RESCUE: Opportunistic Online Scheduling of Model Retraining on Underutilized Edges

Jianping Huang, Xiang Liu, Feng Shan

Southeast University

Email: shanfeng@seu.edu.cn

INFOCOM 2026

Presenter: Feng Shan

Outline

1 Introduction

2 Problem

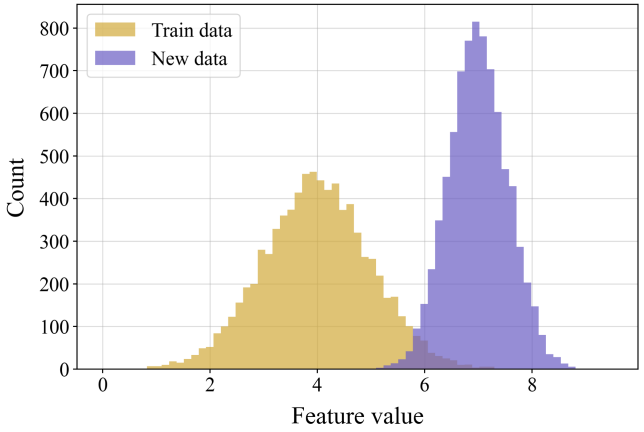
3 Algorithm

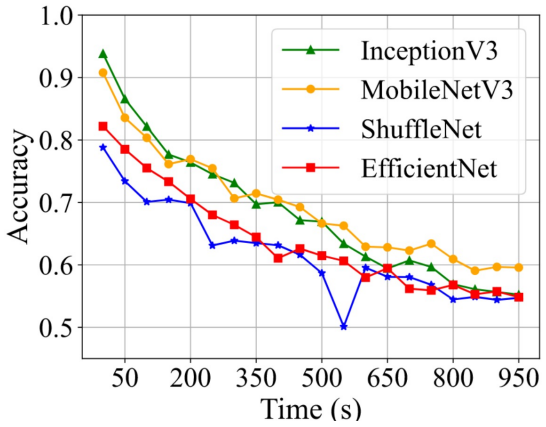
4 Simulation

5 Conclusion

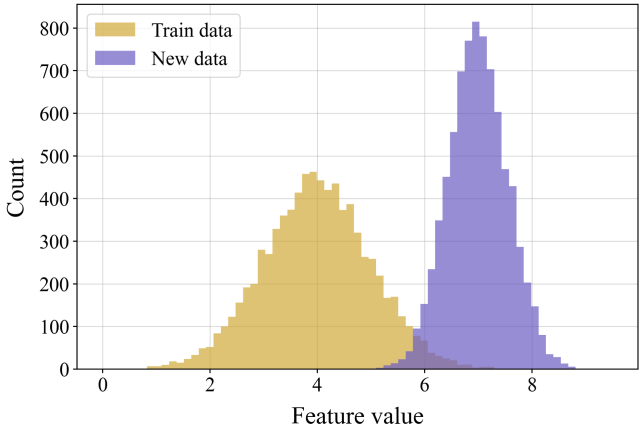
Proliferation of AI on Edge Devices

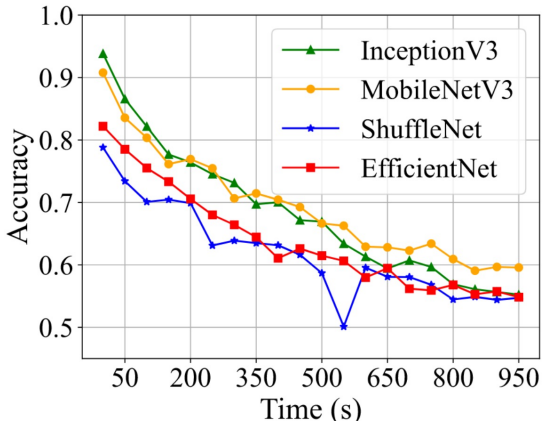
Vulnerability to Data Drift



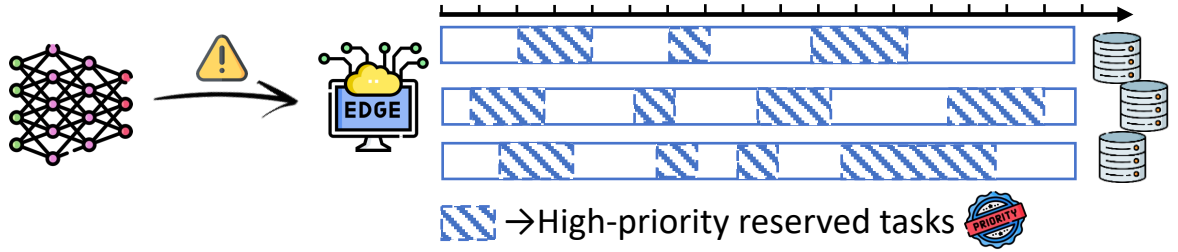


Vulnerability to Data Drift

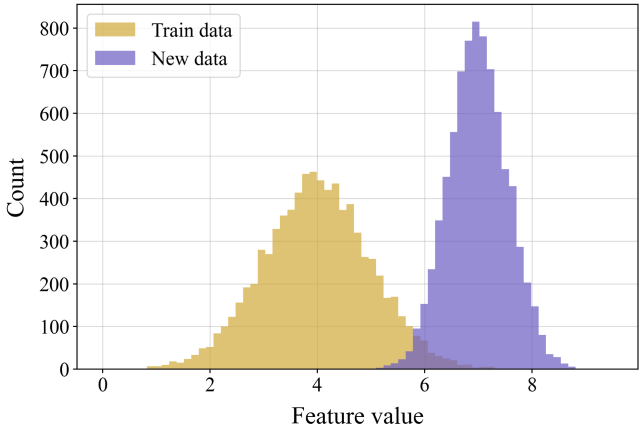


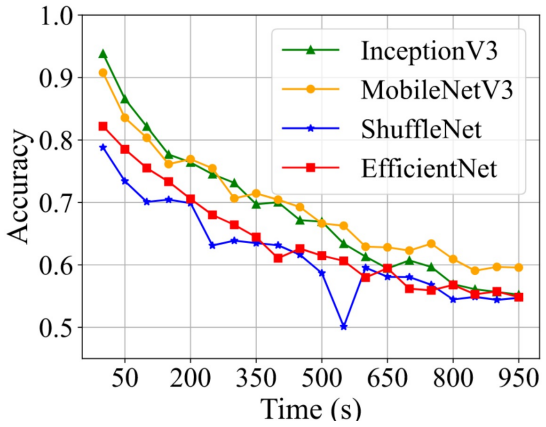


Resource Contention in Retraining

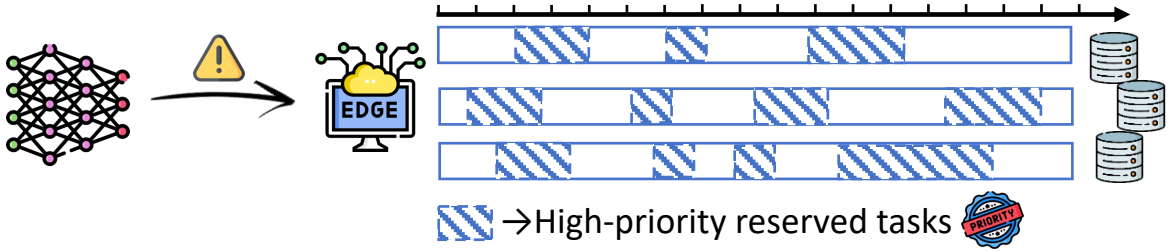


Vulnerability to Data Drift



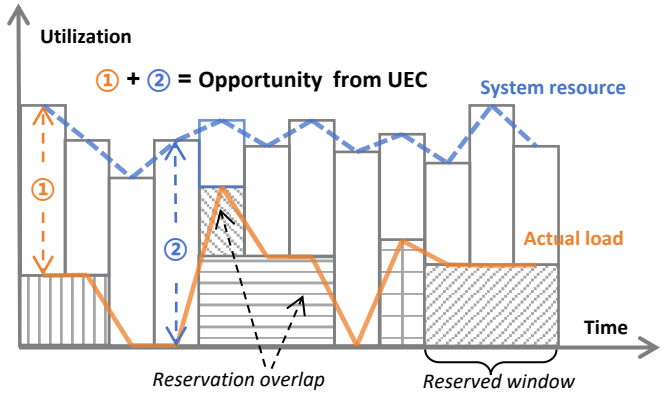


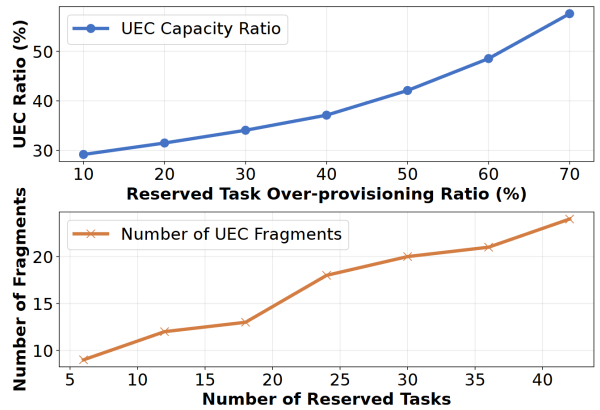
Resource Contention in Retraining



Current Methods Limitation

- ✗ Dedicated resource assumption
- ✗ Poor gap-filling due to fragmented resource
- ✗ Heuristic-based is unreliable in online arrival

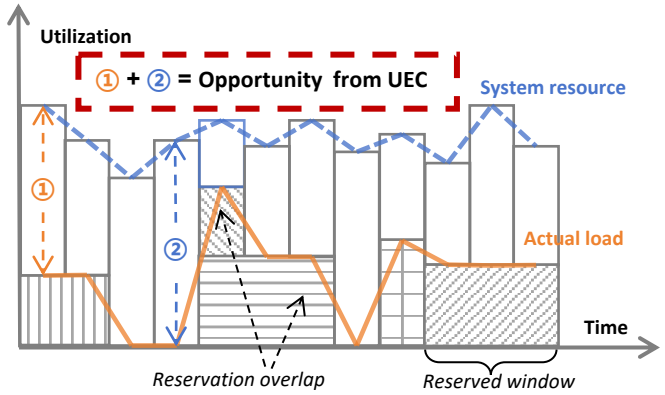


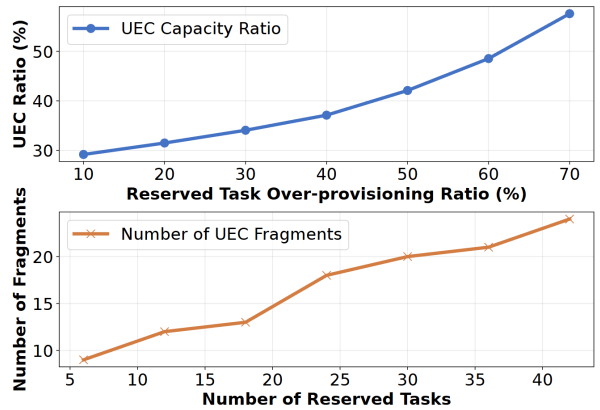


Underutilized Edge Computing (UEC)

Key Observation

- ✓ **UEC** is the dynamic resource **gaps** between static reservations and actual load of reserved tasks.
- ✓ Reserved tasks lead to **over-provisioning** (for QoS) and **resource fragmentation** (idle gaps between tasks).





Underutilized Edge Computing (UEC)

Key Observation

- ✓ Reserved tasks lead to **over-provisioning** (for QoS) and **resource fragmentation** (idle gaps between tasks).
- ✓ **UEC** is the dynamic resource **gaps** between static reservations and actual load of reserved tasks.

Core Question

- ✓ *Can we systematically transform **UEC** "wasteland" into "fertile ground" for **model retraining**?*

Outline

1. Introduction
2. **Problem**
3. Algorithm
4. Simulation
5. Conclusion

Problem Objective

Design an **online scheduling** framework to maximize the **long-term expected profit** from retraining tasks.

Constraints

- Guarantee allocation for reserved tasks
- Schedule opportunistic retraining jobs
- Make real-time, irrevocable decisions

Value Prop

- ✓ Enhance edge efficiency
- ✓ Cost reduction
- ✓ Sustainability

Problem Objective

Design an **online scheduling** framework to maximize the **long-term expected profit** from retraining tasks.

Constraints

- Guarantee allocation for reserved tasks
- Schedule opportunistic retraining jobs
- Make real-time, irrevocable decisions

Value Prop

- ✓ Enhance edge efficiency
- ✓ Cost reduction
- ✓ sustainability

Challenges

Real-time, **Irrevocable** Decisions under **Uncertainty**

- Unknown future tasks, stochastic retraining durations, fluctuating UEC availability.
- Trade off immediate profit from current tasks vs. expected value of reserving resources for potentially better future tasks.

Problem Objective

Design an **online scheduling** framework to maximize the **long-term expected profit** from retraining tasks.

Constraints

- Guarantee allocation for reserved tasks
- Schedule opportunistic retraining jobs
- Make real-time, irrevocable decisions

Value Prop

- ✓ Enhance edge efficiency
- ✓ Cost reduction
- ✓ sustainability

Challenges

- **Real-time**, **Irrevocable** Decisions under **Uncertainty**
- **Heterogeneous** Resources & Tasks Assignment
  - Heterogeneous server capacities, diverse task delay constraints, accuracy gain.
  - Suboptimal matching wastes precious UEC and harms future opportunities.

Problem Objective

Design an **online scheduling** framework to maximize the **long-term expected profit** from retraining tasks.

Constraints

- Guarantee allocation for reserved tasks
- Schedule opportunistic retraining jobs
- Make real-time, irrevocable decisions

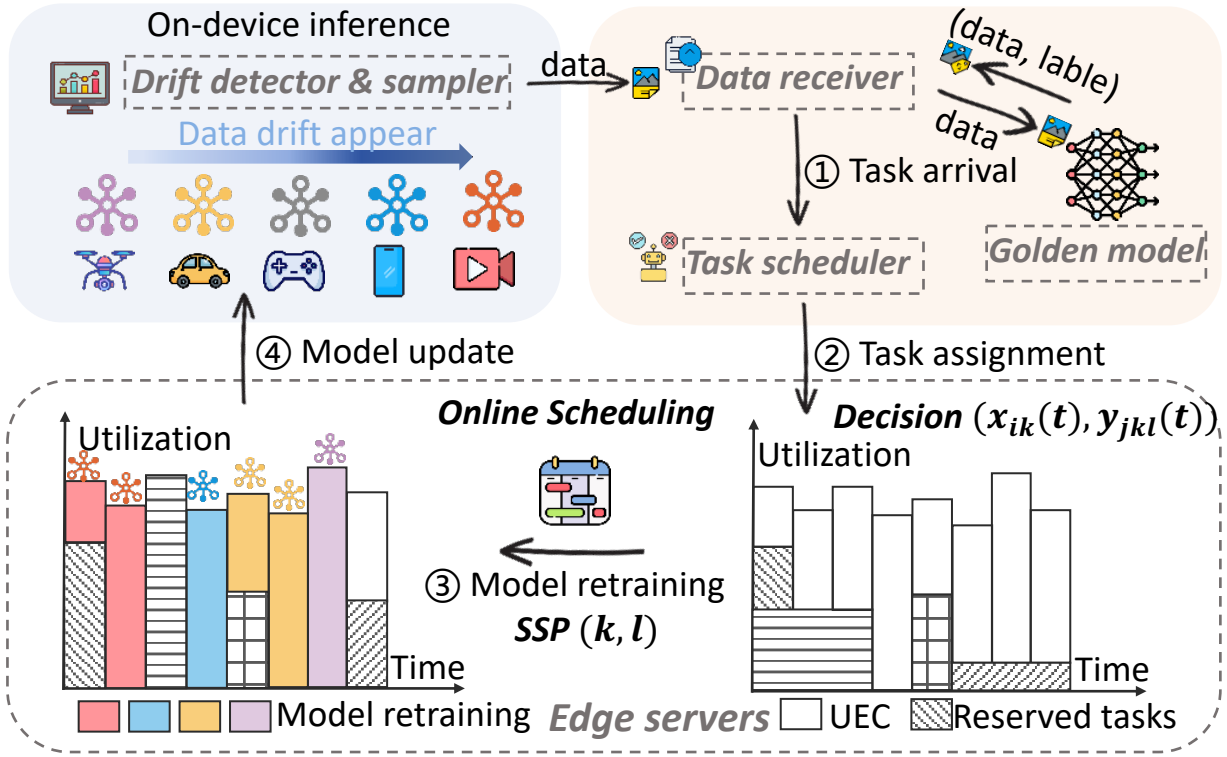
Value Prop

- ✓ Enhance edge efficiency
- ✓ Cost reduction
- ✓ sustainability

Challenges

- **Real-time**, **Irrevocable** Decisions under **Uncertainty**
- **Heterogeneous** Resources & Tasks Assignment
- **Co-scheduling** with High-Priority Reserved Tasks
  - Not just fill gaps passively, but proactively co-schedule both task types.
  - Strategically shape UEC to opportunistic retraining tasks while ensuring reserved demands.

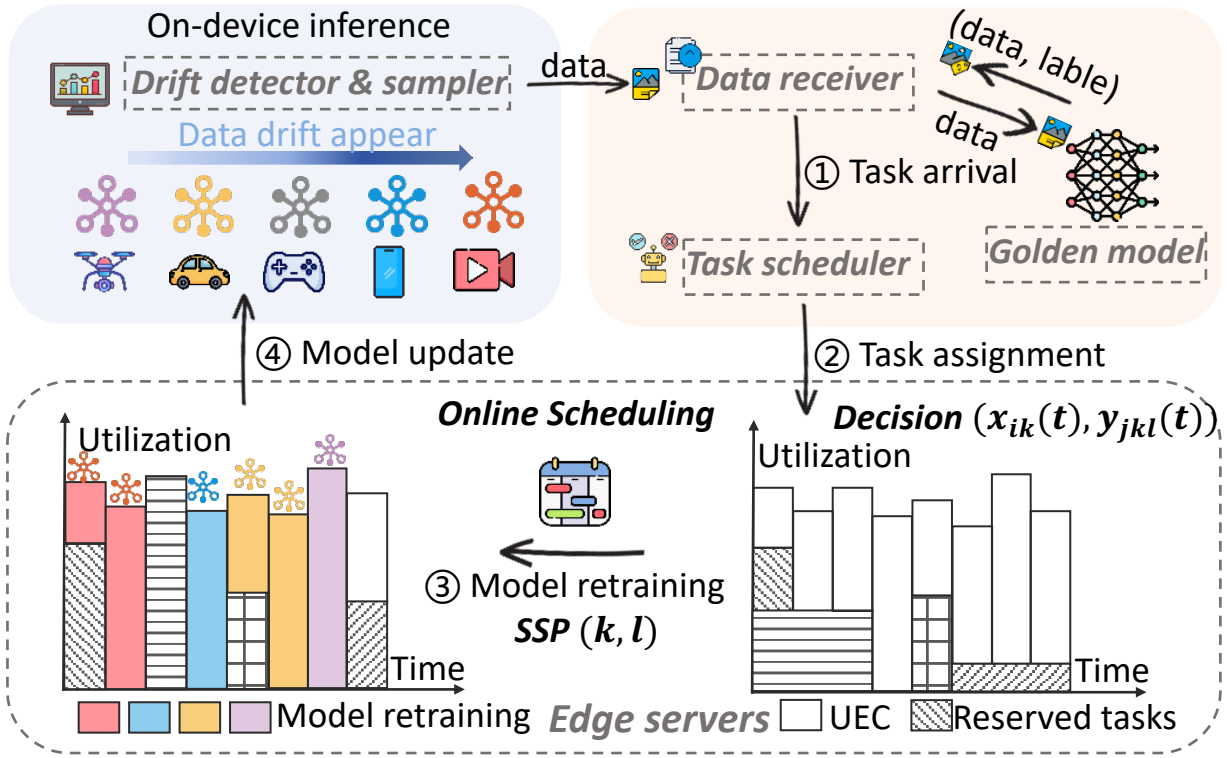
Workflow & Formulation



System Workflow

$$
\begin{aligned}
(\mathrm{P1}) \quad \max \quad & \sum_{t,j,k,l} p_j(t) y_{jkl}(t) R_{jkl}(t) \\
s.t.: \quad & \sum_{k,l} y_{jkl}(t) \le 1, \quad \forall j, t, \\
& \underbrace{\sum_{j,l} \sum_{t' \le t} p_j(t') y_{jkl}(t') \Pr(d_l^r \ge t - t' + 1)}_{\text{Expected fraction for retraining tasks}} \\
& + \underbrace{\sum_{i \in \mathcal{H}_k} x_{ik}(t)}_{\text{Fraction for reserved tasks}} \le 1, \quad \forall k, t, \\
& y_{jkl}(t) \in \{0, 1\}, \quad \forall j, k, l, t, \\
& \sum_{t=s_i^k}^{d_i^k} c_k(t) x_{ik}(t) \ge \phi_i^k, \quad \forall k, i, \\
& 0 \le x_{ik}(t) \le 1, \quad \forall k, i, t, \\
& x_{ik}(t) = 0, \quad \forall k, i, t \notin [s_i^k, d_i^k].
\end{aligned}
$$

Workflow & Formulation



System Workflow

$$\text{(P1)} \quad \max \sum_{t,j,k,l} p_j(t) y_{jkl}(t) R_{jkl}(t)$$

excepted profit combining accuracy gain and delay

$$s.t.: \quad \sum_{k,l} y_{jkl}(t) \le 1, \quad \forall j, t,$$

retraining task decision

$$\underbrace{\sum_{j,l} \sum_{t' \le t} p_j(t') y_{jkl}(t') \Pr(d_l^r \ge t - t' + 1)}_{\text{Expected fraction for retraining tasks}} + \underbrace{\sum_{i \in \mathcal{H}_k} x_{ik}(t)}_{\text{Fraction for reserved tasks}} \le 1, \quad \forall k, t,$$

resource capacity

$$y_{jkl}(t) \in \{0, 1\}, \quad \forall j, k, l, t,$$

$$\sum_{t=s_i^k}^{d_i^k} c_k(t) x_{ik}(t) \ge \phi_i^k, \quad \forall k, i,$$

reserved task demand

$$0 \le x_{ik}(t) \le 1, \quad \forall k, i, t,$$

reserved task decision

$$x_{ik}(t) = 0, \quad \forall k, i, t \notin [s_i^k, d_i^k].$$

Outline

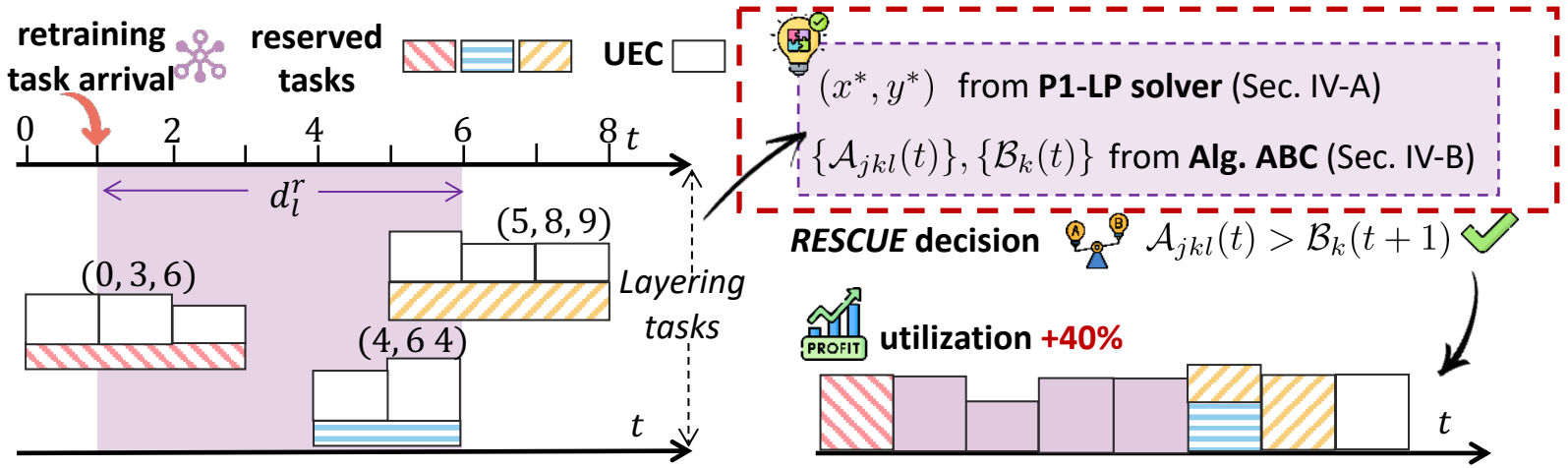
1 Introduction

2 Problem

3 **Algorithm**

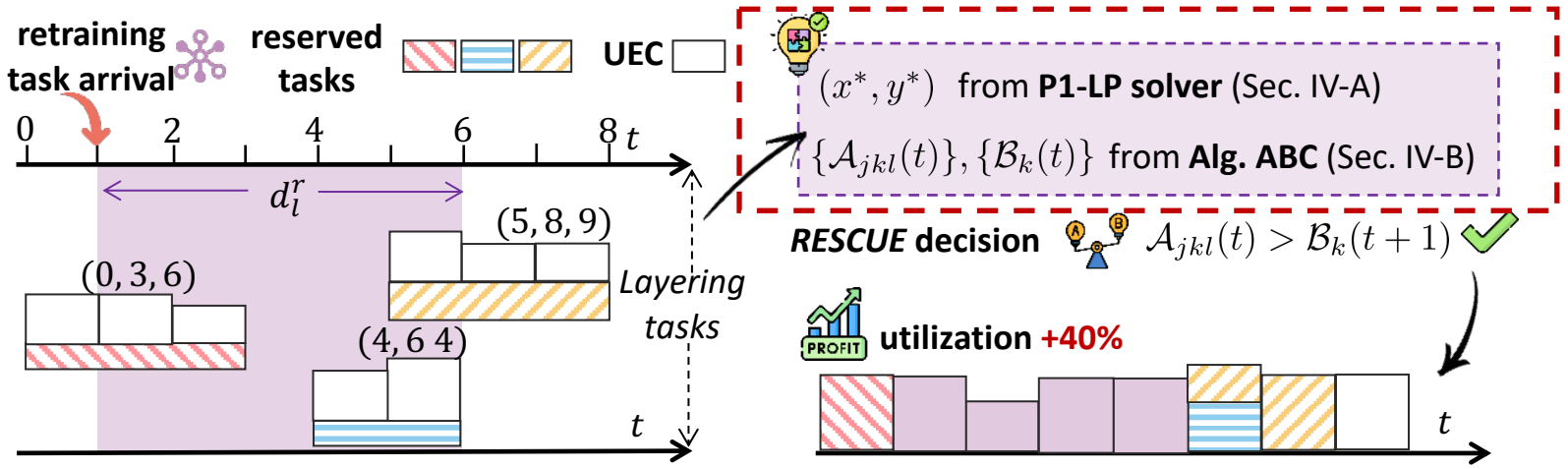
4 Simulation

5 Conclusion



Core Insight

Foresight of **offline global planning** + efficiency of **online local decision-making**



Core Insight

Foresight of **offline global planning** + efficiency of **online local decision-making**

Offline Planning

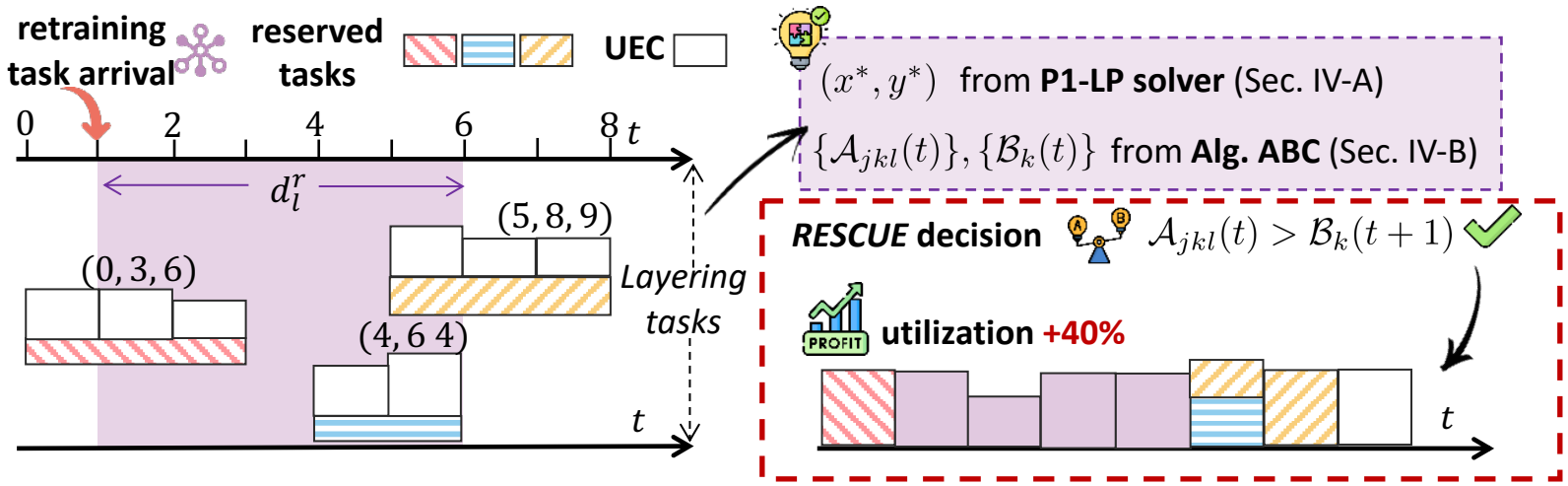
- ✓ Solves the **LP Relaxation** of P1, obtaining (x^*, y^*) as an upper bound
- ✓ Computes **value functions** $\{\mathcal{A}_{jkl}(t)\}, \{\mathcal{B}_k(t)\}$ via DP for pricing future resource

$\mathcal{A}_{jkl}(t) = R_{jkl}(t) + \sum_{d=1}^{T-t} Pr(d_l^r = d)\mathcal{B}_k(t+d)$

→ *the total expected profit from **accepting** task j with SSP (k, l) at time t*

$\mathcal{B}_k(t) = \sum_{j,l} y_{jkl}^*(t) \max\{\mathcal{A}_{jkl}(t), \mathcal{B}_k(t+1)\} + (1 - \sum_{j,l} y_{jkl}^*(t))\mathcal{B}_k(t+1)$

→ *the maximum expected **profit achievable** from server k from time t onward*



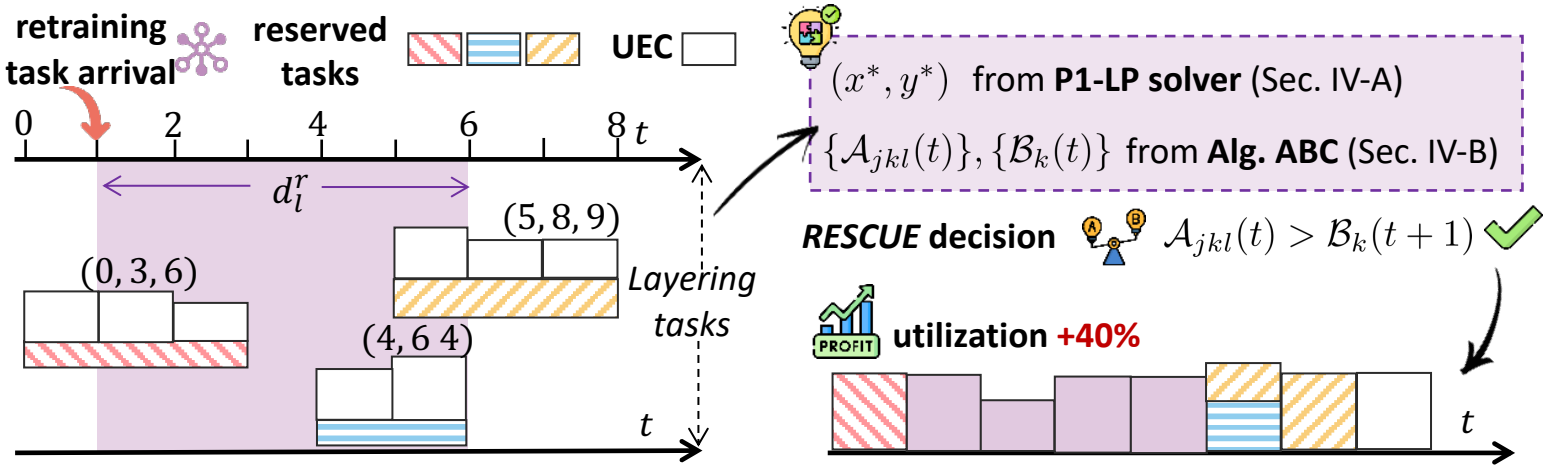
Core Insight

Foresight of **offline global planning** + efficiency of **online local decision-making**

Online Decision

- ✓ **Randomized selection** of a promising Server-Service Profile (SSP) pair
- ✓ **Threshold-based judgment** $\mathcal{A}_{jkl}(t) > \mathcal{B}_k(t+1)$ to accept or reject

"profit from accepting now" > "opportunity cost of reserving the server for the future"



Theoretical Analysis

- The **RESCUE** online algorithm achieves a tight **1/2 competitive ratio**

Proof Sketch

✓ Construct an auxiliary *single-server instance* by $p'_{kl}(t) = \sum_j y^*_{jkl}(t), p'_{kl}(t)R'_{kl}(t) = \sum_j y^*_{jkl}(t)R_{jkl}(t)$

$$\begin{cases} \tilde{\mathcal{A}}_{kl}(t) = R'_{kl}(t) + \sum_{d=1}^{T-t} Pr(d_l^r = d)\tilde{\mathcal{B}}_k(t+d) \\ \tilde{\mathcal{B}}_k(t) = \sum_l p'_{kl}(t) \max\{\tilde{\mathcal{A}}_{kl}(t), \tilde{\mathcal{B}}_k(t+1)\} + (1 - \sum_l p'_{kl}(t))\tilde{\mathcal{B}}_k(t+1) \end{cases}$$

✓ Prove $\mathcal{B}_k(1) \geq \tilde{\mathcal{B}}_k(1) \geq \frac{1}{2}\sum_t \sum_{j,l} y^*_{jkl}(t)R_{jkl}(t)$ via *backward induction* and *primal-dual analysis*

✓ Derive the global 1/2 ratio via summation and linkage arguments

Outline

1 Introduction

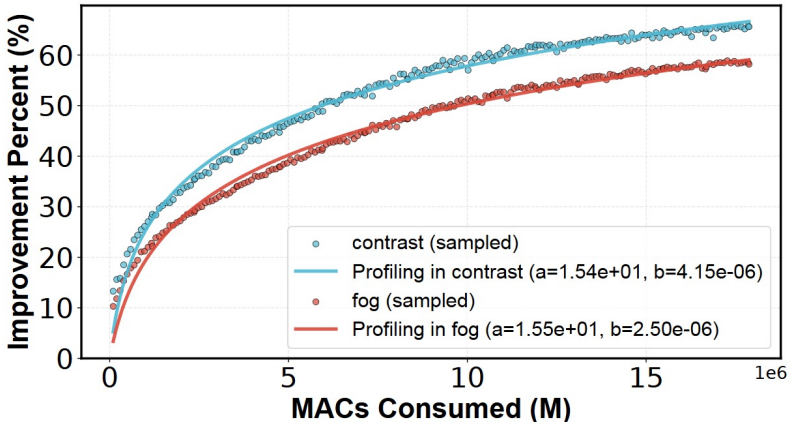
2 Problem

3 Algorithm

4 **Simulation**

5 Conclusion

Simulation Setting



Dataset & Models

- CIFAR-10-C (19 corruption types).
- MobileNetV2 (compressed DNN for devices)
- ResNet50 (generates pseudo-labels)
- Empirical *resource-to-performance profiling*

System & Network

- 2–7 edge servers, 10–18 devices
- Heterogeneous, 5–10 tera MACs/slot
- Common parameters (bandwidth, power, distance)

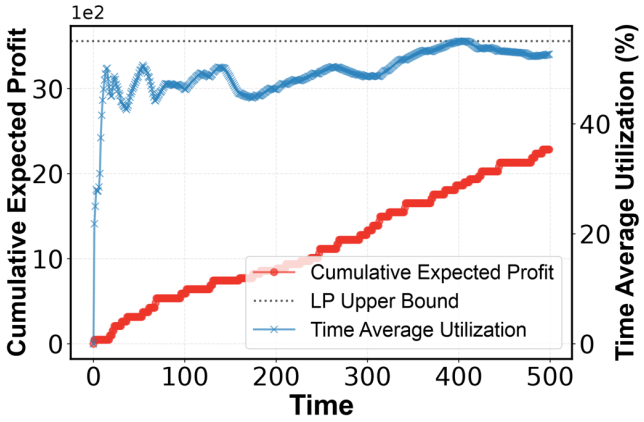
UEC Environment (Controlled by)

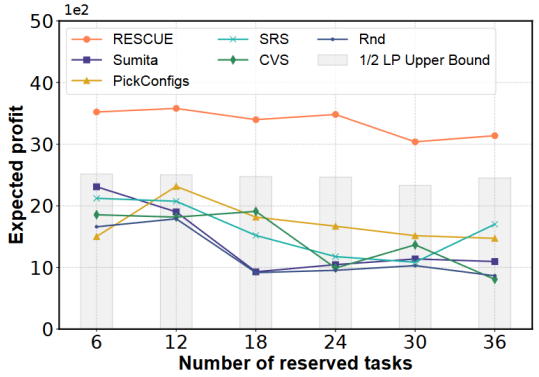
- 6–36 reserved tasks.
- [10%, 60%] occupation
- [5%, 30%] compute requirement

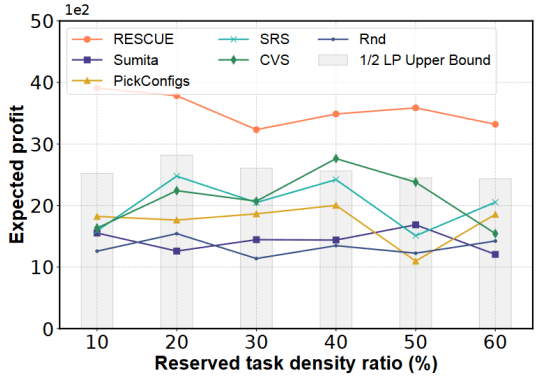
Online Retraining Tasks

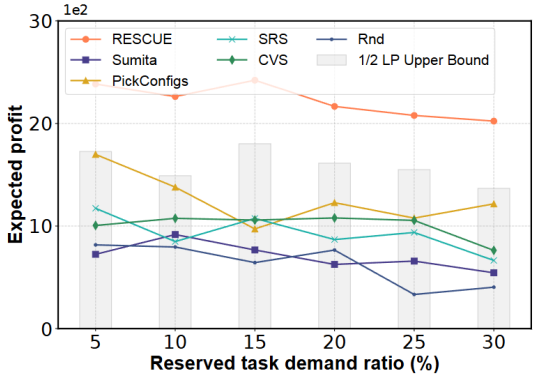
- Sinusoidal arrival probability [20%, 70%]
- Upload 8–32MB, Deadline 18–28 slots
- 3–8 service profiles, profit weight [0.5, 10]

Simulation results



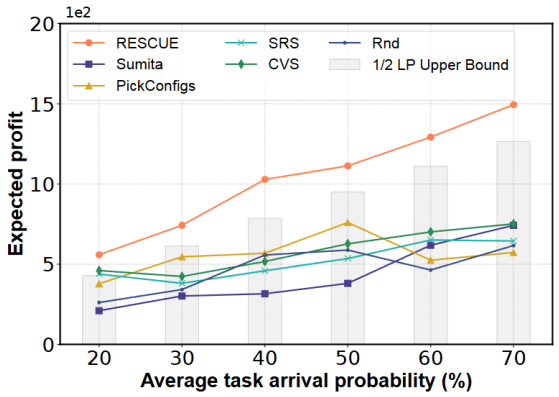


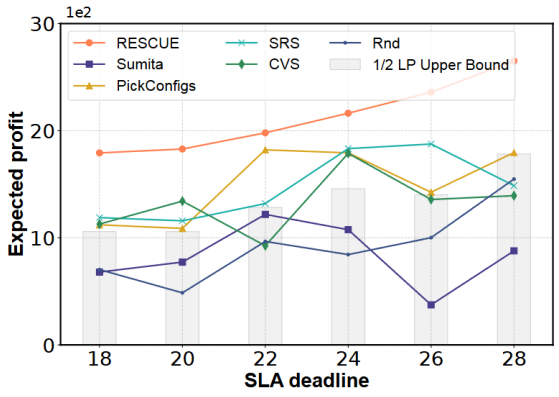


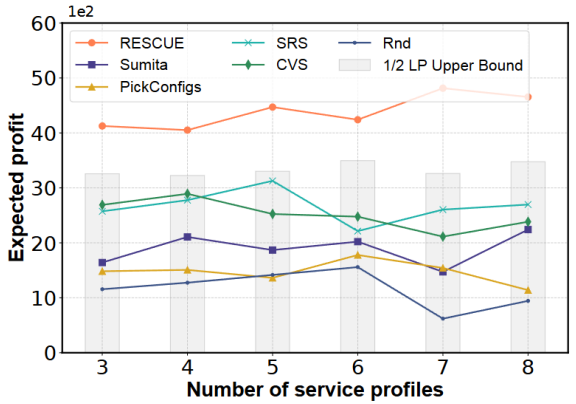


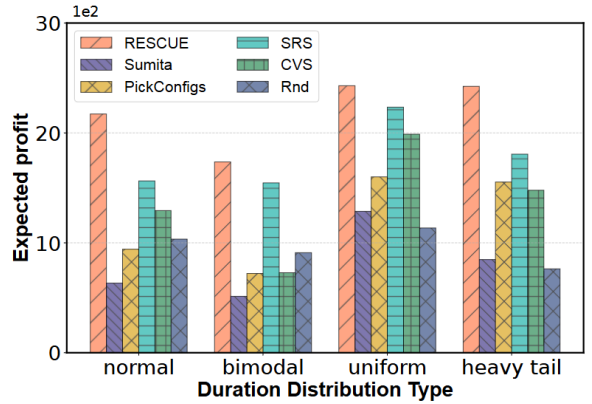
- *RESCUE* outperforms all baselines across all scenarios, achieving an average profit increase of over **60%**, with advantages magnified under high fragmentation.
- The empirical competitive ratio stabilizes at **0.64**, profit accumulates near-linearly, and UEC utilization reaches ~**53%**, validating the core design.

Simulation results









- *RESCUE*'s advantage increases with higher arrival rates, and it achieves higher profit when SLA deadlines are more relaxed.
- *RESCUE* maintains a high empirical competitive ratio (**0.61–0.87**) regardless of profile number or distribution, while baselines suffer significant performance drops.

Outline

1 Introduction

2 Problem

3 Algorithm

4 Simulation

5 **Conclusion**

- ✓ **Key Observation.** Edge retraining faces resource scarcity, yet UEC resources exist
- ✓ **Our Solution *RESCUE***
  - Offline: Computes value functions, giving resources a future view.
  - Online: Value-based randomized threshold decisions enable foresightful scheduling
- ✓ **Key Contributions**
  - First unified model for co-scheduling reserved and opportunistic retraining tasks
  - Design and proof of a 1/2-competitive online algorithm
  - Extensive validation showing >60% profit gain
- ✓ **Future Work**
  - Extend to non-stationary or unknown task arrival distributions
  - Incorporate more practical constraints like energy consumption, network dynamics

Thanks for your listening!

Q & A