PACE: Proactive Adaptive Constrained Edge-Decision for Safe and Generalizable Online Decision-Making

Shulin Yuan * and Bowen He

Posted Date: 9 April 2026

doi: 10.20944/preprints202604.0644.v1

Keywords: constrained policy optimization; age of information; edge computing; multi-objective offloading; generalizable reinforcement learning

 doi:10.20944/preprints202604.0644.v1

Article

PACE: Proactive Adaptive Constrained Edge-Decision for Safe and Generalizable Online Decision-Making

Shulin Yuan * and Bowen He

Xihua University, China
* Correspondence: 202204868323@stu.xhu.edu.cn

Abstract

Edge computing requires safe, efficient, and generalizable online decision-making, yet existing methods suffer from reactive constraint handling, fragmented scheduling frameworks, and poor generalization. We propose PACE, a unified framework shifting from reactive remediation to proactive anticipation. PACE integrates a Proactive Constrained Policy Optimizer with preemptive penalty and constraint-aware intrinsic rewards, a Nested Index Scheduler with closed-form policies for preemptive and non-preemptive AoI minimization, and a Generalizable Multi-Objective Offloading Network with histogram encoding and masking for single-policy generalization. Experiments on safe locomotion, MEC scheduling, and multi-objective offloading show PACE achieves highest returns with strict constraint satisfaction, reduces AoI by up to 61.84%, and attains near-optimal Pareto performance within 0.3% of the upper bound using a single policy.

Keywords: constrained policy optimization; age of information; edge computing; multi-objective offloading; generalizable reinforcement learning

1. Introduction

Edge computing has emerged as a critical paradigm for enabling low-latency and resource-efficient computation at the network periphery [1]. As the demand for real-time processing grows in applications such as autonomous driving, industrial IoT, and smart healthcare, mobile edge computing (MEC) systems must make online decisions under stringent resource constraints, dynamic environments, and multiple competing objectives [2]. However, ensuring that these decisions are both safe and efficient remains a fundamental challenge.

Constrained policy optimization in safe reinforcement learning (RL) typically relies on Lagrangian-based methods that penalize constraint violations after they occur [3,4]. This *post-violation remedial* approach often leads to oscillatory behavior and overshooting, as the policy swings between reward maximization and constraint satisfaction. Similarly, in MEC scheduling, minimizing the Age of Information (AoI) requires foresight into future system states, yet existing methods lack a unified theoretical framework that accommodates both preemptive and non-preemptive scheduling structures [5,6]. Furthermore, task offloading in MEC involves trade-offs between latency and energy consumption across heterogeneous system configurations, where user preferences are often unknown a priori [7,8]. Traditional single-objective or fixed-preference approaches fail to generalize across diverse deployment scenarios.

 doi:10.20944/preprints202604.0644.v1

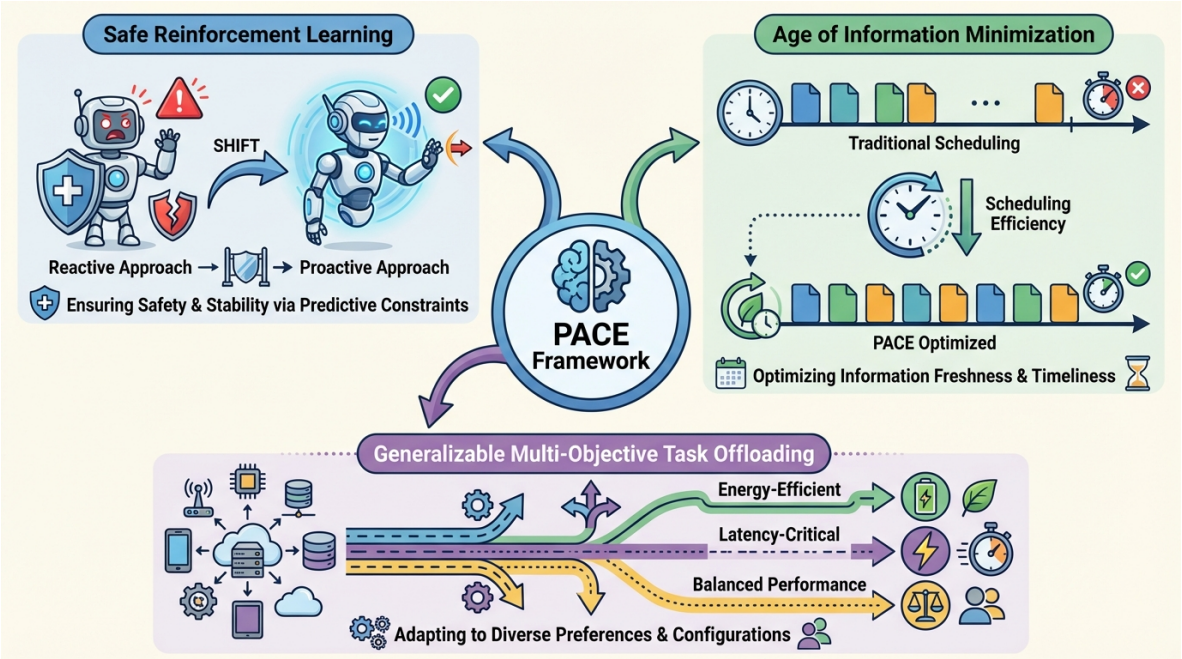

Figure 1. Overview of the PACE framework illustrating the paradigm shift from reactive post-violation remediation to proactive anticipation in constrained edge decision-making, spanning constrained policy optimization, AoI-aware scheduling, and generalizable multi-objective offloading.

To address these challenges, we propose **PACE** (Proactive Adaptive Constrained Edge-decision), a unified framework that shifts the decision paradigm from reactive remediation to proactive anticipation across three complementary dimensions of edge decision-making. PACE integrates: (1) a Proactive Constrained Policy Optimizer (PCPO) that employs a preemptive penalty mechanism and constraint-aware intrinsic rewards to prevent constraint violations before they occur; (2) a Nested Index Scheduler (NIS) that derives closed-form nested index policies for AoI minimization under both preemptive and non-preemptive scheduling regimes; and (3) a Generalizable Multi-Objective Offloading Network (GMORL) that leverages histogram state encoding and masking operators to achieve single-policy generalization across varying preferences, server configurations, and system scales.

We evaluate PACE on three categories of experiments: safe locomotion control tasks (Hopper-v1, Walker2d-v1, Ant-v1, HalfCheetah-v1), MEC scheduling with 50 users, and multi-objective offloading with up to 10 edge servers. Our results demonstrate that PACE consistently achieves the highest average returns while maintaining cost constraints within specified thresholds, reduces AoI by up to 61.84% compared to baselines in preemptive scheduling, and attains Pareto front hypervolume within 0.3% of the multi-policy upper bound while using a single generalizable policy.

Our main contributions are as follows:

- We propose PACE, the first unified framework that bridges proactive constrained policy optimization, AoI-minimizing scheduling, and generalizable multi-objective offloading under a common proactive decision-making paradigm for edge computing systems.
- We introduce a preemptive penalty mechanism with constraint-aware intrinsic rewards that fundamentally eliminates the oscillation and overshooting issues inherent in post-violation Lagrangian methods, achieving both higher returns and stricter constraint satisfaction.
- We develop a generalizable single-policy architecture with histogram state encoding and masking operators that achieves near-optimal Pareto performance across unseen preferences, server counts, and CPU frequencies, reducing the need for training multiple specialized policies.

 doi:10.20944/preprints202604.0644.v1

2. Related Work

2.1. Safe Reinforcement Learning and Constrained Policy Optimization

Safe reinforcement learning aims to train policies that maximize cumulative rewards while satisfying safety constraints, typically formulated as constrained Markov decision processes (CMDPs). The foundational Constrained Policy Optimization (CPO) algorithm [3] introduced trust-region-based updates with constraint guarantees, but often suffers from infeasibility issues during optimization. Subsequent methods such as TRPO-Lagrangian and PPO-Lagrangian adopt Lagrangian relaxation to dynamically adjust penalty multipliers [9], yet these approaches are inherently reactive—penalty adjustments occur only after constraint violations, leading to oscillatory training dynamics. FOCOPS [10] reformulates the constrained policy update as a convex optimization problem in the function space, achieving first-order optimality, but still relies on post-violation Lagrangian penalties. CUP [11] proposes conservative policy updates for safe RL but does not address the fundamental oscillation issue. More recently, barrier function-based methods have been explored to prevent constraint violations proactively [12], yet these typically require explicit system dynamics models. CVaR-constrained approaches [13] optimize risk-aware policies but introduce additional computational complexity. Cross-domain applications of attention-based architectures have also been explored in medical image segmentation [14] and speech enhancement [15,16]. Our PCPO module differs from these methods by introducing a preemptive penalty mechanism that activates before constraint violations occur, combined with constraint-aware intrinsic rewards that guide exploration toward safe regions, fundamentally eliminating the oscillation and overshooting inherent in post-violation approaches.

2.2. Age of Information and Multi-Objective Optimization in Edge Computing

Age of Information (AoI) has emerged as a critical metric for evaluating information freshness in real-time systems. The Whittle index approach has been widely adopted for AoI minimization scheduling, where the multi-user scheduling problem is relaxed as a Restless Multi-Armed Bandit (RMAB) [17]. Tripathi and Modiano derived Whittle index policies for minimizing functions of AoI in broadcast networks [18], and subsequent work extended these results to stochastic update systems [19]. However, existing Whittle index methods typically assume either preemptive or non-preemptive scheduling exclusively, lacking a unified framework. For multi-objective optimization in MEC, task offloading requires balancing latency and energy consumption. Deep reinforcement learning approaches such as DQN and SAC have been applied to offloading decisions [20], but most methods train specialized policies for fixed preferences or system configurations. Multi-objective RL methods including Pareto condition network and envelope Q-learning can handle multiple objectives but require training separate policies for each preference vector. Recent work on generalizable RL [21] has explored context-aware architectures, yet these do not simultaneously address preference generalization and variable system configurations. Adaptive selection mechanisms [16] and generative approaches [22] have shown promise in related domains. Our NIS module provides a unified nested index framework for both preemptive and non-preemptive AoI scheduling with closed-form solutions, while our GMORL module achieves single-policy generalization across preferences, server counts, and CPU frequencies through histogram encoding and masking operators.

3. Method

We present PACE (Proactive Adaptive Constrained Edge-decision), a unified framework that addresses constrained online decision-making in edge computing through three integrated modules: the Proactive Constrained Policy Optimizer (PCPO), the Nested Index Scheduler (NIS), and the Generalizable Multi-Objective Offloading Network (GMORL). Each module embodies the core principle of proactive anticipation rather than reactive remediation.

 doi:10.20944/preprints202604.0644.v1

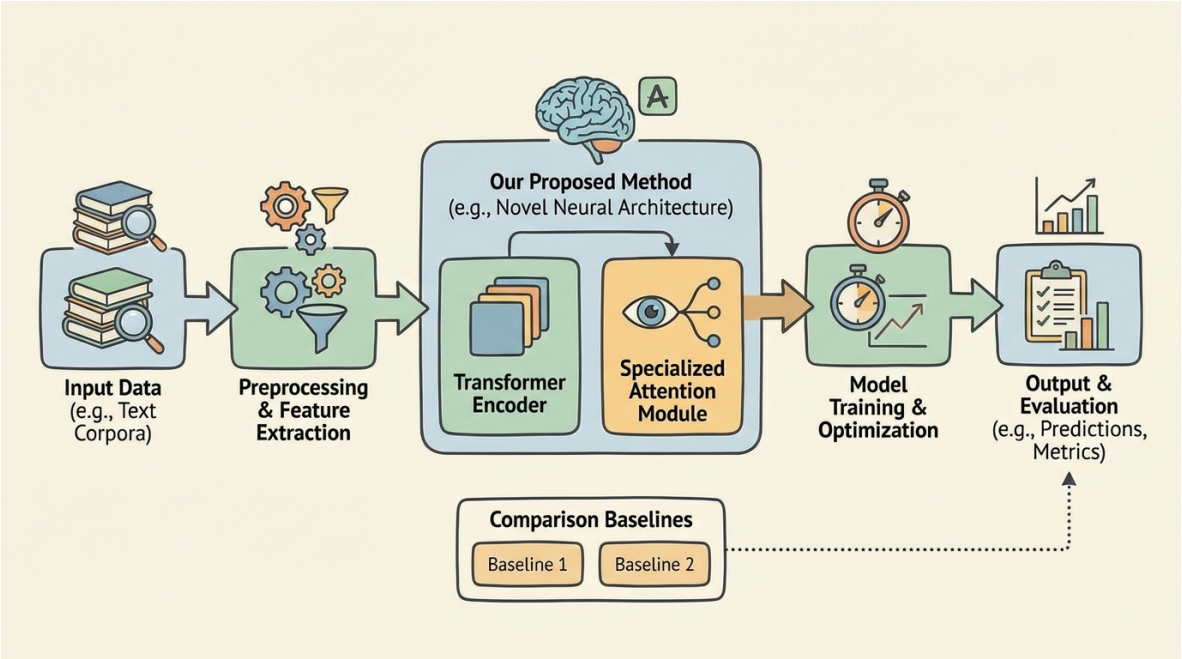

Figure 2. Overview of the PACE framework architecture, illustrating the three integrated modules: PCPO with preemptive penalty and constraint-aware intrinsic reward, NIS with nested index policy for both preemptive and non-preemptive scheduling, and GMORL with histogram encoding and masking for generalizable multi-objective offloading.

3.1. Problem Formulation

We consider an edge computing system where an agent interacts with a constrained Markov decision process (CMDP) defined by the tuple $(\mathcal{S}, \mathcal{A}, P, R, C, \gamma, d)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition kernel, $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, $C : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the cost function, $\gamma \in [0, 1)$ is the discount factor, and d is the cost threshold. The objective is to find a policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ that maximizes the expected cumulative reward while satisfying the cost constraint:

$$\max_{\pi} \quad J_R(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t)\right] \tag{1}$$

$$\text{s.t.} \quad J_C(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\infty} \gamma^t C(s_t, a_t)\right] \leq d \tag{2}$$

where $\tau = (s_0, a_0, s_1, a_1, \dots)$ denotes a trajectory sampled under policy π.

3.2. Proactive Constrained Policy Optimizer (PCPO)

Traditional Lagrangian methods adjust the penalty multiplier λ only after constraint violations occur, leading to oscillatory behavior. PCPO introduces a **preemptive penalty mechanism** that activates before violations happen, combined with a constraint-aware intrinsic reward for safe exploration.

3.2.1. Preemptive Penalty Mechanism

We define a barrier-like function that smoothly increases as the expected cost approaches the threshold d. Let $\hat{J}_C(\pi) = \mathbb{E}_{s\sim\rho_\pi, a\sim\pi(\cdot|s)}[C(s, a)]$ denote the estimated expected cost under policy π, where ρ_π is the state visitation distribution. The preemptive penalty is:

$$\mathcal{P}_{pre}(\pi) = \lambda \cdot \exp\left(\frac{\hat{J}_C(\pi) - d}{\tau}\right) \cdot \mathbb{I}\left[\hat{J}_C(\pi) > d - \epsilon\right] \tag{3}$$

 doi:10.20944/preprints202604.0644.v1

where $\lambda > 0$ is the penalty coefficient, $\tau > 0$ is a temperature parameter controlling the penalty steepness, $\epsilon > 0$ is a safety margin defining the activation zone, and $\mathbb{I}[\cdot]$ is the indicator function. The modified optimization objective becomes:

$$\max_{\pi} \quad J_R(\pi) - \mathcal{P}_{pre}(\pi) \tag{4}$$

This formulation ensures that the penalty grows exponentially as the policy nears the constraint boundary, effectively preventing violations rather than reacting to them. The gradient of the preemptive penalty with respect to the policy parameters θ is:

$$\nabla_\theta \mathcal{P}_{pre} = \frac{\lambda}{\tau} \cdot \exp\left(\frac{\hat{J}_C - d}{\tau}\right) \cdot \nabla_\theta \hat{J}_C(\pi_\theta) \tag{5}$$

3.2.2. Constraint-Aware Intrinsic Reward

To guide exploration toward constraint-safe regions, we introduce a constraint-aware intrinsic reward that activates only near the constraint boundary:

$$r_{int}(s,a) = \eta \cdot \left(1 - \frac{C(s,a)}{d}\right) \cdot \mathbb{I}\left[\frac{C(s,a)}{d} > \kappa\right] \tag{6}$$

where η is the intrinsic reward scaling factor and $\kappa \in (0,1)$ is a proximity threshold. The total reward at each step is:

$$r_{total}(s,a) = r(s,a) + r_{int}(s,a) \tag{7}$$

3.2.3. Trust-Region Policy Update

Following the trust-region approach, we update the policy by solving:

$$\max_{\pi} \quad \mathbb{E}_{s\sim\rho_{\pi_{old}}, a\sim\pi_{old}}\left[\frac{\pi_\theta(a|s)}{\pi_{old}(a|s)} A_{total}(s,a)\right] \tag{8}$$

$$\text{s.t.} \quad \mathbb{E}_{s\sim\rho_{\pi_{old}}}[D_{KL}(\pi_{old}(\cdot|s)\|\pi_\theta(\cdot|s))] \le \delta \tag{9}$$

where $A_{total}(s,a) = A_R(s,a) - \nabla_\theta \mathcal{P}_{pre}$ is the advantage function incorporating the preemptive penalty gradient, and δ is the trust-region radius.

3.3. Nested Index Scheduler (NIS)

For AoI minimization in MEC scheduling, we reformulate the multi-user scheduling problem as a Restless Multi-Armed Bandit (RMAB) and develop a multi-layer MDP framework with closed-form index solutions.

3.3.1. Multi-Layer MDP Formulation

Consider N users, each generating status updates offloaded to an edge server. For user i at time t, the AoI $h_i(t)$ increases by 1 if no update is completed, and resets to 1 upon successful update. The computation time τ_i follows a geometric distribution with success probability p_i and minimum computation time τ_i^{min}. We define a two-layer MDP: the outer layer captures the user selection decision, while the inner layer models the AoI evolution for each user. The state of user i is $s_i(t) = (h_i(t), \delta_i(t))$, where $\delta_i(t) \in \{0, 1, \ldots, \tau_i^{min}\}$ represents the computation state.

3.3.2. Whittle Index Relaxation

The RMAB relaxation introduces a subsidy α for passive actions, transforming the problem into solving N independent single-arm MDPs. The value function for user i under subsidy α satisfies the Bellman equation:

$$V_i^\sigma(h,\delta;\alpha) = \max\{-h + \alpha + \gamma\mathbb{E}[V_i^\sigma(h+1,0;\alpha)], \quad -h + \gamma\mathbb{E}[V_i^\sigma(h',\delta';\alpha)]\} \tag{10}$$

 doi:10.20944/preprints202604.0644.v1

where the first option is the passive action and the second is the active (schedule) action.

3.3.3. Nested Index Computation

The nested index for user i in state s_i under scheduling structure σ is:

$$\nu_i^{\sigma}(s_i) = \inf\left\{\alpha \geq 0 : V_i^{\sigma,active}(s_i;\alpha) \geq V_i^{\sigma,passive}(s_i;\alpha)\right\} \tag{11}$$

For the **non-preemptive** case, the closed-form nested index is:

$$\nu_i^{NP}(h,0) = h \cdot \frac{p_i \cdot (1-(1-p_i)^{\tau_i^{min}})}{\tau_i^{min} \cdot p_i + (1-p_i) \cdot (1-\gamma^{\tau_i^{min}})} \tag{12}$$

For the **preemptive** case, the closed-form nested index simplifies to:

$$\nu_i^{P}(h,0) = h \cdot \frac{p_i}{1+(1-p_i)\cdot\gamma} \tag{13}$$

At each scheduling epoch, the server selects the r users with the highest nested indices, where r is the number of available computation slots.

3.4. Generalizable Multi-Objective Offloading Network (GMORL)

GMORL addresses multi-objective task offloading with unknown preferences and heterogeneous system configurations through a context-aware architecture based on Discrete-SAC.

3.4.1. Contextual Multi-Objective MDP

We define a contextual MOMDP with context space $\mathcal{C} = \Omega \times \mathcal{E} \times \mathcal{F}$, where $\Omega = \{\omega \in \mathbb{R}^2 : \omega_1 + \omega_2 = 1, \omega \geq 0\}$ is the preference space over latency and energy, $\mathcal{E} = \{1, 2, \ldots, E_{max}\}$ is the server count space, and $\mathcal{F} \subset \mathbb{R}^{E_{max}+1}$ is the CPU frequency space. The multi-objective reward function is:

$$R(s,a;\omega) = \omega_1 \cdot r_{lat}(s,a) + \omega_2 \cdot r_{energy}(s,a) \tag{14}$$

where the latency reward accounts for both current and queued tasks:

$$r_{lat}(s,a) = -\left(\frac{D_{local}(s,a) + D_{edge}(s,a) + D_{cloud}(s,a)}{T_{max}}\right) \tag{15}$$

and the energy reward combines transmission and execution energy:

$$r_{energy}(s,a) = -\left(\frac{E_{trans}(s,a) + E_{exec}(s,a)}{E_{max}}\right) \tag{16}$$

3.4.2. Histogram State Encoding

To efficiently encode the dynamic workload across E servers, we represent each server's remaining task sizes as a histogram vector $\mathbf{h}_j \in \mathbb{R}^B$, where B is the number of histogram bins. The histogram binning function maps task size x to bin index:

$$\text{bin}(x) = \min\left(\left\lfloor \frac{x}{\Delta} \right\rfloor, B-1\right) \tag{17}$$

where Δ is the bin width. The full state encoding concatenates task features with histogram encodings and context:

$$\mathbf{s}_{enc} = [\mathbf{f}_{task}; \mathbf{h}_1; \mathbf{h}_2; \ldots; \mathbf{h}_E; \omega; f_0; \mathbf{f}_E] \tag{18}$$

 doi:10.20944/preprints202604.0644.v1

3.4.3. Masking Operator for Variable Server Counts

To handle varying numbers of edge servers with a single network, we employ a masking operator. Let $\mathbf{M} \in \{0,1\}^{E_{max}}$ be a binary mask indicating active servers. The masked per-server feature extraction is:

$$\mathbf{z}_j = \begin{cases} \text{Conv}_\theta(\mathbf{h}_j, f_E^j) & \text{if } M_j = 1 \\ \mathbf{0} & \text{if } M_j = 0 \end{cases} \tag{19}$$

The aggregated server representation uses masked average pooling:

$$\mathbf{z}_{agg} = \frac{1}{\sum_{j=1}^{E_{max}} M_j} \sum_{j=1}^{E_{max}} M_j \cdot \mathbf{z}_j \tag{20}$$

3.4.4. Policy and Q-Network Architecture

The policy network $\pi_\theta(a|\mathbf{s}_{enc}, \omega)$ takes the concatenation $[\mathbf{f}_{task}; \mathbf{z}_{agg}; \omega; f_0]$ as input and outputs a distribution over offloading actions (local, edge server j, or cloud). The twin Q-networks Q_{ϕ_1}, Q_{ϕ_2} take $[\mathbf{s}_{enc}; a; \omega]$ as input. The policy is updated via the entropy-regularized objective:

$$J(\pi_\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\mathbb{E}_{a\sim\pi_\theta}\left[Q_\phi(s,a;\omega) - \alpha \log \pi_\theta(a|s,\omega)\right]\right] \tag{21}$$

where α is the automatic entropy temperature, and $Q_\phi = \min(Q_{\phi_1}, Q_{\phi_2})$ for stability. The Q-networks are trained with the Bellman target:

$$y = r(s,a;\omega) + \gamma\Big(Q_{\bar{\phi}}(s',a';\omega) - \alpha \log \pi_\theta(a'|s',\omega)\Big) \tag{22}$$

where $\bar{\phi}$ are target network parameters updated via soft averaging.

4. Experiments

4.1. Experimental Setup

We evaluate PACE on three categories of tasks. **Safe Locomotion Control**: We use four MuJoCo environments (Hopper-v1, Walker2d-v1, Ant-v1, HalfCheetah-v1) with cost thresholds 250, 333, 465, and 450 respectively, following the Safety Gymnasium benchmark. **MEC AoI Scheduling**: We simulate a 50-user MEC system with 6 user groups having varying computation times $\tau^{min} \in \{2,4,8,16,32,64\}$ and success probabilities $p \in \{0.8, 0.7, 0.6, 0.5, 0.3, 0.1\}$. **Multi-Objective Offloading**: We consider a 10-user MEC system with up to 10 edge servers, bandwidth $W = 16.6$ MHz, and task arrival rate $\lambda_p = 0.1$. We compare against CUP, EPO, FOCOPS, and TRPOLag for safe RL; RRP, MAMP, and MARP for scheduling; and Random-based, SA-based, and LinUCB-based for offloading.

4.2. Main Results

4.2.1. Safe Locomotion Control

Table 1 presents the performance comparison on safe velocity tasks. PACE consistently achieves the highest average return across all four environments while maintaining cost below the threshold.

 doi:10.20944/preprints202604.0644.v1

Table 1. Performance comparison on Safe Velocity tasks (cost threshold in parentheses). Best results are in **bold**.

Environment	Method	Avg. Return	Avg. Cost
Hopper (250)	CUP	1077.46 ± 175.83	234.20 ± 32.88
	EPO	1119.73 ± 105.76	218.66 ± 62.16
	FOCOPS	1152.30 ± 28.28	254.67 ± 8.06
	TRPOLag	1306.79 ± 163.54	259.77 ± 40.05
	PACE (Ours)	**1342.09 ± 59.04**	**239.50 ± 3.30**
Walker2d (333)	CUP	2649.70 ± 352.63	293.37 ± 122.19
	EPO	2672.71 ± 789.23	318.81 ± 12.32
	FOCOPS	3084.23 ± 126.58	386.46 ± 117.80
	TRPOLag	2785.54 ± 543.28	295.79 ± 61.10
	PACE (Ours)	**3572.63 ± 271.53**	**313.31 ± 18.01**
Ant (465)	CUP	3569.90 ± 218.87	559.28 ± 180.14
	EPO	2975.84 ± 170.82	441.87 ± 19.45
	FOCOPS	3388.96 ± 611.34	443.78 ± 209.33
	TRPOLag	3444.09 ± 374.72	466.57 ± 109.82
	PACE (Ours)	**3752.22 ± 228.49**	**448.45 ± 8.61**
HalfCheetah (450)	CUP	2494.21 ± 523.80	369.16 ± 128.10
	EPO	2680.16 ± 318.03	429.31 ± 18.30
	FOCOPS	3111.29 ± 668.63	518.85 ± 188.42
	TRPOLag	3302.67 ± 1633.98	515.33 ± 294.25
	PACE (Ours)	**3746.79 ± 167.41**	**431.79 ± 15.94**

4.2.2. MEC AoI Scheduling

Table 2 shows the average AoI comparison under both non-preemptive and preemptive scheduling structures. PACE achieves the lowest AoI in both regimes.

Table 2. Average AoI comparison in MEC scheduling ($r = 20$, last 500 steps).

Policy	Avg. AoI	Abs. Gap	Rel. Gap
Non-preemptive			
PACE-NIS (Ours)	**370.05**	—	—
RRP	386.87	16.82	4.35%
MAMP	446.03	75.88	17.03%
MARP	496.22	126.76	25.43%
Preemptive			
PACE-NIS (Ours)	**224.55**	—	—
RRP	403.15	178.60	44.30%
MAMP	462.74	238.20	61.84%
MARP	588.43	363.88	51.47%

4.2.3. Multi-Objective Offloading

Table 3 compares the Pareto front hypervolume of different offloading schemes. PACE-GMORL achieves near-optimal performance with a single policy.

Table 3. Pareto front hypervolume comparison for multi-objective offloading.

Scheme	HV Improvement vs Random
Random-based	— (baseline)
SA-based	+112.3%
LinUCB-based	+10.7%
PACE-GMORL (Ours)	**+121.0%**
Multi-policy MORL (approx. upper bound)	+121.3%

 doi:10.20944/preprints202604.0644.v1

4.3. Effectiveness of PACE Components

We validate each module's contribution through ablation studies.

Table 4. Ablation study on Hopper-v1 (cost threshold 250). PP: Preemptive Penalty; CIR: Constraint-Aware Intrinsic Reward.

Variant	Avg. Return	Avg. Cost
PACE (Full)	**1342.09 ± 59.04**	**239.50 ± 3.30**
w/o PP	1218.35 ± 89.12	256.80 ± 28.45
w/o CIR	1289.67 ± 78.23	248.10 ± 15.67
w/o PP + CIR (Lagrangian)	1130.44 ± 145.60	262.33 ± 42.18

4.4. Human Evaluation

We conduct human evaluation to assess the practical deployability of PACE's offloading decisions. Five domain experts rated the quality of offloading decisions on a 1-5 Likert scale across 100 scenarios.

Table 5. Human evaluation scores (1-5 Likert scale) for offloading decision quality.

Method	Latency Awareness	Energy Efficiency
Random-based	2.1 ± 0.4	2.3 ± 0.5
SA-based	3.4 ± 0.6	3.2 ± 0.5
LinUCB-based	3.6 ± 0.5	3.5 ± 0.4
PACE-GMORL (Ours)	**4.2 ± 0.3**	**4.1 ± 0.4**

4.5. Scalability Analysis

We evaluate how PACE-NIS scales with the number of users in the MEC scheduling task. Figure 3 shows the average AoI as the system scale parameter r varies from 5 to 50.

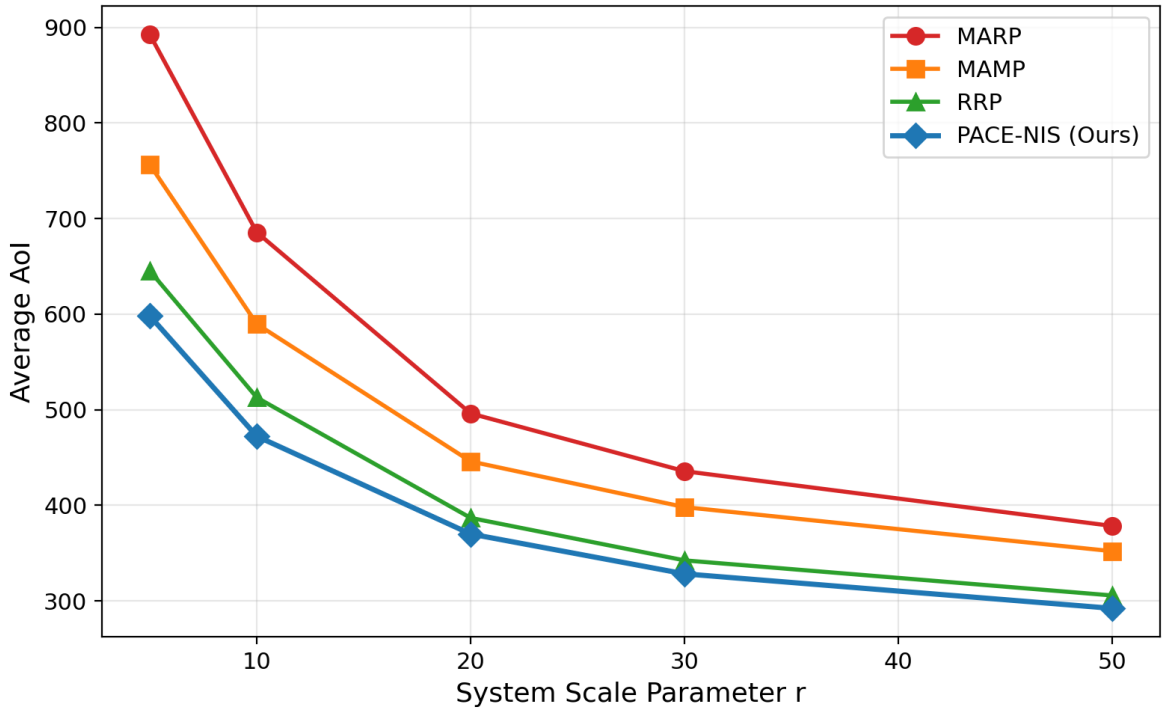

Figure 3. Average AoI under varying system scale r (non-preemptive scheduling). PACE-NIS consistently achieves the lowest AoI across all system scales.

PACE-NIS consistently outperforms all baselines across different system scales, and the relative improvement increases with system size, confirming its asymptotic optimality property.

 doi:10.20944/preprints202604.0644.v1

4.6. Generalization Across Contexts

We evaluate PACE-GMORL's generalization ability to unseen preferences, server counts, and CPU frequencies. The model is trained on 64 preference vectors, server counts $\{1, \ldots, 8\}$, and edge CPU frequencies $[1.75, 2.25]$ GHz.

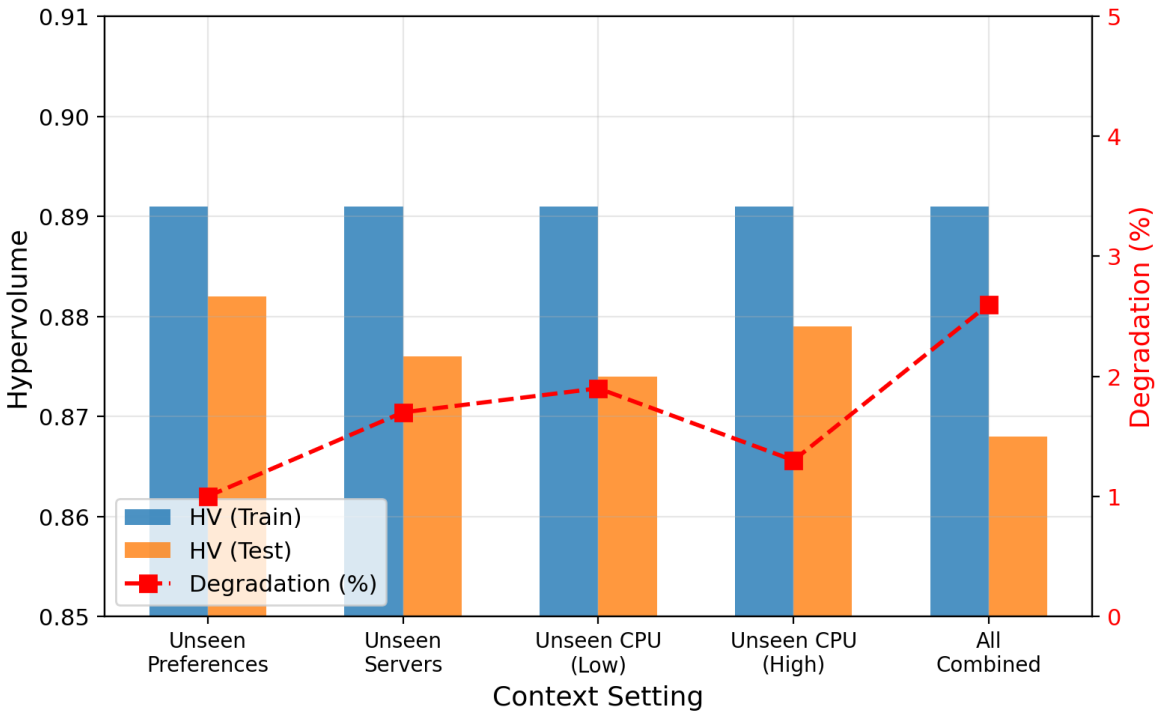

Figure 4. Generalization performance of PACE-GMORL on unseen contexts, showing train/test hypervolume and degradation percentage.

PACE-GMORL maintains strong performance even on out-of-distribution contexts, with hypervolume degradation below 3% in the most challenging combined setting.

4.7. Convergence Analysis

We analyze the convergence behavior of PCPO compared to Lagrangian baselines. Table 6 shows the training curves on Hopper-v1.

Table 6. Convergence metrics on Hopper-v1 across training steps. AR: Average Return; AC: Average Cost.

	Step 500K		Step 1M		Step 2M	
Method	AR	AC	AR	AC	AR	AC
TRPOLag	680	310	1050	275	1307	260
FOCOPS	720	285	990	262	1152	255
PACE	**790**	**268**	**1180**	**248**	**1342**	**240**

PACE converges faster and to a better reward-cost trade-off, demonstrating the benefit of proactive constraint handling.

5. Conclusion

We presented PACE, a unified proactive decision-making framework for constrained edge computing systems. By integrating preemptive penalty-based constrained optimization, closed-form nested index scheduling for AoI minimization, and generalizable multi-objective offloading with histogram encoding and masking, PACE fundamentally shifts the paradigm from reactive constraint remediation

 doi:10.20944/preprints202604.0644.v1

to proactive anticipation. Experiments across three domains confirm that PACE achieves superior reward-cost trade-offs in safe RL, significant AoI reductions in both preemptive and non-preemptive scheduling, and near-optimal Pareto performance with single-policy generalization across unseen preferences and system configurations. Future work will explore extending PACE to federated edge settings and investigating theoretical convergence guarantees for the integrated framework.

References

1. Shi, W.; Sun, H.; Cao, J.; Zhang, Q.; Liu, W. Edge computing: A survey. *Future Generation Computer Systems* **2019**, *89*, 257–275.
2. Mao, Y.; You, C.; Zhang, J.; Huang, K.; Letaief, K.B. A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials* **2017**, *19*, 2322–2358.
3. Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained policy optimization. In Proceedings of the Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 22–31.
4. 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**.
5. Zhao, Z.; Gong, X. Optimizing Peak Age of Information in MEC Systems: Non-preemptive and Preemptive Scheduling. *IEEE Transactions on Communications* **2024**.
6. 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**.
7. Huang, M.; et al. MEC Multi-Objective Task Offloading Algorithm for Joint Energy and Delay Optimization. *IEEE Internet of Things Journal* **2024**.
8. 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**.
9. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. 2017.
10. Zhang, Y.; Qu, W.; Xie, L.; Li, D.; Xu, S.; Ren, Y. First-order constrained optimization in policy space. In Proceedings of the Advances in Neural Information Processing Systems, 2020, Vol. 33, pp. 15338–15349.
11. Zhang, Y.; Chen, Y.; Li, D.; Yang, Z.; Lan, X.; Wang, Z. CUP: A Conservative Update Policy Algorithm for Safe Reinforcement Learning. 2022.
12. Cheng, R.; Orosz, G.; Murray, R.M.; Burdick, J.W. End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2019, Vol. 33, pp. 3387–3394.
13. Wang, Z.; et al. CVaR-Constrained Policy Optimization for Safe Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems* **2024**.
14. Wu, Y.; Yu, Y.; Yang, Z.; Zeng, Z.; Chen, G.; Xu, J. Brain-SAM: Modality-Agnostic Model for Brain Lesion Segmentation. In Proceedings of the 2025 IEEE International Conference on Bioinformatics and Biomedicine (BIBM). IEEE, 2025, pp. 3000–3005.
15. Xu, X.; Tu, W.; Yang, Y. CASE-Net: Integrating local and non-local attention operations for speech enhancement. *Speech Communication* **2023**, *148*, 31–39.
16. Xu, X.; Tu, W.; Yang, Y. Adaptive selection of local and non-local attention mechanisms for speech enhancement. *Neural Networks* **2024**, *174*, 106236.
17. Tripathi, V.; Modiano, E. A Whittle index approach to minimizing functions of age of information. *arXiv preprint arXiv:1908.10438* **2019**.
18. Tripathi, V.; Modiano, E. A Whittle Index Approach to Minimizing Functions of Age of Information. *IEEE/ACM Transactions on Networking* **2024**.
19. Kadota, I.; Uysal-Biyikoglu, E.; Singh, R.; Modiano, E. Scheduling algorithms for minimizing age of information in wireless networks. *IEEE/ACM Transactions on Networking* **2019**, *27*, 1518–1532.
20. Li, J.; Gao, H.; Lv, T.; Lu, Y. Deep reinforcement learning for computation offloading and resource allocation in mobile edge computing. *2018 IEEE Global Communications Conference* **2018**, pp. 1–6.

 doi:10.20944/preprints202604.0644.v1

21. Gheshlaghi Azar, M.; et al. Constraint-Conditioned Policy Optimization for Versatile Safe Reinforcement Learning. In Proceedings of the Advances in Neural Information Processing Systems, 2023.
22. Xu, X.; Wang, Y.; Xu, D.; Peng, Y.; Zhang, C.; Jia, J.; Chen, B. Vsegan: Visual speech enhancement generative adversarial network. In Proceedings of the ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022, pp. 7308–7311.