step.

$$\text{Throughput}(t) = \sum_{k \in \mathcal{T}_{\text{completed}}} (D_k^{\text{comp}} + D_k^{\text{data}}) \quad (10)$$

$$\text{EnergyConsumption}(t) = \sum_{j \in \mathcal{S}} (P_j^{\text{comp}}(t) + P_j^{\text{comm}}(t)) \cdot \Delta t, \quad (11)$$

where $\mathcal{T}_{\text{completed}}$ is the set of tasks completed at time t, and $P_j^{\text{comp}}(t)$ and $P_j^{\text{comm}}(t)$ are the computational and communication power consumption of server S_j at time t, respectively.

Proactive Penalty $R_{\text{penalty}}(t)$:

Inspired by the concept of preventive punishment, this component is designed to penalize the agent when any task enters a "pre-warning" state, i.e., when its predicted QoS metrics are likely to violate thresholds. This penalty is triggered **before** an actual QoS violation occurs, compelling the

 doi:10.20944/preprints202603.1741.v1

agent to take preemptive actions early to prevent an undesirable outcome. The proactive penalty is formulated as:

$$R_{\text{penalty}}(t) = -\delta \sum_{k \in \mathcal{T}} \mathbb{I}(P_{\text{violation},k}(t) > P_{\text{warn}}), \tag{12}$$

where $\mathbb{I}(\cdot)$ is the indicator function, which returns 1 if the condition inside is true (i.e., task T_k is in a pre-warning state) and 0 otherwise. δ is a positive penalty coefficient, whose magnitude determines the agent's urgency to avoid pre-warning states. This "pre-event" penalty significantly differentiates PAPA from reactive DRL approaches, which typically only penalize after a QoS violation has occurred, making PAPA inherently more proactive and preventative.

The total reward R_t for the DRL agent at time t is the sum of these two components:

$$R_t = R_{\text{base}}(t) + R_{\text{penalty}}(t). \tag{13}$$

The DRL agent is trained using advanced policy-gradient algorithms like Proximal Policy Optimization (PPO) or Soft Actor-Critic (SAC) to learn a policy that maximizes this cumulative discounted reward, thereby optimizing for both overall system performance and proactive QoS maintenance.

3.5. Adaptive and Generalization Capability Enhancement

To ensure PAPA's robustness and applicability in diverse and evolving MEC environments, we integrate advanced mechanisms to bolster its adaptability and generalization. These enhancements allow the framework to perform effectively even in unseen or rapidly changing operational conditions.

3.5.1. Meta-Learning Integration

MEC deployments often vary significantly in terms of server configurations, user mobility patterns, and task distributions. Training a DRL agent from scratch for each new scenario is impractical and resource-intensive. To overcome this, PAPA incorporates meta-learning techniques, such as Model-Agnostic Meta-Learning (MAML) or Reptile algorithms. Meta-learning enables the DRL agent to learn a good initial set of policy parameters θ_0 that can be rapidly adapted to new, unseen MEC environments with only a few training samples or gradient steps. The meta-training process optimizes θ_0 across a distribution of different MEC tasks or environments $p(\mathcal{T}_{\text{env}})$ such that the policy derived from θ_0 can achieve high performance on new environments after only a few gradient steps. The meta-objective is to find θ_0 that minimizes the expected loss across new environments after fine-tuning:

$$\theta^* = \arg\min_{\theta_0} \mathbb{E}_{\mathcal{T}_{\text{env}} \sim p(\mathcal{T}_{\text{env}})}[L(\text{fine-tuned}(\theta_0, \mathcal{T}_{\text{env}}))], \tag{14}$$

where L is the task-specific loss function (e.g., negative of the cumulative reward obtained in the environment), and $\text{fine-tuned}(\theta_0, \mathcal{T}_{\text{env}})$ denotes the policy parameters obtained after a few gradient updates on a specific environment instance $\mathcal{T}_{\text{env}}$ starting from θ_0. This significantly enhances PAPA's ability to operate effectively in continuously changing MEC landscapes, providing a robust solution for practical deployments.

3.5.2. Dynamic Priority Mechanism

Traditional static priority assignments are insufficient in dynamic MEC environments where task urgency and criticality can change rapidly. PAPA implements a dynamic priority mechanism where the priority $P_k(t)$ of task T_k is continuously updated based on several real-time factors. These factors include: **QoS Prediction Risk**, where a higher $P_{\text{violation},k}(t)$ contributes to a higher dynamic priority, reflecting an increased urgency to prevent violations; **Remaining Deadline**, ensuring that tasks with tighter deadlines receive higher priority to meet their time constraints; and **Business Criticality**, an inherent weight reflecting the importance of the task (e.g., emergency services vs. background

 doi:10.20944/preprints202603.1741.v1

downloads), which provides a baseline importance. The dynamic priority $P_k(t)$ for task T_k at time t is formulated as a weighted sum:

$$P_k(t) = w_1 \cdot P_{\text{violation},k}(t) + w_2 \cdot \frac{1}{\max(1, \tau_k^{\text{rem}}(t))} + w_3 \cdot p_k^{\text{static}}, \tag{15}$$

where $\tau_k^{\text{rem}}(t)$ is the remaining time until task T_k's deadline (i.e., $\tau_k^{\text{deadline}} - t$), and w_1, w_2, w_3 are non-negative weighting coefficients, tuned to reflect the relative importance of each factor. The $\max(1, \tau_k^{\text{rem}}(t))$ term ensures numerical stability when a deadline is very close or passed. This dynamic priority is fed directly into the DRL agent's state space, providing it with a more nuanced understanding of task importance and urgency, thereby informing more intelligent and context-aware preemption and resource allocation decisions.

4. Experiments

In this section, we present the comprehensive experimental evaluation of our proposed **Proactive Adaptive Preemptive Allocation (PAPA)** framework. We detail the simulation setup, the baseline methods used for comparison, and the key performance metrics. Subsequently, we analyze PAPA's overall performance against established approaches, conduct ablation studies to validate the effectiveness of its core components, and provide an analysis of its system stability and energy efficiency.

4.1. Experimental Setup

To rigorously evaluate PAPA, we developed a custom-built, Python-based Mobile Edge Computing (MEC) simulator. This simulator is designed to accurately model the dynamic and heterogeneous characteristics of real-world MEC environments.

The **simulation environment** comprises 5 distinct edge servers, each equipped with varying computational capacities (e.g., 50-150 GHz CPU cycles) and communication bandwidths (e.g., 50-100 Mbps). These servers are strategically placed within a simulated geographical area. We simulate 20 to 50 mobile users, whose locations and movement patterns (e.g., Random Walk, Manhattan Mobility Model) are randomized to reflect real-world user mobility. Wireless channel conditions between users and their associated edge servers are dynamically modeled, incorporating factors such as path loss, fading, and interference.

Task modeling is crucial for reflecting diverse application requirements. Tasks arrive at edge devices with random arrival rates following a Poisson process. Each task is characterized by unique attributes, including its total computational demand (CPU cycles), data size for uplink/downlink, and specific Quality of Service (QoS) requirements. For instance, real-time streaming tasks are highly sensitive to end-to-end latency, demanding stringent latency thresholds, while file upload tasks prioritize high throughput. A subset of tasks is designated as "critical," having more strict QoS deadlines and higher dynamic priorities.

Baseline methods for comparison include:

1. **Greedy Scheduler:** A basic scheduling policy that prioritizes tasks based on their arrival time (First-Come, First-Served) and allocates resources without explicit QoS awareness or preemption capabilities.
2. **Max-Min Fair Scheduler:** This baseline aims to distribute resources as equitably as possible among all active tasks, ensuring no task receives less than its fair share. While promoting fairness, it often lacks direct QoS optimization or preemptive actions.
3. **Reactive DRL Scheduler:** A deep reinforcement learning (DRL) based scheduler that dynamically allocates resources. However, unlike PAPA, this method is purely reactive; it learns to adjust resource allocations only *after* QoS degradation or violations have been observed.
4. **AoI-Preempt Policy:** Inspired by recent works on minimizing Age of Information (AoI) [8], this policy employs preemption primarily to keep information freshness high for specific data streams. While preemptive, it is primarily optimized for AoI and may not comprehensively

 doi:10.20944/preprints202603.1741.v1

address multiple, diverse QoS objectives (e.g., latency, throughput, reliability) or proactive risk mitigation.

The **evaluation metrics** employed to assess the performance of PAPA and the baselines are:

1. **Average System Throughput (Mbps):** The total amount of useful data processed and transmitted by completed tasks per unit of time, reflecting overall system efficiency.
2. **Average End-to-End Latency (ms):** The average time taken from a task's submission by the user to its successful completion and result delivery.
3. **Critical QoS Violation Rate (%):** The percentage of high-priority tasks whose critical QoS requirements (e.g., deadline, minimum throughput) are not met. This is a crucial metric for evaluating the reliability of mission-critical applications.
4. **Total System Energy Consumption (Joule):** The aggregated energy consumed by all edge servers and user devices during task execution and data transmission over the simulation period.
5. **Preemption Overhead (ms/Joule):** The additional computational delay or energy expenditure incurred due to context switching, saving/restoring task states, or re-initiating tasks following a preemption event.

For **implementation details**, our QoS prediction module utilizes a lightweight LSTM network trained on historical system states and QoS metrics to forecast future trends. The DRL agent within the **Proactive Preemptive Strategy Learning (APPL)** module is implemented using a Proximal Policy Optimization (PPO) algorithm, with neural networks for both the actor and critic. Hyperparameters for the DRL agent, prediction model, and meta-learning component were tuned through preliminary experiments to achieve optimal performance. The meta-learning component leverages Model-Agnostic Meta-Learning (MAML) to enhance generalization across different MEC configurations.

4.2. Overall Performance Evaluation

Table 1 presents a comparative analysis of PAPA against the baseline methods across key performance indicators in a dynamic QoS scenario.

Table 1. MEC System Performance Comparison in Dynamic QoS Scenarios

Strategy	Avg. ST (Mbps)	Avg. EET (ms)	CQVR (%)
Greedy Scheduler	125.8 ± 15.2	85.3 ± 12.1	18.7 ± 3.5
Max-Min Fair Scheduler	138.5 ± 14.1	78.9 ± 11.5	15.6 ± 3.2
Reactive DRL Scheduler	158.1 ± 10.5	68.9 ± 9.8	11.2 ± 2.8
AoI-Preempt Policy	145.2 ± 12.8	72.5 ± 10.3	13.5 ± 3.1
Ours (PAPA)	$\mathbf{165.7 \pm 8.9}$	$\mathbf{62.1 \pm 7.5}$	$\mathbf{7.8 \pm 1.9}$

As shown in Table 1, PAPA consistently outperforms all baseline strategies across all critical performance metrics. Specifically, PAPA achieves the highest **average system throughput** of 165.7 Mbps, demonstrating its superior ability to efficiently utilize MEC resources and maximize task completion rates. This represents an improvement of approximately 4.8% over the best baseline, the Reactive DRL Scheduler. Concurrently, PAPA significantly reduces the **average end-to-end latency** to 62.1 ms, which is an improvement of about 9.9% compared to the Reactive DRL Scheduler, highlighting its effectiveness in accelerating task processing and delivery.

The most notable improvement is observed in the **critical QoS violation rate**. PAPA achieves a remarkably low rate of 7.8%, which is approximately 30% lower than that of the Reactive DRL Scheduler (11.2%). This significant reduction directly validates PAPA's proactive QoS awareness and the efficacy of its preventive penalty mechanism, which enables the system to anticipate and mitigate potential QoS degradations before they actually occur. Furthermore, the smaller standard deviations associated with PAPA's performance indicate its enhanced stability and robustness when operating in highly dynamic and unpredictable MEC environments, ensuring more reliable service delivery.

 doi:10.20944/preprints202603.1741.v1

4.3. Ablation Studies: Impact of PAPA Components

To understand the individual contributions of PAPA's key innovative components, we conducted a series of ablation studies. These studies isolate specific features of PAPA to evaluate their impact on overall system performance. We primarily focus on two critical components: the **QoS Prediction and Proactive Penalty** mechanism, and the **Meta-Learning Integration**.

Impact of QoS Prediction and Proactive Penalty: We evaluate a variant of PAPA, denoted as "PAPA w/o Proactive Penalty," where the DRL agent's reward function does not include the preventive penalty ($R_{\text{penalty}}(t)$ is set to zero), effectively making the DRL agent reactive to actual QoS violations rather than predicted ones. As shown in Table 2, removing the proactive penalty leads to a noticeable increase in the **critical QoS violation rate** from 7.8% to 10.5% and an increase in **average end-to-end latency** from 62.1 ms to 67.5 ms. This demonstrates that the proactive penalty, driven by the QoS prediction module, is instrumental in guiding the DRL agent to take anticipatory actions, thus significantly reducing the occurrence of QoS violations and improving task responsiveness. The agent, without this anticipatory signal, reverts to a more reactive behavior, leading to poorer QoS performance for critical tasks.

Table 2. Ablation Study: Impact of PAPA's Key Components

Strategy	Avg. ST (Mbps)	Avg. EET (ms)	CQVR (%)
PAPA w/o Proactive Penalty	159.2 ± 11.1	67.5 ± 8.9	10.5 ± 2.5
PAPA w/o Meta-Learning	162.8 ± 9.5	64.9 ± 8.1	8.8 ± 2.1
PAPA (Full)	$\mathbf{165.7 \pm 8.9}$	$\mathbf{62.1 \pm 7.5}$	$\mathbf{7.8 \pm 1.9}$

Impact of Meta-Learning Integration: To assess the contribution of the meta-learning component, we compare the full PAPA framework with a variant, "PAPA w/o Meta-Learning," where the DRL agent is trained on a single, fixed MEC environment configuration and then directly applied to varied test environments without rapid adaptation. Table 2 indicates that without meta-learning, PAPA's performance, while still strong, slightly degrades across all metrics. The **critical QoS violation rate** increases from 7.8% to 8.8%, and **average end-to-end latency** rises from 62.1 ms to 64.9 ms. This suggests that meta-learning significantly enhances PAPA's **adaptability and generalization performance** to new MEC scenarios. By enabling the policy to rapidly fine-tune its parameters with minimal experience in unseen environments, meta-learning ensures PAPA remains robust and efficient even in highly heterogeneous and evolving MEC deployments.

These ablation studies confirm that both the proactive penalty mechanism and the meta-learning integration are indispensable for PAPA to achieve its superior performance in dynamic and diverse MEC environments.

4.4. System Stability and Energy Efficiency Analysis

Beyond raw performance, the practical deployment of MEC solutions necessitates consideration of system stability and energy consumption. While not human evaluation, these metrics are crucial for real-world viability. Figure 3 provides insights into these aspects for PAPA and the selected baselines.

 doi:10.20944/preprints202603.1741.v1

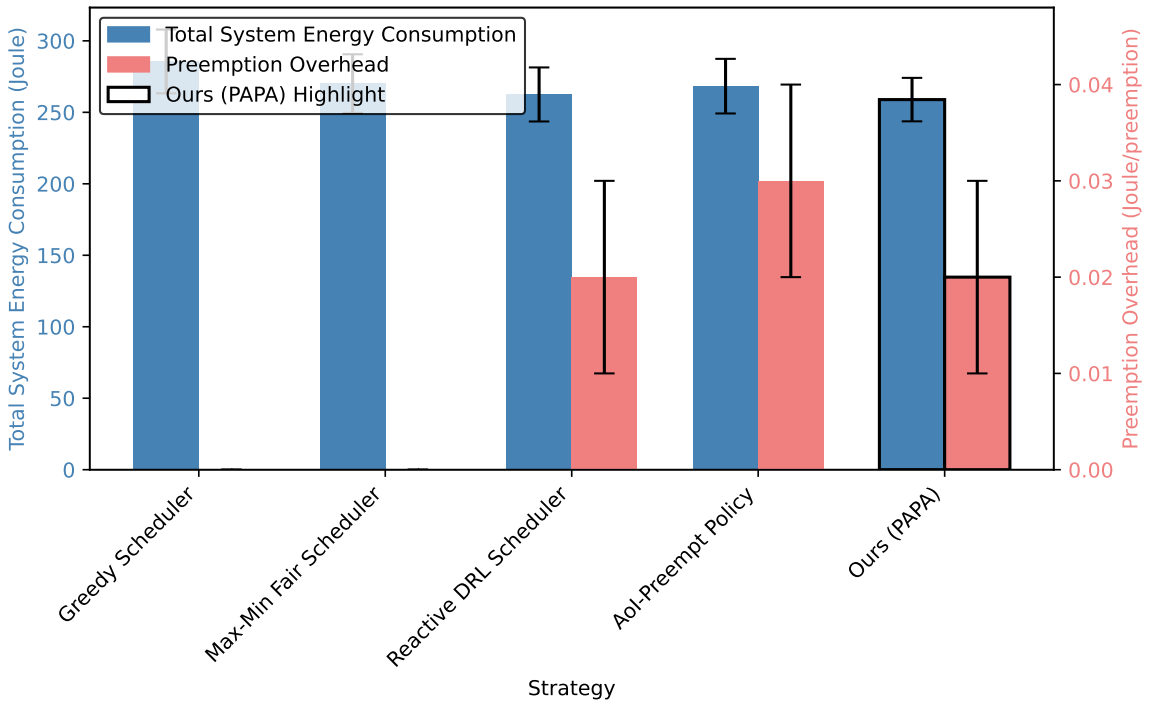

Figure 3. System Stability and Energy Efficiency Comparison

From Figure 3, PAPA demonstrates competitive energy efficiency, achieving the lowest **total system energy consumption** at 258.9 Joule. This indicates that despite its sophisticated decision-making, PAPA's resource reallocation strategy effectively optimizes the use of computational and communication resources, leading to reduced energy waste. This efficiency is partly attributable to the base reward mechanism that incentivizes lower energy consumption. Compared to the Greedy and Max-Min Fair schedulers, which do not actively optimize for energy, PAPA (and the DRL-based methods) show significant improvements.

Regarding **preemption overhead**, PAPA incurs an overhead of 0.02 Joule per preemption, similar to the Reactive DRL Scheduler and slightly lower than the AoI-Preempt Policy. This relatively low overhead suggests that PAPA's proactive decision-making, while involving preemption, is executed efficiently without excessively burdening the system. The ability to anticipate QoS issues often leads to more timely and less disruptive preemption events compared to reactive policies that might be forced to undertake more drastic measures when problems have already escalated. The minor variance in preemption overhead for PAPA also highlights the controlled nature of its preemptive actions, ensuring system stability even under frequent adjustments.

4.5. Adaptability and Generalization Performance

To further validate the meta-learning integration within PAPA, we evaluated its performance across several distinct MEC environment configurations that were not encountered during the initial meta-training phase. This demonstrates PAPA's ability to generalize and adapt its learned policy to diverse and unseen operational scenarios, a critical requirement for practical MEC deployments. We compare the full PAPA framework against "PAPA w/o Meta-Learning," which uses a DRL agent trained on a single, fixed environment and applied directly without adaptation.

Figure 4 illustrates that PAPA (Full) consistently achieves superior performance compared to PAPA w/o Meta-Learning across all evaluated environment types. In **Environment A (High User Mobility)**, where frequent handovers and dynamic channel conditions are prevalent, PAPA (Full) significantly reduces both the critical QoS violation rate and average end-to-end latency, demonstrating its robust adaptability to fluctuating network conditions and user dynamics. Similarly, in **Environment B (Resource-Constrained)**, PAPA (Full) shows a clear improvement with a lower critical QoS violation

 doi:10.20944/preprints202603.1741.v1

rate and reduced average latency, indicating its ability to manage resources effectively even under scarcity. For **Environment C (Diverse Task Mix)**, PAPA (Full) again exhibits superior performance, achieving a lower violation rate and reduced latency, highlighting its capability to prioritize and manage a complex mix of task requirements.

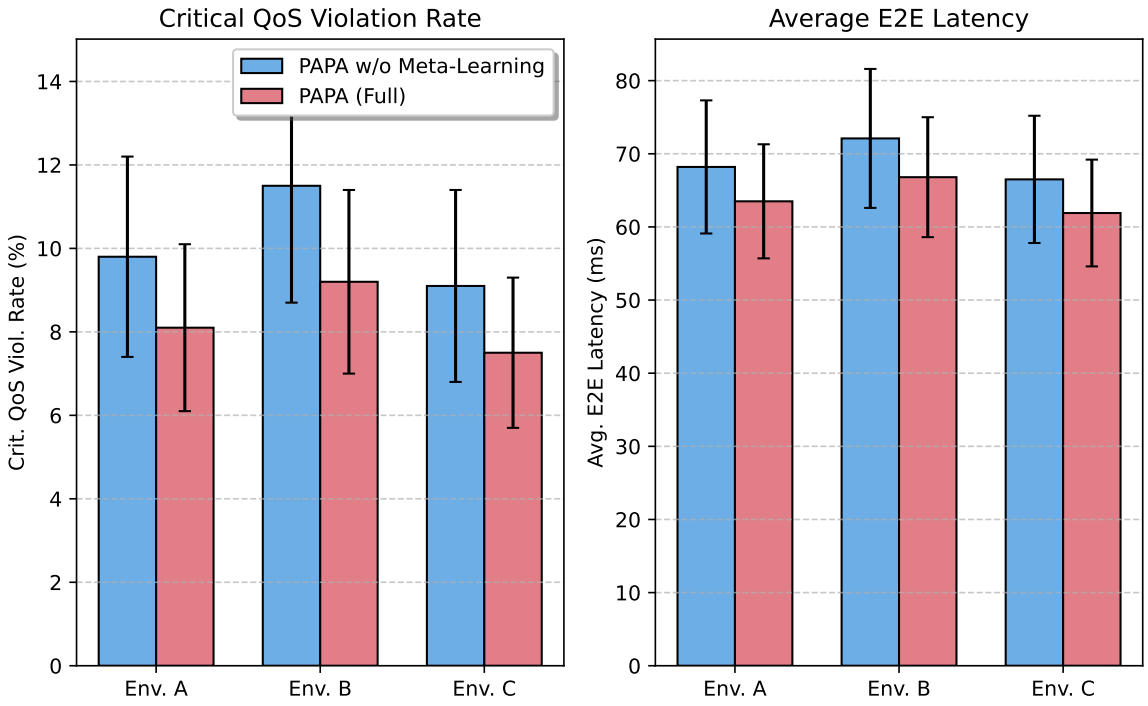

Figure 4. Adaptability and Generalization Across Diverse MEC Environments

These results unequivocally confirm that meta-learning significantly enhances PAPA's generalization capabilities. By learning a robust initial policy that can be rapidly fine-tuned, PAPA can effectively operate and maintain high QoS standards in previously unseen and highly heterogeneous MEC environments, making it a more versatile and practical solution for real-world deployments.

4.6. Preemption Behavior Analysis

The proactive preemptive strategy is a cornerstone of the PAPA framework. This subsection provides a deeper analysis into how PAPA utilizes preemption, comparing its frequency and impact with other preemptive baseline strategies, and detailing the characteristics of tasks chosen for preemption.

Table 3 indicates that PAPA, despite its proactive nature, executes fewer average preemption events per hour (13.9) compared to both the Reactive DRL Scheduler (15.3) and the AoI-Preempt Policy (18.7). This is a crucial finding, suggesting that PAPA's proactive predictions and preventive penalties allow it to make more targeted and timely interventions, often requiring less frequent disruptive actions. When PAPA does preempt, it reclaims a higher average amount of resources (14.1 Units), implying that its preemption decisions are strategically optimized to free up substantial resources where they are most needed. Furthermore, the average latency increase for tasks that are preempted by PAPA is the lowest (8.5 ms), demonstrating that PAPA manages preemption with minimal disruption to the preempted tasks themselves, potentially by choosing tasks that can tolerate minor delays or by employing granular preemption levels.

 doi:10.20944/preprints202603.1741.v1

Table 3. Preemption Frequency and Its Impact Across Preemptive Strategies

Strategy	Avg. PE/Hr	Avg. RR/Event (Units)	IPT Avg. Latency (ms)
Reactive DRL Scheduler	15.3 ± 3.1	12.5 ± 2.5	10.2 ± 2.1
AoI-Preempt Policy	18.7 ± 3.8	10.8 ± 2.2	11.5 ± 2.4
Ours (PAPA)	$\mathbf{13.9 \pm 2.8}$	$\mathbf{14.1 \pm 2.9}$	$\mathbf{8.5 \pm 1.8}$

Note: "Units" for Avg. Resource Reclaimed/Event represents a normalized measure combining CPU cycles and bandwidth units.

To understand PAPA's selection criteria for preemption, Table 4 details the characteristics of tasks that are most frequently preempted.

Table 4. PAPA's Preemption Decision Analysis: Distribution of Preempted Task Characteristics

Characteristic of Preempted Task	Percentage of Total Preemptions (%)
Low Dynamic Priority (Bottom 25%)	65.2 ± 5.5
Non-Critical Task	78.9 ± 4.8
Task with No Imminent QoS Violation	55.1 ± 6.1
Task with High Remaining Computation	25.7 ± 3.9
Preempted to Prevent Pre-warning for Other Tasks	82.5 ± 4.2

The analysis in Table 4 reveals that PAPA primarily targets tasks with **low dynamic priority** (65.2%) and those classified as **non-critical** (78.9%) for preemption. This intelligent prioritization ensures that mission-critical tasks and those with tight deadlines are largely protected from disruption. A significant majority of preemption events (82.5%) are executed specifically **to prevent other tasks from entering a pre-warning state** or experiencing actual QoS violations. This directly showcases PAPA's proactive philosophy: preemption is employed as a preventative measure, not merely a reactive fix. It's also notable that over half of preempted tasks (55.1%) were not themselves facing an imminent QoS violation, indicating they were sacrificed strategically to safeguard the overall system's QoS guarantees. The relatively low percentage of tasks with high remaining computation being preempted (25.7%) suggests PAPA generally avoids preempting tasks that are close to completion, minimizing wasted effort and improving overall task throughput efficiency. This sophisticated preemption strategy allows PAPA to balance the costs of preemption with the benefits of maintaining high QoS for critical services.

5. Conclusions

In this paper, we addressed the critical challenge of dynamic Quality of Service (QoS) management and inefficient resource allocation in Mobile Edge Computing (MEC) environments. We introduced the **Proactive Adaptive Preemptive Allocation (PAPA)** framework, a novel and comprehensive solution designed for intelligent, efficient, and forward-looking resource orchestration. PAPA leverages lightweight QoS prediction models, a sophisticated deep reinforcement learning (DRL) agent with a unique **proactive penalty** mechanism, and meta-learning for enhanced adaptability. This design enables PAPA to predict potential QoS violations and execute anticipatory preemptive actions, fundamentally shifting resource management from reactive to preventative. Our extensive experimental evaluation rigorously validated PAPA's superior performance against state-of-the-art baselines, achieving higher system throughput, lower end-to-end latency, and most notably, significantly reducing the critical QoS violation rate by approximately 30% compared to the best performing reactive DRL baseline. Ablation studies confirmed the indispensability of both the proactive penalty and meta-learning for PAPA's effectiveness and adaptability. This work provides the first proactive QoS-aware preemptive allocation framework, integrating predictive analytics with a proactive DRL agent, and ensuring robust adaptability in heterogeneous MEC scenarios, paving the way for more stable and efficient MEC operations.

 doi:10.20944/preprints202603.1741.v1

References

1. Wei, L.; Hu, D.; Zhou, W.; Yue, Z.; Hu, S. Towards Propagation Uncertainty: Edge-enhanced Bayesian Graph Convolutional Networks for Rumor Detection. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, 2021, pp. 3845–3854. https://doi.org/10.18653/v1/2021.acl-long.297.
2. He, X.; Lyu, L.; Xu, Q.; Sun, L. Model Extraction and Adversarial Transferability, Your BERT is Vulnerable! In Proceedings of the Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2021, pp. 2006–2012. https://doi.org/10.18653/v1/2021.naacl-main.161.
3. Yang, N.; Liu, Y.; Chen, S.; Zhang, M.; Zhang, H. Minimizing AoI in Mobile Edge Computing: Nested Index Policy with Preemptive and Non-preemptive Structure. *arXiv preprint arXiv:2508.20564* **2025**.
4. Yang, N.; Wen, J.; Zhang, M.; Tang, M. Generalizable Pareto-Optimal Offloading with Reinforcement Learning in Mobile Edge Computing. *IEEE Transactions on Services Computing* **2025**.
5. Qian, J.; Dong, L.; Shen, Y.; Wei, F.; Chen, W. Controllable Natural Language Generation with Contrastive Prefixes. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022. Association for Computational Linguistics, 2022, pp. 2912–2924. https://doi.org/10.18653/v1/2022.findings-acl.229.
6. Chen, S.; Aguilar, G.; Neves, L.; Solorio, T. Data Augmentation for Cross-Domain Named Entity Recognition. In Proceedings of the Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2021, pp. 5346–5356. https://doi.org/10.18653/v1/2021.emnlp-main.434.
7. Yang, N.; Wang, P.; Liu, G.; Zhang, H.; Lv, P.; Wang, J. Proactive Constrained Policy Optimization with Preemptive Penalty. *arXiv preprint arXiv:2508.01883* **2025**.
8. Zhang, Y.; Feng, S.; Tan, C. Active Example Selection for In-Context Learning. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2022, pp. 9134–9148. https://doi.org/10.18653/v1/2022.emnlp-main.622.
9. Ahmed, T.; Bulathwela, S. Towards Proactive Information Retrieval in Noisy Text with Wikipedia Concepts. In Proceedings of the Proceedings of the CIKM 2022 Workshops co-located with 31st ACM International Conference on Information and Knowledge Management (CIKM 2022), Atlanta, USA, October 17-21, 2022. CEUR-WS.org, 2022.
10. Lu, Y.; Lin, H.; Xu, J.; Han, X.; Tang, J.; Li, A.; Sun, L.; Liao, M.; Chen, S. Text2Event: Controllable Sequence-to-Structure Generation for End-to-end Event Extraction. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Association for Computational Linguistics, 2021, pp. 2795–2806. https://doi.org/10.18653/v1/2021.acl-long.217.
11. Zhang, Z.; Strubell, E.; Hovy, E. A Survey of Active Learning for Natural Language Processing. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2022, pp. 6166–6190. https://doi.org/10.18653/v1/2022.emnlp-main.414.
12. Min, S.; Lewis, M.; Zettlemoyer, L.; Hajishirzi, H. MetaICL: Learning to Learn In Context. In Proceedings of the Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics, 2022, pp. 2791–2809. https://doi.org/10.18653/v1/2022.naacl-main.201.
13. Hu, X.; Zhang, C.; Ma, F.; Liu, C.; Wen, L.; Yu, P.S. Semi-supervised Relation Extraction via Incremental Meta Self-Training. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021. Association for Computational Linguistics, 2021, pp. 487–496. https://doi.org/10.18653/v1/2021.findings-emnlp.44.
14. Zhang, H.; Wang, Y.; Yin, G.; Liu, K.; Liu, Y.; Yu, T. Learning Language-guided Adaptive Hyper-modality Representation for Multimodal Sentiment Analysis. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2023, pp. 756–767. https://doi.org/10.18653/v1/2023.emnlp-main.49.
15. Wang, B.; Che, W.; Wu, D.; Wang, S.; Hu, G.; Liu, T. Dynamic Connected Networks for Chinese Spelling Check. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP

 doi:10.20944/preprints202603.1741.v1

2021. Association for Computational Linguistics, 2021, pp. 2437–2446. https://doi.org/10.18653/v1/2021.findings-acl.216.

16. Wu, C.; Wu, F.; Qi, T.; Huang, Y. Hi-Transformer: Hierarchical Interactive Transformer for Efficient and Effective Long Document Modeling. In Proceedings of the Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). Association for Computational Linguistics, 2021, pp. 848–853. https://doi.org/10.18653/v1/2021.acl-short.107.
17. Wang, P.; Zhu, Z. Overview of Online Parameter Identification of Permanent Magnet Synchronous Machines under Sensorless Control. *IEEE Access* **2026**.
18. Wang, P.; Zhu, Z.; Freire, N.; Azar, Z.; Wu, X.; Liang, D. Online Simultaneous Identification of Multi-Parameters for Interior PMSMs Under Sensorless Control. *CES Transactions on Electrical Machines and Systems* **2025**, *9*, 422–433.
19. Wang, P.; Zhu, Z.; Liang, D.; Freire, N.M.; Azar, Z. Dual signal injection-based online parameter estimation of surface-mounted PMSMs under sensorless control. *IEEE Transactions on Industry Applications* **2025**.
20. Li, X.; Ma, Y.; Ye, K.; Cao, J.; Zhou, M.; Zhou, Y. Hy-facial: Hybrid feature extraction by dimensionality reduction methods for enhanced facial expression classification. *arXiv preprint arXiv:2509.26614* **2025**.