Budget-Aware Cost-Efficient Edge AI Inference for Data Streams via Blockchain-Empowered Auctions

Yining Zhang, Lei Jiao, Konglin Zhu, Yebo Feng, Lin Zhang, Jiahua Xu

***Abstract*—To achieve efficient edge AI inference, edge AI service providers often acquire pre-trained models from the model market. However, this process becomes complex due to unknown bid and inference quality distributions, the strategic behavior of self-interested providers, strict budgetary constraints, and the single-point-of-failure risks inherent in centralized platforms. To address all these challenges, we propose a blockchain-empowered model auction mechanism that facilitates secure and cost-efficient model procurement. Specifically, we design a novel budget-aware auction-assisted dual-learning algorithm framework, named `AucLearn`, which jointly optimizes model auction and blockchain control decisions. `AucLearn` leverages real-time performance feedback to minimize long-term total cost while strictly adhering to both financial and computational resource budgets. We further provide rigorous proofs of multiple performance guarantees for `AucLearn`, including sub-linear regret for both subproblems and the overall problem, sub-linear fit for long-term capacity constraints, as well as truthfulness and individual rationality in each auction. Extensive evaluations driven by real-world traces demonstrate that `AucLearn` exhibits significant advantages over multiple state-of-the-art solutions.**

***Index Terms*—Edge AI, Data Stream, Blockchain, Auction, Online Optimization.**

I. Introduction

Edge AI inference [1], [2] deploys machine learning models on edge infrastructures to enable low-latency inference close to end-users. Operating both infrastructures and services, *edge AI service providers* can often source AI models either from remote clouds, or through AI model marketplaces from external *model providers* [3]–[5] who offer a variety of pre-trained models, for agility, specialization, and cost-efficiency reasons. This paradigm promises highly responsive AI services, yet

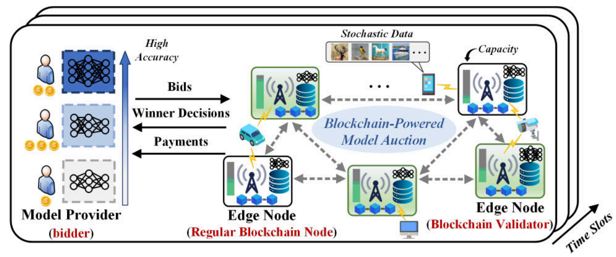


Fig. 1: Blockchain-empowered model auction framework

faces two fundamental hurdles. First, edge AI inference often needs to tackle dynamic data streams. The content and quantity of these streams typically follow unknown stochastic distributions [6], leading to inherent uncertainties in inference quality (e.g., accuracy). Second, model providers are self-interested actors. Without effective incentive mechanisms, their participation cannot be guaranteed, threatening the supply of high-quality models. Therefore, a comprehensive solution for edge AI inference must handle both the randomness of the data and the incentives for the model provision.

To incentivize model providers to contribute their models, a monetization mechanism is required. Some AI marketplaces adopt direct pricing [7], [8], which can be inferior due to risk of mispricing [9], lack of market efficiency, and inability to adapt to real-time demand and supply. *Auctions* are well-suited for addressing these disadvantages. In this paper, we advocate an auction-based approach, and model providers act as bidders from which edge AI service providers procure models for real-time inference execution. However, designing an auction-based mechanism to continuously operate and coordinate edge AI to achieve cost-efficient and performance-guaranteed inference over stochastic data streams is a complex task. We identify three unique and fundamental challenges as follows.

Uncertainty in Model Performance: At each edge, edge AI service providers lack prior knowledge of true model performance on future data streams, observing only sample-based feedback (e.g., post-inference loss) from stochastic arrivals. Consequently, selecting the optimal model becomes an inherent stochastic optimization problem that aims to minimize expected loss and overhead without access to the underlying data distributions. This necessitates dynamically balancing the trade-off between exploiting the best model discovered so far and exploring new models with unknown potential. The fundamental challenge lies in the fact that the expected performance is invisible and must be optimized solely based on noisy, bandit-style sample observations, making theoretical convergence guarantees difficult to achieve.

Strategic Behavior of Model Providers: In real-world

This work was supported in part by the National Key Research and Development Program of China (2023YFB2704500), the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, the Beijing Key Laboratory of Multimodal Data Intelligent Perception and Governance, and the U.S. National Science Foundation (CNS-2047719 and CNS-2225949). (Corresponding authors: Lei Jiao, Konglin Zhu.)

Y. Zhang and L. Zhang are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China (e-mail: {ynzhang,zhanglin}@bupt.edu.cn).

L. Jiao is with the Center for Cyber Security and Privacy, University of Oregon, Eugene, OR 97403, USA (e-mail: ljiao2@uoregon.edu).

K. Zhu is with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing 100876, China and with the Beijing Key Laboratory of Multimodal Data Intelligent Perception and Governance, Beijing 100876, China (e-mail: klzhu@bupt.edu.cn).

Y. Feng is with the College of Computing and Data Science, Nanyang Technological University, Singapore 639798 (e-mail: yebo.feng@ntu.edu.sg).

J. Xu is with the Department of Computer Science and the Centre for Blockchain Technologies, University College London, London WC1E 6BT, UK (e-mail: jiahua.xu@ucl.ac.uk).

scenarios, model providers are typically self-interested, often misreporting costs strategically, with their bids following unknown distributions. Furthermore, due to variations in training data and resources, models from different providers exhibit disparate inference performance. This creates a risk where service providers may inadvertently select cheap but low-quality models, undermining inference accuracy. Given the typically constrained procurement budgets, model auctions become even more challenging: it is essential to design a performance-aware auction mechanism that is incentive-compatible, ensuring both truthfulness and individual rationality in the face of unknown bids distributions, while simultaneously learning to identify cost-effective models with unknown quality distributions.

Resource Trade-offs in Decentralized Auctions: Edge AI service providers and model providers often lack mutual trust [10], and centralized auction platforms rely on centralized auctioneers, which introduce single-point-of-failure risks. While decentralized mechanisms like *blockchain*s mitigate these trusts issues, they introduce significant resource trade-offs. Specifically, maintaining a blockchain requires selecting *validator* nodes to balance operational risks and operational costs. However, each edge node typically possesses limited computational capacity and selecting a specific node as the validator may hinder its future usage. This intertwines with the model auction decision-making process, as both the execution of validators and model inference consume computational resources in the shared edge computing environment.

Existing research is not adequate for addressing the aforementioned challenges. The works in [11]–[16] focus on model partitioning or offloading in static settings. They fail to account for the stochastic nature of online data streams, rendering them ineffective for dynamic model selection. The works on incentives in cloud-edge systems [17]–[23] mainly adopt direct pricing or game-theoretic approaches without establishing incentive-compatible guarantees, and often ignore long-term optimization under procurement budgets. Meanwhile, blockchain system studies [20]–[23] often focus on data or resource transactions, overlooking the characteristics of model trading for edge AI. See Section VI for detailed discussions.

In this paper, we address all the aforementioned challenges via a rigorous mathematical and algorithmic approach, accompanied by solid theoretical analysis. Fig. 1 illustrates our target scenario. Our contributions are as follows:

First, we formulate a time-cumulative stochastic optimization problem to jointly minimize the expected inference loss over dynamic data streams, blockchain operational risks, and costs, while enforcing long-term auction budget and computational capacity constraints. Our formulation controls winning-bid selection, payments to bids in each auction, and validator selection in an online manner, capturing arbitrarily unknown stochastic distributions of data samples, arrivals, and bidding costs. We prove that this problem is a non-linear mixed-integer programming problem and is NP-hard.

Second, we propose `AucLearn`, a novel polynomial-time framework based on auction-assisted dual learning. To address the distinct characteristics of the coupled long-term constraints, `AucLearn` decouples the original problem into two interdependent subproblems. For the budget-constrained model auction, we design a budget-aware *bandit* mechanism that integrates auction dynamics with exploration-exploitation trade-offs. By estimating inference performance and bidding costs in an unbiased manner, it adaptively selects the most cost-effective model to maximize budget utilization. It further incorporates payments based on Myerson's theorem to guarantee the economic properties. For the capacity-constrained validator selection, we develop an *online learning* algorithm [24], [25] based on Online Convex Optimization (OCO) with time-varying long-term constraints. Leveraging a convex-concave reformulation and rectified online primal-dual steps, it decomposes the problem into sequential one-shot optimizations, balancing operational risk and cost without future knowledge. A *randomized rounding* algorithm is further applied to convert fractional control decisions into integers without violating any constraints. Notably, `AucLearn` establishes the first unified framework combining OCO, bandit learning, and auction theory, making it broadly applicable to similar scenarios.

Third, we perform rigorous theoretical analysis for our proposed algorithms. For the model auction, we prove sub-linear *regret* [26], ensuring that the performance gap against the hindsight optimum grows sub-linearly. We further establish expected *truthfulness* and *individual rationality*, guaranteeing that participants have no incentive to misreport costs and incur no losses. For validator selection, we derive sub-linear bounds on both *regret* and *fit* [27], implying that time-averaged cumulative deviation from the cost of the sequence of instantaneous optimizations and long-term constraint violations vanish asymptotically. Finally, by constructing auxiliary intermediate terms, we extend these results to derive global regret bounds for the overall joint optimization system.

Fourth, we conduct experiments using inference data from MNIST, CIFAR-10, and CIFAR-100 [28], [29], deploying up to 400 edge nodes under realistic settings [5], [30]–[35] based on London Underground user workloads [36], and utilizing diverse DNN models [37]–[40]. Our evaluations reveal the following results: (i) Compared to nine combinations of different model auction methods (`AUCB` [41], `CMABA` [42], and `the` ε`-first` [43]) and validator selection methods (`Greedy`, `Lyapunov` [44], and `OGD` [45]), `AucLearn` reduces the cumulative total cost on average by $17\% \sim 59\%$; (ii) `AucLearn` outperforms all online approaches under comparison with an accuracy gain of 0.06%-4.62%; (iii) `AucLearn` incurs the lowest regret and nearly the lowest fit; (iv) `AucLearn` achieves truthfulness and individual rationality in expectation, while ensuring the selection of the most cost-effective model; (v) `AucLearn` demonstrates good scalability as the total number of edge nodes or other parameter settings varies.

II. Model and Problem Formulation

A. System Modeling

Edge Infrastructure: We consider an edge network consisting of a set of distributed and heterogeneous edge nodes, denoted by $\mathcal{I} = \{1, \ldots, I\}$. Each edge node represents a micro data center or server cluster co-located with a cellular base station or WiFi access point in close proximity to end users, who typically access the edge via wireless connections.

We assume the nodes are operated by different *edge AI service providers* who may lack mutual trust. Each edge node $i \in \mathcal{I}$ is characterized by a limited computational capacity C_i, measured in Floating Point Operations (FLOPs) [46]. The entire system operates over the time slots $\mathcal{T} = \{1, \ldots, T\}$.

Stochastic Data Streams: Without loss of generality, data samples from users are considered to be drawn from an unknown time-invariant stochastic distribution $\mathscr{D}$ [47], [48]. Specifically, we denote a sample as $(a, b) \sim \mathscr{D}$, where a represents the feature and b denotes the ground-truth label. Consequently, the data stream arriving at each edge constitutes an Independent and Identically Distributed (IID) stochastic process. We further assume that the number of data samples arriving at edge i, denoted by M_i, follows another unknown time-invariant stochastic distribution.

Machine Learning Models: To process these data streams, edge nodes procure models from a set of *model providers* $\mathcal{N} = \{1, \ldots, N\}$ via the blockchain-empowered auction. We assume each provider offers a single model; providers with multiple models are treated as multiple "virtual" providers. Each model n is characterized by a computational workload α_n, e.g., FLOPs required for one inference; the computational latency incurred by edge node i to perform inference on a single data sample using model n is denoted by $v_{i,n}$. Crucially, $v_{i,n}$ is unknown *a priori* and observable only after deployment. Functionally, model n is defined by a prediction function $h_n : \mathcal{X} \to \mathcal{Y}$. h_n is model-agnostic and can be instantiated by any pre-trained classification or regression model (e.g., deep neural networks such as CNNs, MobileNet, ResNet, and DenseNet; see Table I in Section V). Then we quantify the model's performance using the squared loss $\ell_n(a, b) = (h_n(a) - b)^2$. Since $(a, b) \sim \mathscr{D}$, the inference loss ℓ_n also follows some distribution $\mathscr{D}_n$. Accordingly, for a batch of arriving data M_i, the average inference loss is calculated as $L_{i,n} = \frac{1}{M_i} \sum_{v=1}^{M_i} (h_n(a_v) - b_v)^2$, where both M_i and the samples $\{a_v, b_v\}$ are random variables.

Blockchain-Empowered Model Auction: To facilitate secure and efficient decentralized model trading in edge AI, we design a blockchain-based auction mechanism comprising four core components: (i) *Blockchain:* A private ledger jointly maintained by edge nodes to record transactions immutably. This enhances resilience against single-point failures and attacks while mitigating data leakage and inefficiencies in public chains. (ii) *Smart Contract:* This component automates the auction logic, including bid collection, winner determination, and payment settlement, thereby eliminating reliance on a trusted third-party auctioneer. (iii) *Model Providers:* These entities participate as bidders, offering trained models by submitting model specifications and bidding costs. (iv) *Edge Nodes:* Edge nodes participate as buyers, which purchase selected models for local inference and make payments via smart contracts. A dynamic subset of these nodes also serve as validators, responsible for consensus participation, smart contract execution, and verification of auction outcomes.

At each t, the auction is subject to a remaining auction budget B^t. B^t is initialized as $B^0 = B$ and updated as $B^t = B^{t-1} - \sum_i \sum_{n \in \Phi_i^t} p_{i,n}^t$ after each round, where Φ_i^t denotes the set of winning models selected for edge node i at slot t.

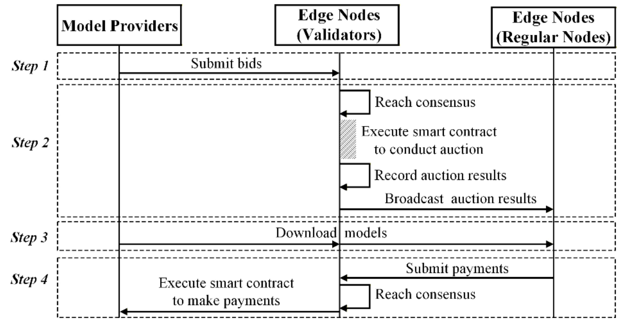


Fig. 2: System workflow in a single time slot

It governs whether subsequent auction rounds are admissible until the budget B is exhausted. The auction proceeds in four steps, as illustrated in Fig. 2:

- *Step 1:* Each model provider n submits a transaction to the smart contract containing the model specification (i.e., metadata such as architecture descriptor and version) and a set of bids $\{c_{1,n}^t, \ldots, c_{I,n}^t\}$. $c_{i,n}^t$ denotes the specific price for selling the model to edge node i, drawn from an unknown IID distribution. Each provider offers a unique model, and $c_{i,n}^t$ targets a specific buyer i.
- *Step 2:* The submitted transactions are broadcast to the network. Upon reaching consensus, validators execute the smart contract to determine winners and compute payments $p_{i,n}^t$. The finalized auction results are immutably recorded on-chain and notified to participants.
- *Step 3:* Each edge node downloads the corresponding models from the winning providers' facilities and deploys them for local inference.
- *Step 4:* Edge nodes submit payment transactions. Once confirmed by the validators, the smart contract automatically transfers $p_{i,n}^t$ to the winning providers and updates the remaining budget. The system enforces that no procurement occurs if the budget is depleted.

Note that *Step 4* can be placed before *Step 3* depending on the mutual agreement reached between the model providers and the edge AI service providers; our framework is compatible with either ordering.

Blockchain Consensus: Unlike Proof-of-Work systems that consume excessive energy, we adopt a lightweight voting-based consensus protocol inspired by the XRP Ledger [49].

Consensus Process Overview: In each consensus round, each validator collects pending transactions and proposes a candidate set. Validators engage in multi-round voting, exchanging proposals and refining their views of acceptable transactions. Once a transaction achieves supermajority agreement among the current validators, it is included in the consensus set. Validators then deterministically execute the agreed-upon transactions in a fixed order to produce the new state of the blockchain, which is broadcast to the entire network to ensure consistency. Let e^t denote the computational resource consumed by each validator during consensus at time t.

Dynamic Validator Selection: To mitigate targeted attacks and enhance decentralization, our framework adopts dynamic validator selection. The validator set is updated at the end of each t via a smart contract, based on a predefined selection mechanism, and becomes responsible for consensus at $t + 1$.

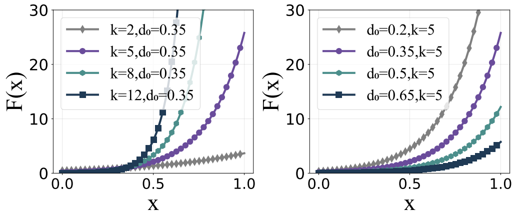


Fig. 3: Risk function under different parameters

This ensures that at least $D \geq 1$ edge nodes are selected as validators using the online algorithm in Section III-C, where D is set by the system's operational needs. However, configuring edge node i as a validator incurs an operational cost d_i^t (e.g., computation overhead). This creates a fundamental trade-off: increasing the number of validators enhances resilience but escalates the system's operational expenditure.

Operational Risk Modeling: To quantify blockchain reliability, we model the consensus success probability using a sigmoid approach [50], [51]. Specifically, let $x \in [0,1]$ denote the fraction of faulty resources. The probability of success is given by $\mathrm{F}_1(x) = \frac{1}{1+e^{k(x-d_0)}}$, where $d_0 \in \mathbb{R}$ is the threshold and $k > 0$ controls the sensitivity. To facilitate the design of an efficient online algorithm, we transform this into a convex risk function: $\mathrm{F}(x) = \frac{1}{\mathrm{F}_1(x)} - 1 = e^{k(x-d_0)}$. As illustrated in Fig. 3, $\mathrm{F}(x)$ grows exponentially when x exceeds d_0, and the rate of increase is determined by k. Based on this risk function, the system's operational risk at time t is defined as:

$$\mathrm{F}\left(\frac{A^t}{C_{\text{total}}^t \delta}\right) = \exp\left(k\left(\frac{A^t}{C_{\text{total}}^t \delta} - d_0\right)\right), \tag{1}$$

where A^t represents the magnitude of faulty/attacked resources, which is unpredictable at $t-1$. C_{total}^t denotes the total operational cost of the active validators, and δ is the conversion factor mapping operational cost to computational resources. This convex risk metric (1) allows the system to balance the cost of activating validators against the risk.

Machine Learning Inference: Following the auction, the machine learning inference workflow on each edge node i at time slot t proceeds as follows:

- *Step 1:* Based on the auction outcome, edge node i deploys K winning models, which can be represented as $n \in \mathcal{K}_i^t = \{1, \ldots, K\}, \forall i$.
- *Step 2:* M_i^t data samples arrive sequentially, where M_i^t drawn from the random variable M_i. The model deployment is performed once at the start of slot t, and the same set $\mathcal{K}_i^t$ serves all M_i^t samples arriving within the slot. Then, repeat the following *Steps 2.1∼2.3* for each $m \in \{1, \ldots, M_i^t\}$:
  - *Step 2.1:* The edge receives the input feature a_m.
  - *Step 2.2:* The edge executes inference using all deployed models in $\mathcal{K}_i^t$ to obtain a set of predicted labels $\{h_n(a_m)\}_{n \in \mathcal{K}_i^t}$.
  - *Step 2.3:* The edge obtains the ground-truth label b_m.
- *Step 3:* The edge then evaluates the incurred loss for each selected model $n \in \mathcal{K}_i^t$, $L_{i,n}^t = \frac{1}{M_i^t}\sum_{m_i=1}^{M_i^t}\left(h_{n_i}(a_{m_i}) - b_{m_i}\right)^2$ as a sample of the random variable $L_{i,n}$. Additionally, if the computational latency $v_{i,n}$ for a model n was previously unknown, it is measured and recorded at this stage.
- *Step 4:* The system records both $L_{i,n}^t$ and $v_{i,n}$ for all $n \in \mathcal{K}_i$. These observations are used to update the historical statistics, which provide necessary information for making auction decisions in the next time slot $t+1$.

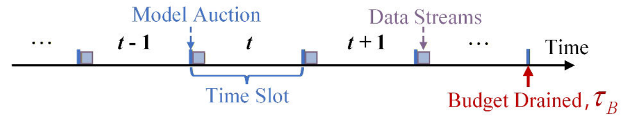


Fig. 4: Illustration of time slot flows

Control Decisions: At each time slot t, the system dynamically optimizes the following set of control variables: Let $x_{i,n}^t \in \{0,1\}$ be a binary variable indicating whether model provider n's bid for edge i wins the auction. Let $y_i^t \in \{0,1\}$ be a binary variable indicating whether edge i is activated as a validator for the consensus process. Let $p_{i,n}^t \geq 0$ denote the payment amount made to model provider n from edge node i. For brevity, we denote the system-wide decision vectors at time t as $\boldsymbol{x}^t \triangleq \{x_{i,n}^t\}_{i\in\mathcal{I},n\in\mathcal{N}}$, $\boldsymbol{y}^t \triangleq \{y_i^t\}_{i\in\mathcal{I}}$, and $\boldsymbol{p}^t \triangleq \{p_{i,n}^t\}_{i\in\mathcal{I},n\in\mathcal{N}}$.

System Total Cost: The total cost incurred by the edge system at time slot t is composed of three key components: (i) The *operational risk* of the blockchain system, as formalized in (1). Here, we explicitly express the total operational cost as $C_{\text{total}}^t = \sum_i y_i^t d_i^t$ by incorporating the decision variable $\boldsymbol{y}^t$; (ii) The *operational cost* for keeping the selected validators active, quantified as $\sum_i y_i^t d_i^t$; (iii) The *inference cost*, which aggregates the expected inference loss over the data distribution $\mathscr{D}$ and the inference latency for a single data sample $v_{i,n}$, averaged over the K models: $\frac{1}{K}\sum_i \sum_n x_{i,n}^t(\mathbb{E}_{\ell_n\sim\mathscr{D}_n}[\ell_n] + v_{i,n})$. Note that the total cost we defined does not include payments, as we have placed payments in the constraint, which needs to be carefully determined to satisfy economic properties.

B. Problem Formulation and Algorithmic Challenges

Total Cost Minimization: We formulate the total cost minimization problem $\mathbb{P}_0$ as follows. Note that $\mathbb{P}_0$ is a stochastic non-linear mixed-integer programming problem and is provably NP-hard.

$$\begin{aligned}
\mathbb{P}_0: \min & \textstyle\sum_{t\leq\tau_B}\{\mathrm{F}\left(\frac{A^t}{\sum_i y_i^t d_i^t \delta}\right) + \sum_i y_i^t d_i^t \\
& + \tfrac{1}{K}\textstyle\sum_i\sum_n x_{i,n}^t(\mathbb{E}_{\ell_n\sim\mathscr{D}_n}[\ell_n] + v_{i,n})\} && (2)\\
s.t. \quad & \textstyle\sum_n x_{i,n}^t = K, \forall i\in\mathcal{I}, \forall t\leq\tau_B, && (2a)\\
& \textstyle\sum_i y_i^t \geq D, \forall t\leq\tau_B, && (2b)\\
& \textstyle\sum_{t\leq\tau_B}\sum_n x_{i,n}^t\alpha_n M_i^t + \sum_{t\leq\tau_B} y_i^t e^t \leq C_i, \forall i\in\mathcal{I}, && (2c)\\
& \tau_B = \max_\tau\left\{\tau \mid \textstyle\sum_{t\leq\tau}\sum_i\sum_n p_{i,n}^t \leq B\right\}, && (2d)\\
\text{var.} \quad & x_{i,n}^t, y_i^t \in \{0,1\}, p_{i,n}^t \geq 0, \forall i\in\mathcal{I}, \forall n\in\mathcal{N}, \forall t\leq\tau_B.
\end{aligned}$$

The objective (2) minimizes the total cost over time. Due to the different dimensionalities, each term in our objective function (2) is preceded by a weight (e.g., $\alpha_1, \alpha_2, \ldots$). However, for the sake of brevity, we do not explicitly present the weight values. Constraint (2a) ensures that each edge hosts K

models at each time slot. Constraint (2b) ensures the minimum number of validators for blockchain. Constraint (2c) ensures the long-term computational capacity of each edge. Constraint (2d) captures the definition of the auxiliary variable τ_B. The domains for the decision variables are specified finally. As illustrated in Fig. 4, the variable τ_B represents the "stopping time", and the optimization objective considers the time scope of $t \leq \tau_B$, where $\tau_B \leq |\mathcal{T}|$, rather than over the entire time scope of $t \leq |\mathcal{T}|$, with $\mathcal{T}$ denoting the set of all possible time slots.

Algorithmic Goal: We aim to develop an online algorithmic framework for problem $\mathbb{P}_0$, producing a solution of the form $\{\{\bar{x}_{i,n}^t, \bar{p}_{i,n}^t, \forall i, \forall n\}, \{\bar{y}_i^t, \forall i\}, \forall t\}$, which guarantees a regret bound of $\mathcal{P}(\{\bar{x}_{i,n}^t, \bar{y}_i^t\}_{t=1}^{\tau_B}) - \mathcal{P}^* \leq \Omega$, where $\mathcal{P}$ is the objective of $\mathbb{P}_0$, $\mathcal{P}^*$ is the optimal cumulative cost, and the regret bound Ω scales sub-linearly with the time horizon τ_B.

Algorithmic Challenges: ① Since $\mathscr{D}_n$ is unknown, the associated expectation cannot be computed directly. Moreover, at each t, we do not observe direct samples drawn from l_n itself, but rather from $L_{i,n}$ for each i, which involves another random variable M_i. Optimizing the expectation of l_n based on observations from $L_{i,n}$ is non-trivial. ② The *long-term constraint* of computational capacity needs to be considered carefully. Determining $\boldsymbol{x}^t$ and $\boldsymbol{y}^t$ at t affects not only the cost at t but also the determination of decisions for future t, in order to satisfy this constraint. ③ The *auction budget* B is related to the algorithm's stopping time τ_B; thus, it needs to be allocated carefully. Misallocation of B may cause early depletion, leading to degraded real-time inference performance or interruptions in future model auction. ④ Our problem is nonlinear and NP-hard by Theorem 1. As the inputs gradually reveal themselves, solving the NP-hard problem becomes increasingly challenging, especially considering the non-linear objective function with two competing components. ⑤ Designing a payment mechanism that ensures truthfulness and individual rationality is challenging. Standard auction theories are often inapplicable due to the stochastic bid distributions and strict budget constraints in our framework.

Theorem 1 *The problem $\mathbb{P}_0$ is NP-hard.*

Proof. We show that $\mathbb{P}_0$ contains the classic 0/1 knapsack problem, which is NP-hard, as a special case. Consider the entire time horizon that consists of only one time slot t, where the variables $\boldsymbol{y}^t$ are fixed and the risk term is ignored. The problem then reduces to selecting a subset of models $\boldsymbol{x}^t$ to optimize the inference cost subject to the resource constraint (2c). This subproblem is equivalent to the knapsack problem: the resource consumption $\alpha_n M_i^t$ corresponds to the "weight" of an item, the capacity C_i corresponds to the knapsack's limit, and the decision variable $x_{i,n}^t$ represents the item selection. □

III. Online Algorithm Design

To address the challenges ①–⑤, we propose a decomposition framework that decouples $\mathbb{P}_0$ into two interdependent subproblems, each handling a specific constraint type. We then design tailored polynomial-time online algorithms for each subproblem, and in the next section derive the rigorous performance guarantees provided by this framework.

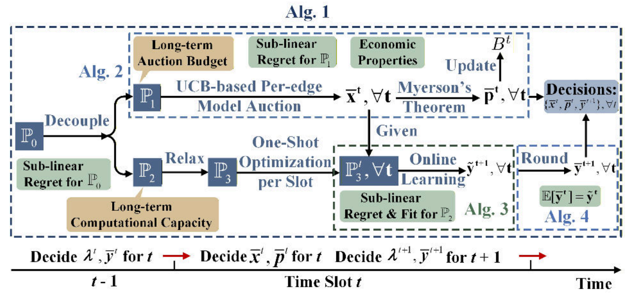


Fig. 5: Structure of AucLearn

A. Algorithm Overview with Problem Decomposition

Algorithm 1 outlines the AucLearn framework, also illustrated in Fig. 5. In the initialization phase, each edge selects all models once to estimate parameters $\beta_{i,n}^t$, $\bar{r}_{i,n}^t$, and $\bar{c}_{i,n}^t$, with payments set to c_{max}. In Lines 2–9, we iteratively invoke Algorithm 2 through 4 until the global auction budget B is exhausted. Functionally, Algorithm 2 orchestrates the model auction process and governs the stopping time τ_B, whereas Algorithms 3 and 4 collectively manage validator selection subject to long-term computational capacity constraints.

To effectively tackle the distinct long-term constraints, we structurally decompose the original problem $\mathbb{P}_0$ into two tractable subproblems, $\mathbb{P}_1$ and $\mathbb{P}_2$:

Subproblem $\mathbb{P}_1$: Problem $\mathbb{P}_1$ is formulated as:

$$\mathbb{P}_1 : \min \tfrac{1}{K} \textstyle\sum_{t \leq \tau_B} \sum_i \sum_n x_{i,n}^t (\mathbb{E}_{\ell_n \sim \mathscr{D}_n}[\ell_n] + v_{i,n}) \tag{3}$$

$$s.t. \qquad \textstyle\sum_n x_{i,n}^t = K, \forall i \in \mathcal{I}, \forall t \leq \tau_B \tag{3a}$$

$$\tau_B = \max_\tau \left\{ \tau \middle| \textstyle\sum_{t \leq \tau} \sum_i \sum_n p_{i,n}^t \leq B \right\}, \tag{3b}$$

$$\text{var.} \qquad x_{i,n}^t \in \{0,1\}, \forall i \in \mathcal{I}, \forall n \in \mathcal{N}, \forall t \leq \tau_B.$$

$\mathbb{P}_1$ involves the control variables $\{x_{i,n}^t, p_{i,n}^t \forall i, \forall n, \forall t\}$ and the long-term auction budget constraint, which directly determines the stopping time of the system.

Subproblem $\mathbb{P}_2$: For problem $\mathbb{P}_2$, we introduce additional notations to facilitate the algorithm design presented later. We denote $f^t(\boldsymbol{y}^t) = \mathrm{F}\left(\frac{A^t}{\sum_i y_i^t d_i^t \delta}\right) + \sum_i y_i^t d_i^t$; $g_i^t = \sum_n \bar{x}_{i,n}^t \alpha_n M_i^t + y_i^t e^t - C_i, \forall i$, $\mathbf{g}^t(\boldsymbol{y}^t) = [g_1^t, \ldots, g_I^t]$; where the values $\{\bar{x}_{i,n}^t, \forall i, \forall n\}$ are solved from $\mathbb{P}_1$ at t; and $h^t(\boldsymbol{y}^t) = D - \sum_i y_i^t$. Then we have

$$\mathbb{P}_2 : \min \textstyle\sum_{t \leq \tau_B} f^t(\boldsymbol{y}^t), \tag{4}$$

$$s.t. \qquad h^t(\boldsymbol{y}^t) \leq 0, \tag{4a}$$

$$\textstyle\sum_{t \leq \tau_B} \mathbf{g}^t(\boldsymbol{y}^t) \preceq \mathbf{0}, \tag{4b}$$

$$\text{var.} \qquad \boldsymbol{y}^t \in \mathcal{X} = \{\boldsymbol{y}^t | y_i^t \in \{0,1\}, \forall i \in \mathcal{I}, \forall t \leq \tau_B\}.$$

$\mathbb{P}_2$ involves the control variables $\{y_i^t, \forall i, \forall t\}$ and the computational capacity constraint (4b), which takes $\{\bar{x}_{i,n}^t, \forall i, \forall n\}$ as the input at each t. The long-term computational capacity constraint is time-varying and is often only observable as $\mathbf{g}^t(\cdot)$ after executing the decision $\boldsymbol{y}^t$. Leveraging the distinct nature of the two long-term constraints, we design customized algorithms to solve them respectively, as described below.

B. Model Auction via Budget-Aware Bandit Learning

To capture real-world dynamics, we extend the auction setup to scenarios where both bids and inference losses are

Algorithm 1: The `AucLearn` Framework

Input: $\mathcal{N}, B, K$
Output: $\bar{\boldsymbol{x}}, \bar{\boldsymbol{p}}, \bar{\boldsymbol{y}}, \tau_B$

Initialize: $t = 1$, each edge selects all models once, $p^1_{i,n} = c_{max}, \forall i, n$, primal iterate $\bar{\boldsymbol{y}}^1$, dual iterate $\boldsymbol{\lambda}^1 = \mathbf{0}$, step sizes γ_1, γ_2, budget $B^0 = B$;
while $true$ **do**
  $t = t + 1$;
  Invoke **Algorithm 2** for each edge and get $\bar{\boldsymbol{x}}^t$, $\bar{\boldsymbol{p}}^t$;
  if *Algorithm 2 returns Terminate* **then**
    $\tau_B = t$, **break**;
  Invoke **Algorithm 3** to obtain the fractions $\widetilde{\boldsymbol{y}}^{t+1}$;
  Invoke **Algorithm 4** to obtain the integers $\bar{\boldsymbol{y}}^{t+1}$;
  return Decisions $\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{p}}^t, \bar{\boldsymbol{y}}^{t+1}$;

stochastic with unknown means, contrasting with constant-bid assumptions in [41], [42]. We propose a budget-aware Upper Confidence Bounds (UCB)-based model auction (Algorithm 2) to address two key challenges: (i) *Winner Selection:* We employ UCB indices [41], [52] to estimate the upper bounds of valuation, along with a carefully designed exploration term to balance the exploration of potential optimal models against the exploitation of known cost-effective ones under a hard budget constraint. (ii) *Payment Determination:* We design a critical payment scheme based on Myerson's theorem [53]. This ensures the economic properties of truthfulness and individual rationality, incentivizing strategic providers to bid truthfully despite the uncertainties.

UCB-based Plays: We first express the inference loss as *reward* by introducing a constant, i.e., $r^t_{i,n} = \chi - L^t_{i,n} - v_{i,n}$, where $\chi > 0$ ensures non-negativity. Let $\beta^t_{i,n}$, $\bar{r}^t_{i,n}$, and $\bar{c}^t_{i,n}$ denote the selection count, empirical average reward, and empirical average bidding cost, respectively. For the selected set Φ^t_i, we update $\beta^t_{i,n} \leftarrow \beta^{t-1}_{i,n} + 1$ and $\bar{r}^t_{i,n} \leftarrow (\bar{r}^{t-1}_{i,n}\beta^{t-1}_{i,n} + r^t_{i,n})/\beta^t_{i,n}$ for $n \in \Phi^t_i$; otherwise, they remain unchanged. The cost $\bar{c}^t_{i,n}$ is updated continuously as $\bar{c}^t_{i,n} = (\bar{c}^{t-1}_{i,n}(t-1) + c^t_{i,n})/t$. Accordingly, we maintain the time-varying UCB score:

$$U^t_{i,n} = \frac{\bar{r}^{t-1}_{i,n}}{\bar{c}^t_{i,n}} + \underbrace{\frac{(c_{\min} + r_{\max})}{(c_{\min})^2} \cdot \varepsilon^t_{i,n}}_{\text{Exploration Term } u^{t-1}_{i,n}}, \tag{5}$$

where $\varepsilon^t_{i,n} = \sigma\sqrt{(K+1)\log t/\beta^t_{i,n}}$, $\sigma = \max\{r_{\max} - r_{\min}, c_{\max} - c_{\min}\}$. Here, $r_{\min\,/\,\max}$ and $c_{\min\,/\,\max}$ denote the global minimum/maximum bounds for rewards and costs, respectively.

Per-edge Auction: Particularly, Algorithm 2 performs a *per-edge* auction, which operates independently on each edge i. Specifically, it ranks models in decreasing order of their UCB scores and selects the top K candidates. Payments are calculated via the critical payment threshold determined by the $(K+1)$-th model's UCB score. Since each model provider has only one bid in the per-edge auction, they cannot influence the payment by manipulating other bids, as the $(K+1)$-th bid always comes from another provider. Subject to budget availability, the resulting inference feedback is used to update the learning parameters. The algorithm's complexity is dominated by sorting, i.e., $\mathcal{O}(N\log N)$.

Algorithm 2: Per-edge Model Auction Algorithm, $\forall i$

Input: B^{t-1}
Output: $\bar{\boldsymbol{x}}^t, \bar{\boldsymbol{p}}^t, \boldsymbol{\Phi}^t, \boldsymbol{r}^t$

Observe the bidding cost for edge i, i.e., $c^t_{i,n}, \forall n \in \mathcal{N}$, update $\bar{c}^t_{i,n}$ according to $\bar{c}^t_{i,n} = (\bar{c}^{t-1}_{i,n}(t-1) + c^t_{i,n})/t$;
Sort models in descending order of $\frac{\bar{r}^{t-1}_{i,n}}{\bar{c}^t_{i,n}} + u^{t-1}_{i,n}$;
Select the top K models as Φ^t_i, and set decisions: $\bar{x}^t_{i,n} = 1$ if $n \in \Phi^t_i$, and 0 otherwise;
Compute the payments for each selected model in Φ^t_i,

$$\text{i.e., } p^t_{i,n_j} = \begin{cases} \bar{r}^{t-1}_{i,n_j}/(\frac{\bar{r}^{t-1}_{i,n_{K+1}}}{\bar{c}^t_{i,n_{K+1}}} + u^{t-1}_{i,n_{K+1}} - u^{t-1}_{i,n_j}), \\ \quad \text{if } \frac{\bar{r}^{t-1}_{i,n_{K+1}}}{\bar{c}^t_{i,n_{K+1}}} + u^{t-1}_{i,n_{K+1}} > u^{t-1}_{i,n_j} \\ c_{\max}, \quad \text{otherwise} \end{cases}$$

if $\sum_i \sum_{n\in\Phi^t_i} p^t_{i,n} \geq B^{t-1}$ **then**
  return Terminate;
Receive inference feedback $L^t_{i,n}, v_{i,n}$ and calculate rewards $r^t_{i,n} = \chi - L^t_{i,n} - v_{i,n}$ for $n \in \Phi^t_i$;
Update $\beta^t_{i,n}$ and $\bar{r}^t_{i,n}$ via $\beta^t_{i,n} \leftarrow \beta^{t-1}_{i,n} + 1$ and $\bar{r}^t_{i,n} \leftarrow (\bar{r}^{t-1}_{i,n}\beta^{t-1}_{i,n} + r^t_{i,n})/\beta^t_{i,n}$ for $n \in \Phi^t_i$, $B^t = B^{t-1} - \sum_i \sum_{n\in\Phi^t_i} p^t_{i,n}$.

Algorithm 3: Online Validator Selection Algorithm

Input: Fractional solution $\widetilde{\boldsymbol{y}}^t$, dual solution $\boldsymbol{\lambda}^t$, and $\bar{\boldsymbol{x}}^t$
Output: Fractional solution $\widetilde{\boldsymbol{y}}^{t+1}$

Given $\bar{\boldsymbol{x}}^t$, observe $f^t(\widetilde{\boldsymbol{y}}^t)$ and $\mathbf{g}^t(\widetilde{\boldsymbol{y}}^t)$;
Update the dual variable $\boldsymbol{\lambda}^{t+1}$ according to (9);
Update validator decisions $\widetilde{\boldsymbol{y}}^{t+1}$ according to (8).

C. Validator Selection via Long-Term Online Learning

Classical OCO requires strict adherence to time-invariant constraints [45], which is not suitable for dynamic resource allocation. To accommodate temporal variations, we formulate a framework with time-varying long-term constraints, allowing for instantaneous violations. We consider a scenario where, at each slot t, the learner selects $\boldsymbol{y}^t \in \mathcal{X}$ before observing the loss $f^t(\cdot)$ and the penalty function $\mathbf{g}^t(\cdot)$, which introduces a time-varying constraint $\mathbf{g}^t(\cdot) \preceq \mathbf{0}$. Since $\mathbf{g}^t(\cdot) \preceq \mathbf{0}$ varies arbitrarily and is revealed only post-decision, strict per-slot satisfaction is impractical. Consequently, our goal is to minimize the cumulative loss while ensuring that $\mathbf{g}^t(\cdot) \preceq \mathbf{0}$ is satisfied on average over the long term. This formulation constitutes problem $\mathbb{P}_3$, the relaxed version of $\mathbb{P}_2$.

$$\mathbb{P}_3 : \min \textstyle\sum_{t\leq\tau_B} f^t(\boldsymbol{y}^t), \tag{6}$$

$$s.t. \quad h^t(\boldsymbol{y}^t) \leq 0, \tag{6a}$$

$$\textstyle\sum_{t\leq\tau_B} \mathbf{g}^t(\boldsymbol{y}^t) \preceq \mathbf{0}, \tag{6b}$$

$$\text{var.} \quad \boldsymbol{y}^t \in \mathcal{X} = \{\boldsymbol{y}^t | y^t_i \in [0,1], \forall i \in \mathcal{I}, \forall t \leq \tau_B\}.$$

Lagrangian Relaxation: Then we transform the problem $\mathbb{P}_3$ using an online saddle-point method. At each time t, only $f^t(\cdot)$ and $\mathbf{g}^t(\cdot)$ are observable, with no information available beyond time t. Moreover, the fluctuations in the variable e^t may lead to violations of Constraint (6b). To decouple long-term optimization without future dynamic information, (i.e., challenge ②), we introduce a Lagrange multiplier $\boldsymbol{\lambda}^t \in \mathbb{R}_+$ associated with the time-varying constraint $\mathbf{g}^t(\cdot)$, and derive its convex-concave version at t:

$$\begin{aligned}\min_{\boldsymbol{y}} \max_{\boldsymbol{\lambda}} \textstyle\sum_t \mathcal{L}^t(\boldsymbol{y}^t, \boldsymbol{\lambda}^t) &= \textstyle\sum_t (f^t(\boldsymbol{y}^t) + \boldsymbol{\lambda}^t \mathbf{g}^t(\boldsymbol{y}^t)), \\ s.t. \qquad h^t(\boldsymbol{y}^t) &\leq 0, \boldsymbol{y}^t \in \mathcal{X}, \end{aligned} \tag{7}$$

One-shot Problem Transformation: We solve the online Lagrangian (7) via an online primal-dual approach, which takes a modified descent step to obtain the fractional solution and a standard dual ascent step to update the dual variable. That is, given the previous primal iterate $\widetilde{\boldsymbol{y}}^t$ and the current dual iterate $\boldsymbol{\lambda}^{t+1}$ at each slot $t+1$, the current primal decision $\widetilde{\boldsymbol{y}}^{t+1}$ is obtained by minimizing:

$$\begin{aligned}\mathbb{P}_3^t : \min \nabla f^t(\widetilde{\boldsymbol{y}}^t)(\boldsymbol{y} - \widetilde{\boldsymbol{y}}^t) + \boldsymbol{\lambda}^{t+1}\mathbf{g}^t(\boldsymbol{y}) + \tfrac{\|\boldsymbol{y} - \widetilde{\boldsymbol{y}}^t\|^2}{2\gamma_2} \\ s.t. \qquad h^t(\boldsymbol{y}) \leq 0, \boldsymbol{y} \in \mathcal{X}, \end{aligned} \tag{8}$$

while the Lagrange multiplier $\boldsymbol{\lambda}^{t+1}$ is updated as follows:

$$\boldsymbol{\lambda}^{t+1} = [\boldsymbol{\lambda}^t + \gamma_1 \nabla_{\boldsymbol{\lambda}} \mathcal{L}^t(\widetilde{\boldsymbol{y}}^t, \boldsymbol{\lambda}^t)]^+ = [\boldsymbol{\lambda}^t + \gamma_1 \mathbf{g}^t(\widetilde{\boldsymbol{y}}^t)]^+, \tag{9}$$

where $\gamma_1 > 0$ and $\gamma_2 > 0$ are predefined step sizes; $\nabla f^t(\widetilde{\boldsymbol{y}}^t)$ is the gradient of $f^t(\boldsymbol{y})$ at $\boldsymbol{y} = \widetilde{\boldsymbol{y}}^t$; $\nabla_{\boldsymbol{\lambda}} \mathcal{L}^t(\widetilde{\boldsymbol{y}}^t, \boldsymbol{\lambda}^t)$ is the gradient of online Lagrangian $\mathcal{L}^t(\widetilde{\boldsymbol{y}}^t, \boldsymbol{\lambda})$ with respect to $\boldsymbol{\lambda}$ at $\boldsymbol{\lambda} = \boldsymbol{\lambda}^t$; and $[\cdot]^+ = \max\{\cdot, 0\}$. $\mathcal{L}^t(\boldsymbol{y}, \boldsymbol{\lambda}^{t+1})$ is approximated by $\nabla f^t(\widetilde{\boldsymbol{y}}^t)(\boldsymbol{y} - \widetilde{\boldsymbol{y}}^t) + \boldsymbol{\lambda}^{t+1}\mathbf{g}^t(\boldsymbol{y})$, and $\frac{1}{2\gamma_2}\|\boldsymbol{y} - \widetilde{\boldsymbol{y}}^t\|^2$ is a regularization and proximal term.

Here, we employ a primal-dual scheme where the primal step penalizes the exact constraint violation $\mathbf{g}^t(\boldsymbol{y})$ [25]. To compute $\boldsymbol{y}^{t+1}$ and $\boldsymbol{\lambda}^{t+1}$ at $t+1$, only inputs up to (excluding) $t+1$ are needed, avoiding reliance on *future* information. Via dual updates to the long-term constraints, it penalizes node over-selection, promoting load balancing and decentralization. The problem (8) can be solved using standard convex optimization solvers, which finds the ν-accurate optimal solution in $\mathcal{O}(I^2 \log(1/\nu))$ [54] via the interior-point method.

D. Validator Selection Decision Rounding

Algorithm 4 employs a randomized dependent rounding scheme to convert fractional $\widetilde{\boldsymbol{y}}^t$ from $\mathbb{P}_3$ into integers $\bar{\boldsymbol{y}}^t$, while ensuring the satisfaction of Constraint (4a) and $\mathbb{E}[\bar{\boldsymbol{y}}^t] = \widetilde{\boldsymbol{y}}^t$.

Fractions Scaling: This part as in Lines 2∼4 adjusts $\mathbf{U}^t$ in a randomized manner, ensuring that the sum of the adjusted values equals an integer and that the expectation of each adjusted value equals its corresponding value before such adjustment. This step either decreases each column of $\mathbf{U}^t$ by multiplying $\omega_1 \leq 1$ so that the sum of all columns in $\mathbf{V}^t$ is $\lfloor \mathbf{1}^\top \mathbf{U}^t \rfloor$, or increases each column of $\mathbf{U}^t$ by multiplying $\omega_2 \geq 1$ so that the sum of all columns in $\mathbf{V}^t$ is $\lceil \mathbf{1}^\top \mathbf{U}^t \rceil$. The probabilities of taking these two choices are $\lceil k \rceil - k$ and $k - \lfloor k \rfloor$, respectively, which ensures that $\mathbb{E}[\mathbf{V}^t] = \mathbf{U}^t$. We should mention here that the sum of all columns decreases to at least $\lfloor k \rfloor$, and D^t is an integer, such that Constraint (4a) $D^t \leq \sum_i y_i^t$ would not be violated, i.e., $h(\cdot) \leq 0$.

Algorithm 4: Randomized Rounding Algorithm

Input: Fractions $[\widetilde{y}_1^t, \ldots, \widetilde{y}_I^t]$
Output: Integers $[\bar{y}_1^t, \ldots, \bar{y}_I^t]$
1 ▷ Fractions scaling
2 Denote $\mathbf{U}^t = [\widetilde{y}_1^t, \ldots, \widetilde{y}_I^t]$, $k = \mathbf{1}^\top \mathbf{U}^t$;
3 $\omega_1 = \frac{\lfloor k \rfloor}{k}$, $\omega_2 = \frac{\lceil k \rceil}{k}$;
4 $\mathbf{V}^t = \begin{cases} [\omega_1 \cdot U_1^t, \ldots, \omega_1 \cdot U_I^t] \; with \; prob. \; \lceil k \rceil - k \\ [\omega_2 \cdot U_1^t, \ldots, \omega_2 \cdot U_I^t] \; with \; prob. \; k - \lfloor k \rfloor \end{cases}$
5 ▷ Fractions rounding
6 **while** $V_i^t \in (0,1) \wedge V_j^t \in (0,1)$ **do**
7 $\quad \varphi_1 = \min\{1 - V_i^t, V_j^t\}, \varphi_2 = \min\{V_i^t, 1 - V_j^t\}$;
8 $\quad (V_i^t, V_j^t) = \begin{cases} (V_i^t + \varphi_1, V_j^t - \varphi_1) \; with \; prob. \; \frac{\varphi_2}{\varphi_1 + \varphi_2} \\ (V_i^t - \varphi_2, V_j^t + \varphi_2) \; with \; prob. \; \frac{\varphi_1}{\varphi_1 + \varphi_2} \end{cases}$
9 **Return** $\bar{\boldsymbol{y}}^t = [V_1^t, \ldots, V_I^t]$.

Fractions Rounding: This part as in Lines 6∼9 rounds $\mathbf{V}^t$ into integers, also in a randomized manner, while guaranteeing that the sum of all the values stays unchanged after rounding, each value becomes an integer after rounding, and the expectation of each randomized integer equals its corresponding fractional value before rounding. We use the values of $\mathbf{V}^t$ as the probability to round the columns in pairs into integers, while letting the two fractions compensate for each other. Since the sum of all columns is an integer beforehand, as a result of the previous step, $\mathbf{V}^t$ can be guaranteed as a vector that only contains 0 and 1 after the loop. In Line 8, the sum is preserved, because $V_i^t + V_j^t = (V_i^t + \varphi_1) + (V_j^t - \varphi_1) = V_i^t + V_j^t$, for example. Also, the expectation is preserved. That is, for every i, for the first case in Line 8, we have $E(V_i^t) = (\lceil k \rceil - k)\frac{\varphi_2}{\varphi_1+\varphi_2}(\omega_1 \widetilde{y}_i^t + \varphi_1) + (\lceil k \rceil - k)\frac{\varphi_1}{\varphi_1+\varphi_2}(\omega_1 \widetilde{y}_i^t - \varphi_2) + (k - \lfloor k \rfloor)\frac{\varphi_2}{\varphi_1+\varphi_2}(\omega_2 \widetilde{y}_i^t + \varphi_1) + (k - \lfloor k \rfloor)\frac{\varphi_1}{\varphi_1+\varphi_2}(\omega_2 \widetilde{y}_i^t - \varphi_2) = \widetilde{y}_i^t$; and it is analogous for the second case. Lines 7 and 8 guarantee that in each iteration of the loop, at least one fraction will be rounded to 0 or 1. The complexity of the loop is $\mathcal{O}(I)$ [55].

IV. Theoretical Analysis

A. Regret and Fit Analysis

In Theorem 2, we define the regret experienced by each edge node in the problem $\mathbb{P}_1$. The regret refers to the gap between the inference cost incurred by our algorithm and those incurred by the best set of K models in hindsight. The result shows that the regret grows sub-linearly with time.

Theorem 2 *The expected regret for $\mathbb{P}_1$ from Algorithm 2 is bounded as* $\mathrm{Reg}_b^T := \mathbb{E}[\sum_{t \leq \tau_B} \sum_i \sum_n \bar{x}_{i,n}^t (l_n + v_{i,n})]$ $- \mathbb{E}[\sum_{t \leq \tilde{\tau}_B^*} \sum_i \sum_n \tilde{x}_{i,n}^{t*} (l_n + v_{i,n})] \leq \mathcal{O}(\log \tau_B)$, *where $\tilde{x}_{i,n}^{t*}$ is the optimum of $\mathbb{P}_1$ with stopping time $\tilde{\tau}_B^*$.*

Proof. **Step I: Auxiliary Definitions.** We first define the gaps in cost-efficiency and reward between the optimal set Φ_i^* and any non-optimal set Φ_i^t:

$$\begin{aligned}\Delta_{i,\max} &= \textstyle\sum_{n \in \Phi_i^*} \frac{r_{i,n}}{c_{i,n}} - \min_{\Phi_i^t \neq \Phi_i^*} \sum_{n \in \Phi_i^t} \frac{r_{i,n}}{c_{i,n}} \\ \Delta_{i,\min} &= \textstyle\sum_{n \in \Phi_i^*} \frac{r_{i,n}}{c_{i,n}} - \max_{\Phi_i^t \neq \Phi_i^*} \sum_{n \in \Phi_i^t} \frac{r_{i,n}}{c_{i,n}} \\ \nabla_{i,\max} &= \textstyle\sum_{n \in \Phi_i^*} r_{i,n} - \min_{\Phi_i^t \neq \Phi_i^*} \sum_{n \in \Phi_i^t} r_{i,n}. \end{aligned}$$

Let τ_B^* be the fixed stopping time for the optimal solution (which has prior knowledge of means). With the optimal cost per round $C^* = \sum_i \sum_{n\in\Phi_i^*} c_{i,n}$, we have:

$$\tau_B^* = \lfloor B/C^* \rfloor \implies B/C^* - 1 \le \tau_B^* \le B/C^*.$$

To bound non-optimal selections [52], we introduce counters $\alpha_{i,n}^t$. Whenever $\Phi_i^t \ne \Phi_i^*$, we increment the minimum counter $n' = \arg\min_{n\in\Phi_i^t} \alpha_{i,n}^t$ via $\alpha_{i,n'}^t = \alpha_{i,n'}^t + 1$. By definition, the actual play count satisfies $\beta_{i,n}^t \ge \alpha_{i,n}^t$, and the total counter sum tracks the number of suboptimal rounds:

$$\textstyle\sum_{n=1}^N \alpha_{i,n}^t = \sum_{s=1}^t \mathbb{I}(\Phi_i^s \ne \Phi_i^*) \quad \forall\, t \le \tau_B.$$

Step II: Regret for $r_{i,n}^t$. Based on Lemmas 1 and 2 and their proofs in our Supplementary Material, we derive the regret Reg_B^T for $r_{i,n}^t$ by decomposition:

$$\begin{aligned}
\mathrm{Reg}_B^T &= \tilde{R}^* \tfrac{B}{\tilde{C}^*} - \mathbb{E}[\textstyle\sum_{t=1}^{\tau_B} \sum_i \sum_{n\in\Phi_i^t} r_{i,n}^t] \\
&= \tilde{R}^* \tfrac{B}{\tilde{C}^*} - \tau_B \tilde{R}^* + (\tau_B \tilde{R}^* - \mathbb{E}[\textstyle\sum_{t=1}^{\tau_B} \sum_i \sum_{n\in\Phi_i^t} r_{i,n}^t]) \\
&\le \tilde{R}^* \tfrac{B}{\tilde{C}^*} - \tau_B \tilde{R}^* + \textstyle\sum_i \sum_{n=1}^N \alpha_{i,n}^{\tau_B} \nabla_{i,\max} \\
&\le (\tilde{R}^* \tfrac{B}{\tilde{C}^*} - \tau_B \tilde{R}^*) \\
&\quad + N \textstyle\sum_i \nabla_{i,\max} (\varphi_{1,i} \log(\tfrac{2B}{C^*} + \varphi_3) + \varphi_2) \\
&\le \tilde{R}^* \tfrac{B}{\tilde{C}^*} - (\tfrac{B}{C^*} - \varphi_4) \tilde{R}^* + \mathcal{O}(\log \tau_B^*) \\
&\overset{(a)}{\le} \mathcal{O}(\log \tau_B) + \varphi_4 \tilde{R}^* + \mathcal{O}(\log \tau_B^*) = \mathcal{O}(\log \tau_B),
\end{aligned}$$

where (a) holds since $\tilde{\tau}_B^* - \tau_B^* \le \tau_B - \tau_B^* \le \mathcal{O}(\log \tau_B)$; $\tilde{R}^* = \sum_i \sum_{n\in\tilde{\Phi}_i^*} r_{i,n}$, $\tilde{C}^* = \sum_i \sum_{n\in\tilde{\Phi}_i^*} c_{i,n}$, $\tilde{\Phi}_i^*$ is the optimal set of models of $\mathbb{P}_1$; $\varphi_{1,i} = (K+1)(\frac{2K\sigma(c_{\min}+r_{\max})}{(c_{\min})^2 \Delta_{i,\min}})^2$, $\varphi_2 = 1 + 2K\frac{\pi^2}{3}$, $\varphi_3 = \frac{2Nc_{\max}}{IKc_{\min}} \sum_i (\varphi_2 - \varphi_{1,i} + \varphi_{1,i} \log(\frac{2Nc_{\max}\varphi_{1,i}}{IKc_{\min}}))$, $\varphi_4 = \frac{Nc_{\max} \sum_i (\varphi_{1,i} \log(\frac{2B}{C^*} + \varphi_3) + \varphi_2)}{C^*} + 1$.

Step III: Regret for $L_{i,n}^t$. Substituting $r_{i,n}^t = \chi - (L_{i,n}^t + v_{i,n})$, we relate the regret in terms of loss and latency to the reward regret Reg_B^T derived in Step II:

$$\begin{aligned}
&\mathbb{E}[\textstyle\sum_{t\le\tau_B} \sum_{i,n} \bar{x}_{i,n}^t (L_{i,n}^t + v_{i,n})] \\
&\quad - \mathbb{E}[\textstyle\sum_{t\le\tilde{\tau}_B^*} \sum_{i,n} \tilde{x}_{i,n}^{t*} (L_{i,n}^t + v_{i,n})] \\
&= \underbrace{\mathrm{Reg}_B^T}_{\le \mathcal{O}(\log \tau_B)} + \chi \cdot \mathbb{E}[\textstyle\sum_{t\le\tau_B} \sum_{i,n} \bar{x}_{i,n}^t - \sum_{t\le\tilde{\tau}_B^*} \sum_{i,n} \tilde{x}_{i,n}^{t*}] \\
&\le \mathcal{O}(\log \tau_B) + \chi \cdot K \cdot \mathbb{E}[|\tau_B - \tilde{\tau}_B^*|] = \mathcal{O}(\log \tau_B),
\end{aligned}$$

where the last equality holds because the stopping time difference is logarithmic, i.e., $|\tau_B - \tilde{\tau}_B^*| \le \mathcal{O}(\log \tau_B)$.

Step IV: Regret for ℓ_n. We need to prove that $\mathbb{E}[L_{i,n}] = \mathbb{E}[\ell_n]$. It can be easily proved that

$$\mathbb{E}[\tfrac{1}{D} \textstyle\sum_{i=1}^D X_i \,|D = d] = \mathbb{E}[\tfrac{1}{d} \sum_{i=1}^d X_i] = \tfrac{1}{d} \sum_{i=1}^d \mathbb{E}[X_i] = \mu$$

$$\mathbb{E}[\tfrac{1}{D} \textstyle\sum_{i=1}^D X_i] = \mathbb{E}[\mathbb{E}[\tfrac{1}{D} \sum_{i=1}^D X_i \,|D]] = \mathbb{E}[\mu] = \mathbb{E}[X],$$

where $X_i, \forall i$ all have the mean as μ, and D is a random integer independent of X_i. Due to $L_{i,n} = \frac{1}{M_i} \sum_{\nu=1}^{M_i} \ell_{n,\nu}$, we have $\mathbb{E}[L_{i,n}] = \mathbb{E}[\frac{1}{M_i} \sum_{\nu=1}^{M_i} \ell_{n,\nu}] = \mathbb{E}[\ell_n]$. Then, $\mathrm{Reg}_b^T = \frac{1}{K} \mathbb{E}[\sum_{t\le\tau_B} \sum_i \sum_n \bar{x}_{i,n}^t (\ell_n + v_{i,n})] - \frac{1}{K} \mathbb{E}[\sum_{t\le\tilde{\tau}_B^*} \sum_i \sum_n \tilde{x}_{i,n}^{t*} (\ell_n + v_{i,n})] \le \mathcal{O}(\log \tau_B)$. ☐

Remark 1 (Scaling with N) *Tracing the constants in Lemmas 1–2, $\varphi_{1,i}$ and φ_2 are independent of N, $\varphi_3 = \mathcal{O}(N \log N)$, and $\varphi_4 = \mathcal{O}\big(N \log(\tau_B + N \log N)\big)$. Substituting these into the regret decomposition of Theorem 2 yields $\mathrm{Reg}_b^T = \mathcal{O}\big(N \log(\tau_B + N \log N)\big)$. Thus, the regret scales near-linearly in N (up to a logarithmic factor) and only logarithmically in τ_B. This matches the standard scaling of combinatorial multi-play bandits [41], [52], indicating that AucLearn remains efficient as the marketplace expands.*

In Theorem 3, we introduce the regret and fit for problem $\mathbb{P}_2$. The regret quantifies the gap between the objective value achieved by our approach and the cumulative one-shot optimal values, while the fit measures the violation of constraint (4b). Our analysis shows that regret and fit grow only sub-linearly.

Theorem 3 *The regret and the fit for $\mathbb{P}_2$ satisfy*

$$\mathrm{Reg}_o^T := \mathbb{E}[\textstyle\sum_{t\le\tau_B} f^t(\bar{\boldsymbol{y}}^t)] - \sum_{t\le\tau_B} f^t(\bar{\boldsymbol{y}}^{t*}) \le \mathcal{O}(\tau_B^{\frac{2}{3}}) + Q\sigma_\beta^\beta,$$
$$\mathrm{Fit}_o^T := \|\mathbb{E}[\textstyle\sum_{t\le\tau_B} \mathbf{g}^t(\bar{\boldsymbol{y}}^t)]^+\| \le \mathcal{O}(\tau_B^{\frac{2}{3}}),$$

where for each t, $\bar{\boldsymbol{y}}^t$ represents the output of Algorithm 4; $\bar{\boldsymbol{y}}^{t*} \in \arg\min_{\boldsymbol{y}\in\mathcal{X}^t} f^t(\boldsymbol{y})$; $\mathcal{X}^t := \{\boldsymbol{y} | h^t(\boldsymbol{y}) \le 0, \mathbf{g}^t(\boldsymbol{y}) \preceq \mathbf{0}; y_i^t \in \{0,1\}, \forall i\}, \forall t \le \tau_B$; and Q and σ_β^β are constants from the Jensen Gap.

Proof. We first define the regret $\widetilde{\mathrm{Reg}}_o^T$ and the fit $\widetilde{\mathrm{Fit}}_o^T$ for the relaxed problem $\mathbb{P}_3$:

$$\begin{aligned}
\widetilde{\mathrm{Reg}}_o^T &:= \textstyle\sum_{t\le\tau_B} f^t(\widetilde{\boldsymbol{y}}^t) - \sum_{t\le\tau_B} f^t(\widetilde{\boldsymbol{y}}^{t*}), \\
\widetilde{\mathrm{Fit}}_o^T &:= \|[\textstyle\sum_{t\le\tau_B} \mathbf{g}^t(\widetilde{\boldsymbol{y}}^t)]^+\|, \widetilde{\boldsymbol{y}}^t \in \widetilde{\mathcal{X}}^t, \forall t \le \tau_B.
\end{aligned}$$

Here, $\widetilde{\boldsymbol{y}}^{t*}$ is the fractional optimal solution, and $\widetilde{\boldsymbol{y}}^{t*} \in \arg\min_{\boldsymbol{y}^t\in\widetilde{\mathcal{X}}^t} f^t(\boldsymbol{y}^t)$, where $\widetilde{\mathcal{X}}^t := \{\boldsymbol{y} | h^t(\boldsymbol{y}) \le 0, \mathbf{g}^t(\boldsymbol{y}) \preceq \mathbf{0}; y_i^t \in [0,1], \forall i\}, \forall t \le \tau_B$. Based on Lemma 3 and its proof in our Supplementary Material, we have

$$\mathrm{Reg}_o^T \le \widetilde{\mathrm{Reg}}_o^T + Q\sigma_\beta^\beta, \mathrm{Fit}_o^T \le \widetilde{\mathrm{Fit}}_o^T.$$

Next, following [25] and its standard common assumptions, we can derive bounds for $\mathbb{P}_3$. Specifically, setting step sizes $\gamma_1 = \gamma_2 = \mathcal{O}(\tau_B^{-\frac{1}{3}})$ yields $\mathrm{Reg}_o^T \le \widetilde{\mathrm{Reg}}_o^T + Q\sigma_\beta^\beta = \mathcal{O}(\max\{V(\{\widetilde{\boldsymbol{y}}^{t*}\}_{t\le\tau_B})\tau_B^{\frac{1}{3}}, V(\{\mathbf{g}^t\}_{t\le\tau_B})\tau_B^{\frac{1}{3}}, \tau_B^{\frac{2}{3}}\}) + Q\sigma_\beta^\beta$ and $\mathrm{Fit}_o^T \le \widetilde{\mathrm{Fit}}_o^T = \mathcal{O}(\tau_B^{\frac{2}{3}})$. Consequently, both metrics achieve sub-linear convergence of $\mathcal{O}(\tau_B^{\frac{2}{3}})$, provided the variations satisfy $V(\{\widetilde{\boldsymbol{y}}^{t*}\}_{t\le\tau_B})$ and $V(\{\mathbf{g}^t\}_{t\le\tau_B}) \in \boldsymbol{o}(\tau_B^{\frac{2}{3}})$. ☐

Theorem 4 establishes the global regret for $\mathbb{P}_0$. Note that this derivation is non-trivial: a simple summation of Theorems 2 and 3 is invalid due to their divergent optimal baselines.

Theorem 4 *The regret for $\mathbb{P}_0$ satisfies*

$$\mathrm{Reg}^T \le \mathcal{O}(\tau_B^{\frac{2}{3}}) + \mathcal{O}\left(\log \tau_B\right).$$

Proof Sketch. We decompose Reg^T into three distinct components via auxiliary terms based on the decoupling of model auction decisions ($\boldsymbol{x}$) and validator selection ($\boldsymbol{y}$). *First*, we isolate the regret from validator selection, which maps to Subproblem $\mathbb{P}_2$. Leveraging the analysis of Theorem 3, this component is bounded by $\mathcal{O}(\tau_B^{\frac{2}{3}})$. *Second*, we isolate the regret from model auction decisions, corresponding to Subproblem $\mathbb{P}_1$. Following the analysis of Theorem 2, both the regret and the gap caused by the stopping time difference are bounded by $\mathcal{O}(\log \tau_B)$. *Finally*, the residual coupling term is bounded by $\mathcal{O}(1)$. Detailed proof is in our Supplementary Material. □

B. Economic Guarantees Analysis

We define *truthfulness* (incentivizing honest bidding) and *individual rationality* (guaranteeing no loss for each bid) based on *utility*. Our mechanism attains both economic guarantees.

Definition 1 (Utility) *The utility of the bidder n's i-th bid at time t is*

$$U_{i,n}(b_{i,n}^t, \boldsymbol{b}_{i,-n}^t) = \begin{cases} p_{i,n}^t(b_{i,n}^t, \boldsymbol{b}_{i,-n}^t) - c_{i,n}^t, \\ \quad \text{if } x_{i,n}^t = 1 \\ 0 \text{ , otherwise} \end{cases}$$

where $b_{i,n}^t$ is the bidding price of i-th bid submitted by bidder n at time t; $\boldsymbol{b}_{i,-n}^t$ denotes the set of bidding prices of i-th bid submitted by all bidders except bidder n; and $c_{i,n}^t$ is the true cost associated with i-th bid from bidder n.

Definition 2 (Truthfulness) *An auction is truthful in expectation if each bid maximizes its expected utility by bidding its true cost, i.e., $\mathbb{E}[U_{i,n}^t(c_{i,n}^t, \boldsymbol{b}_{i,-n}^t)] \geq \mathbb{E}[U_{i,n}^t(b_{i,n}^t, \boldsymbol{b}_{i,-n}^t)]$, $\forall i, n$.*

Definition 3 (Individual Rationality) *An auction is individually rational in expectation if each bid always achieves a non-negative expected utility, i.e., $\mathbb{E}[U_{i,n}^t(b_{i,n}^t, \boldsymbol{b}_{i,-n}^t)] \geq 0$, $\forall i, n$.*

Theorem 5 *Our auction is truthful in expectation.*

Proof. According to Myerson's theorem [53], an auction is truthful if and only if (I) the winning bid selection process is monotonic and (II) each winning bid is paid the critical value.

Condition (I): This holds because the ranking score $\frac{\bar{r}_{i,n}^{t-1}}{\bar{c}_{i,n}^t} + u_{i,n}^{t-1}$ is strictly decreasing with respect to the bid $c_{i,n}^t$. Thus, if a bid $c_{i,n}^t$ wins, any lower bid $\tilde{c}_{i,n}^t < c_{i,n}^t$ yields a higher score and preserves the winning status.

Condition (II): Consider a winning bid $\bar{c}_{i,n}^t$. If model n is removed, the K-th position would be filled by the original $(K+1)$-th candidate, $\bar{c}_{i,K+1}^t$, based on the sorted ranking scores. Thus, given other bids, the critical payment $p_{i,n}^t$ is: $p_{i,n}^t = \bar{r}_{i,n}^{t-1}/(\frac{\bar{r}_{i,K+1}^{t-1}}{\bar{c}_{i,K+1}^t} + u_{i,K+1}^{t-1} - u_{i,n}^{t-1})$. If $\bar{c}_{i,n}^t \leq p_{i,n}^t$, then the criterion $\frac{\bar{r}_{i,n}^{t-1}}{\bar{c}_{i,n}^t} + u_{i,n}^{t-1}$ will be greater than $\frac{\bar{r}_{i,K+1}^{t-1}}{\bar{c}_{i,K+1}^t} + u_{i,K+1}^{t-1}$, ensuring selection; otherwise, if $\bar{c}_{i,n}^t > p_{i,n}^t$, it falls below the threshold and loses the auction. Crucially, since each provider submits a single bid per edge, the threshold (determined by the $(K+1)$-th bidder) is independent of the winner's own bid. This confirms that $p_{i,n}^t$ is the critical payment. □

Theorem 6 *Our auction is individually rational in expectation.*

Proof. Since the bid $\bar{c}_{i,n}^t$ is selected prior to $\bar{c}_{i,K+1}^t$ in time slot t, we get $\frac{\bar{r}_{i,n}^{t-1}}{\bar{c}_{i,n}^t} + u_{i,n}^{t-1} \geq \frac{\bar{r}_{i,K+1}^{t-1}}{\bar{c}_{i,K+1}^t} + u_{i,K+1}^{t-1}$ based on the selection criteria. Then, we have $\bar{c}_{i,n}^t \leq \bar{r}_{i,n}^{t-1}/(\frac{\bar{r}_{i,K+1}^{t-1}}{\bar{c}_{i,K+1}^t} + u_{i,K+1}^{t-1} - u_{i,n}^{t-1}) = p_{i,n}^t$, i.e., $p_{i,n}^t - \bar{c}_{i,n}^t \geq 0$, in any time slot t. Since $c_{i,n}^t$ follows an i.i.d. distribution and $\mathbb{E}[|c_{i,n}^t|] \leq \infty$, according to the strong law of large numbers, when t is sufficiently large, the empirical mean converges almost surely to the expected value of the sample, i.e., $\bar{c}_{i,n}^t \xrightarrow{a.s.} c_{i,n}$. Then we have $\mathbb{E}[p_{i,n}^t] \geq \mathbb{E}[\bar{c}_{i,n}^t] \Rightarrow \mathbb{E}[p_{i,n}^t] \geq \mathbb{E}[c_{i,n}]$. □

TABLE I: Datasets and corresponding models

Dataset	Type 1	Type 2	Type 3
MNIST	CNNs*	LeNet-5 [37]	–
CIFAR-10	CNNs	MobileNet V1/V2 [38]	ResNet-32/50 [39]
CIFAR-100	CNNs	DenseNet-100x12/24 [40]	ResNet-32/50 [39]

**Note: The CNNs refer to three CNN variants with identical structures but different channel widths (32, 64, and 128). Each variant has two 3×3 convolutional layers with ReLU activation and 2×2 max-pooling, followed by one fully-connected layer and a softmax output.*

TABLE II: Models and corresponding parameters

Model	CNN	LeNet-5	Mob.V1	Mob.V2	Dense-100	ResNet
$v_{i,n}$ (ms)	3.0-8.0	0.5-2.0	8.9-37.0	11.3-86.1	26.4-70.7	10.7-145.5
α_n (GFLOPs)	0.12	0.001	0.55	0.39	1.79	4.15

V. Experimental Study

A. Experimental Settings

Data and Models: We adopt the MNIST [28] dataset and the CIFAR-10/100 [29] dataset. We consider 6 models of 2∼3 types for each dataset, as detailed in Table I.

Edge and Inference Workload: We consider the top 10∼400 London underground stations with the highest passenger counts [36], and use such dynamic passenger counts to represent the inference workload at each edge. The passenger data are measured for every 15 minutes on a weekend in 2023, covering from 5:00 AM to 1:00 AM on the following day. For each underground station, i.e., each edge node, we randomly sample 8000 data points from the $10,000$ test data of our datasets and use those as the incoming data streams.

Costs and Bids: Based on the real computational resource prices from AWS [5], combined with minimum and recommended specifications to run an XRP Ledger full node [30], we set the operational cost of the node as $[0.24, 0.72]\$/h$. Accordingly, we set $\delta = 66.67$ CPU units per dollar and $A^t \in [10, 15]$ CPU units. We set $k = 5$ and $d_0 = 0.35$. We estimate the FLOPs consumed per transaction based on the energy consumption [56] and the minimum specification of the XRP Ledger, resulting in 0.70 GFLOPs. The computational latency and the computational resources consumed for inferring a single data sample are detailed in Table II [31]–[33]. The computational capacity of each edge is set to 400 GFLOPs by default or be adjusted based on computational capabilities in practice. The bidding price is $[5, 15]\,\$$, following the raw electricity required to develop and deploy the model [35].

Algorithms: We implement multiple algorithms. First, for model auctions, we consider the following:

- `AUCB` [41] considers a deterministic bidding cost and uses an exploration index $u_{i,n}^t = \sqrt{(K+1)\log t/\beta_{i,n}^t}/\bar{c}_{i,n}^t$.
- `CMABA` [42] uses $(\kappa N c_{\max} \ln(\frac{B}{c_{\max}}))^{\frac{1}{3}} B^{\frac{2}{3}}$ periods for exploration and directly adopts the learned results in the

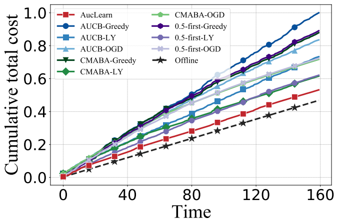


Fig. 6: Cumulative total cost

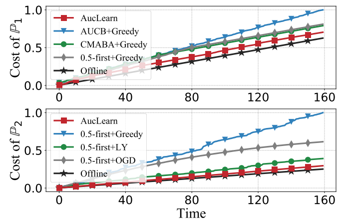


Fig. 7: Cost of $\mathbb{P}_1$ and $\mathbb{P}_2$

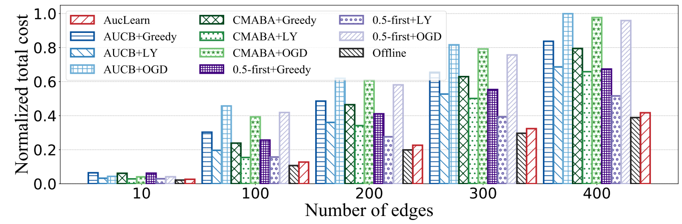


Fig. 8: Scalability

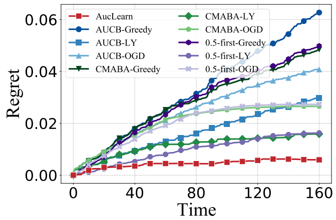


Fig. 9: Regret

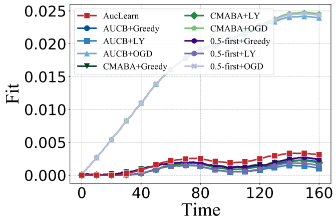


Fig. 10: Fit

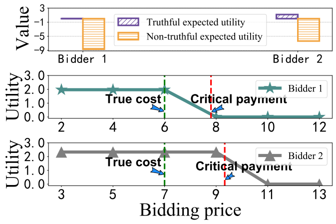


Fig. 11: Utility

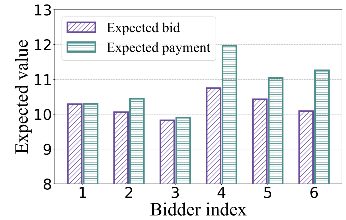


Fig. 12: Payment vs. cost

- remaining exploitation phase, with an exploration index $u_{i,n}^t = \sqrt{\kappa \log t / \beta_{i,n}^t} / \bar{c}_{i,n}^t$, where κ is a hyper parameter.
- `the ε-first` [43] uses the first $\varepsilon \cdot B$ budget as exploration to randomly select K winning models in each time slot. In the remaining $(1-\varepsilon) \cdot B$ budget, the player will always select the top K models in the decreasing order of $\frac{\bar{r}_{i,n}^{t-1}}{\bar{c}_{i,n}^t}$ without any exploration index.

Since the above algorithms are not applicable to the auction scenario where the bids follow an unknown distribution as in this paper, we have adapted them via $\bar{c}_{i,n}^t$ to this setting.

For validator selection, we consider the following:

- `Greedy` selects edge nodes with the lowest cost as validators until the constraint $\sum_i y_i^t \geq D^t, \forall t$ is met.
- `Lyapunov` [44] solves time-averaged stochastic optimizations using virtual queues and drift-plus-penalty. Since the original Lyapunov requires inputs at t to decide $\boldsymbol{y}^t$, we modify it for fairness in our online setting by using the faulty computational resources at $t-1$ to obtain $\boldsymbol{y}^t$.
- `OGD` [45] uses gradient descent to solve the decision variables, without considering long-term constraints.

By pairwise combining the above two groups of methods, we obtain the following nine baseline schemes for comparison: `AUCB+Greedy`, `AUCB+LY`, `AUCB+OGD`, `CMABA+Greedy`, `CMABA+LY`, `CMABA+OGD`, `0.5-first+Greedy`, `0.5-first+LY`, and `0.5-first+OGD`. These pairings are constructed because no single existing baseline natively covers both subproblems; this 3×3 design isolates AucLearn's advantage from being attributable to either dimension alone. We also consider an `Offline` benchmark with full future knowledge. It selects models greedily using $r_{i,n}/c_{i,n}$ (setting $p_{i,n} = c_{i,n}$ [41]), and optimizes validator selection via CasADi [57], yielding an optimal fractional solution to $\mathbb{P}_2$.

B. Experimental Results

Real-Time Cost: Fig. 6 shows the normalized total cost of different algorithms in the cumulative manner as time goes. `AucLearn` has a slower growth trend of the cumulative total cost compared to the others and is closest to the `Offline`.

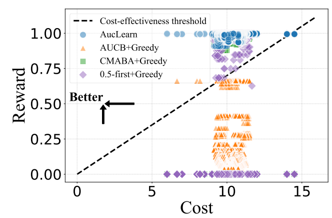


Fig. 13: Cost-reward

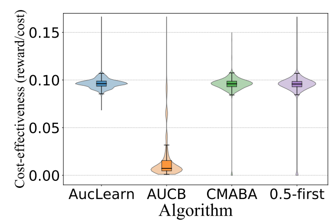


Fig. 14: Effectiveness

Fig. 7 depicts the cumulative objective values for $\mathbb{P}_1$ and $\mathbb{P}_2$ against relevant baselines. The subplots display the normalized inference cost (reflecting model quality) and the blockchain operational cost and risk, respectively. In both cases, `AucLearn` consistently outperforms other online methods and closely tracks the `Offline`, demonstrating superior performance second only to the offline optimum.

Scalability: Fig. 8 presents the normalized average total cost. As the number of the edge nodes increases, `AucLearn` reduces the total cost on average by 54%, 34%, 59%, 50%, 26%, 57%, 47%, 17%, and 56%, respectively, compared to `AUCB+Greedy`, `AUCB+LY`, `AUCB+OGD`, `CMABA+Greedy`, `CMABA+LY`, `CMABA+OGD`, `0.5-first+Greedy`, `0.5-first+LY`, and `0.5-first+OGD`.

Regret & Fit: Fig. 9 and 10 show the regret and the fit of different algorithms for the problem $\mathbb{P}_0$ as the total time range varies. `AucLearn` achieves the lowest regret with sub-linear growth, consistently outperforming all baselines. While methods like `AUCB+Greedy` yield lower fit values by prioritizing low-resource models, they suffer from higher inference costs. `AucLearn` shows sub-linear fit, representing a superior trade-off between constraint satisfaction and inference performance.

Economic Properties: Fig. 11 validates the truthfulness of our auction mechanism. To show this in expectation, we analyze the utilities of two randomly selected bidders under truthful and non-truthful cases. In the latter case, bids are uniformly sampled from $[5, 15]$ excluding the true value. Truthful bidding yields utilities of 0.006 and 1.171; and non-truthful bidding leads to utilities of -8.696 and -6.372.

Fig. 12 illustrates the individual rationality in expectation of our auction mechanism. We present the expected bid and expected payment for each bid. The results confirm that the

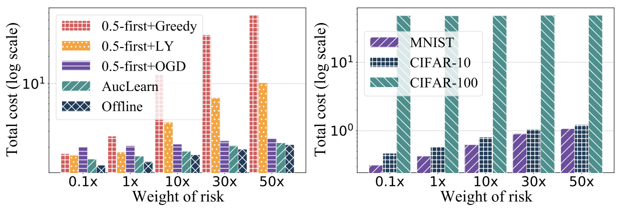


Fig. 15: Impact of risk

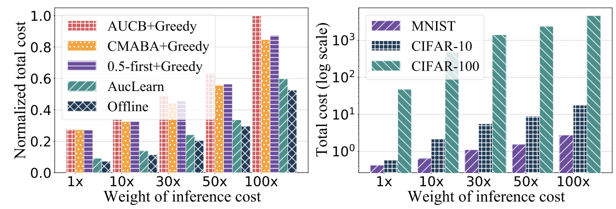


Fig. 16: Impact of inference cost

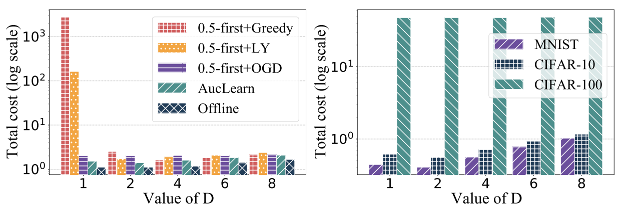


Fig. 17: Impact of D

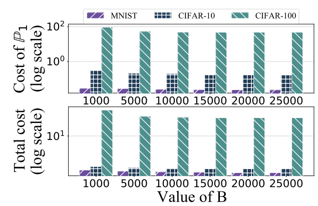


Fig. 18: Impact of B

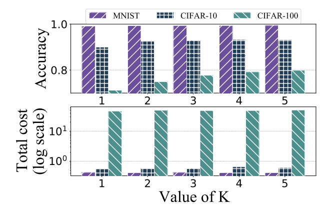


Fig. 19: Impact of K

expected payment for each bid is greater than or equal to its expected bid, implying non-negative expected utility.

Distribution of Cost-Effectiveness: Fig. 13 illustrates the decision distributions of various algorithms in terms of reward and cost. The scatter plot maps the average empirical cost (x-axis) against reward (y-axis) for each model selection. `AucLearn`'s decisions are predominantly in the optimal upper-left region (high reward, low cost), whereas `AUCB+Greedy`'s fall largely within the non-optimal zone.

Fig. 14 details the distributions of the "reward over cost" ratios. The violin plot depicts the probability density, with the width representing data concentration. `AucLearn` exhibits the highest density in the high cost-effectiveness region. `AUCB+Greedy` is clustered in the low cost-effectiveness region with the lowest median. `CMABA+Greedy` and `0.5-first+Greedy` slightly trail behind `AucLearn`.

Impact of Different Factors: Fig. 15 visualizes the normalized total cost as the weight of blockchain operational risk increases. As the weight rises, total costs increase because algorithms must select more validators to mitigate risk, thereby inflating the $\mathbb{P}_2$ cost. While the cost of `0.5-first+Greedy` grows exponentially, `AucLearn` exhibits greater adaptability, maintaining significantly lower total costs. This conclusion is further confirmed by results on different datasets.

Fig. 16 depicts the impact of the inference cost weight. As the weight increases, the total cost for all algorithms rises. `AucLearn` consistently tracks the `Offline` most closely, highlighting its superior model selection and inference performance. Consistent results are observed for different datasets.

Fig. 17 compares performance under different minimum validator counts D. At $D = 1$, operational risk dominates, causing exponential cost growth for `0.5-first+Greedy` and `0.5-first+LY`. Increasing D to 2 mitigates this risk, reducing total costs for these baselines. Yet, as D rises to 8, the blockchain operational cost becomes dominant, leading to increased total costs. Throughout these variations, `AucLearn` consistently outperforms competitors and remains closest to the `Offline`. Results on different datasets remain consistent.

Fig. 18 illustrates the impact of the auction budget B across MNIST, CIFAR-10, and CIFAR-100. As the budget increases, inference costs decrease; however, improvements become negligible once the budget exceeds $20,000$. Consequently, we set the budget to $20,000$ to ensure sufficient bandit exploration for effective inference while balancing resource conservation.

Fig. 19 presents the impact incurred by varying the number of selected models. As the objective accounts for the average loss, increasing K leads to negligible changes to the total cost. When $K > 1$, we employ a soft majority voting mechanism to obtain the inference label, meaning that incorporating more models enhances the inference accuracy. Accuracy stabilizes at $K = 4$, indicating limited marginal benefit beyond this point.

Inference Accuracy: Table III reports the average inference accuracy. `AucLearn` outperforms baselines by 0.06% to 0.11% on MNIST, 1.56% to 3.18% on CIFAR-10, and 0.94% to 4.62% on CIFAR-100, achieving the highest accuracy among all online methods and second only to the `Offline`. This further shows `AucLearn`'s superiority in achieving high accuracy while effectively minimizing total cost.

VI. Related Work

Model Management and Inference in Cloud/Edge: Recent literature has extensively explored DNN model management in edge networks, focusing primarily on model partitioning and collaborative inference [11], [12]. Optimization frameworks have been proposed to balance the trade-offs between accuracy, latency, and resource costs via joint resource provisioning [13] or parameter sharing [14]. To enhance security and privacy, blockchain technologies have been integrated into collaborative inference frameworks for IoT and video analytics [15], [16]. Yet, these works generally focus on *offline* problems. They overlook the online model selection and auction problems in edge AI, failing to address the dynamically-changing environments and the real-time decision-making.

Trading and Incentive in Cloud/Edge: Resource trading in cloud/edge scenarios has been studied using various economic models, including platform-driven pricing [17], game-theoretic approaches [19], [20], and auction mechanisms [18]. With the advent of blockchain, decentralized marketplaces have adopted smart contracts to implement tamper-proof double auctions for resource sharing [21], [22] or fair incentive allocation in federated learning [23]. However, existing mechanisms

TABLE III: Average inference accuracy (%) of algorithms (± the standard deviation of the accuracy for each time slot). • indicates that `AucLearn` performs the best except for `Offline`. The values in boldface indicate the highest accuracy on the corresponding dataset.

Dataset	`AucLearn`	`AUCB+Greedy`	`CMABA+Greedy`	`0.5-first+Greedy`	`Offline`
MNIST	99.24 ± 1.93•	99.13 ± 2.23	99.18 ± 2.42	99.13 ± 2.75	**99.27 ± 2.29**
CIFAR-10	90.00 ± 8.66•	87.59 ± 10.42	86.82 ± 9.84	88.44 ± 9.99	**92.57 ± 7.59**
CIFAR-100	71.98 ± 14.96•	67.36 ± 16.88	70.46 ± 14.63	71.04 ± 15.36	**76.25 ± 12.85**

focus on price-centric data or resource trading and overlook performance-aware model trading. Further, most prior works rely on static optimization or complete information assumptions, lacking the capability to handle unknown bidding distributions and long-term budget in an online setting.

Online Learning and Strategic Selection: Combinatorial Multi-Armed Bandit (CMAB) and auction mechanisms have been widely adopted for online strategic selection problems under incentive constraints, with a primary focus on worker recruitment and task assignment in crowdsensing using UCB-based arm selection and critical payment [41], [42]. Multi-agent UCB with posted pricing has been proposed for enabling inter-agent experience trading [58], while the budgeted CMAB has been further reinforced to resist adversarial reward manipulation through critical-value pricing [59]. Recent works have also extended to context-aware CMAB for large-scale unknown worker incentivization [60], differentially private CMAB to protect sensitive information [61], and discounted UCB for handling client quality in federated learning scheduling [62]. However, these works generally assume bids or costs to be deterministic or constant in stationary environments, and thus fail to handle stochastic bids drawn from unknown distributions, rendering them inapplicable to performance-cost-aware model procurement in online edge AI inference.

VII. Discussions

A. Quality Assurance against False or Underperforming Models

AucLearn safeguards against false or underperforming models through two complementary in-framework mechanisms. *First*, the bandit-driven self-correction is intrinsic to our auction design: the post-inference loss $L_{i,n}^t$ feeds into the empirical reward $\bar{r}_{i,n}^t$ at each slot, progressively lowering the UCB score of any low-quality model so that it becomes increasingly unlikely to be selected, and Theorem 2 further guarantees that the cumulative regret from selecting any sub-optimal (including non-adaptively malicious) model grows only logarithmically in τ_B. *Second*, all model specifications, bids, auction outcomes, and payments are immutably recorded on-chain via smart contracts, enabling the system operator to maintain a verifiable history of provider behavior and to impose out-of-protocol penalties (e.g., revocation of provider eligibility) for repeated misconduct. An additional direction explored in the blockchain-marketplace literature is reputation-based screening, where each provider accumulates a reputation score from historical performance and is either filtered out via thresholds or prioritized through top-K selection [63]–[66]. Integrating such a layer as a pre-screening filter prior to UCB-based winner selection is a promising future extension to harden AucLearn against adaptively adversarial providers.

B. IID Assumption and Robustness to Non-Stationarity

Our analysis assumes that data samples are drawn IID from an unknown but time-invariant distribution $\mathscr{D}$, a standard premise in the online learning and combinatorial bandit literature [41]–[43], [52]. In our framework, this assumption underpins the unbiased convergence of $\bar{r}_{i,n}^t$, which supports the concentration arguments in Lemma 1, as well as the equality $\mathbb{E}[L_{i,n}] = \mathbb{E}[\ell_n]$ used in the proof of Theorem 2. However, real-world edge AI inference scenarios may exhibit concept drift, i.e., temporal shifts in the underlying data distribution that can degrade the inference quality of pre-trained models. AucLearn retains effectiveness under mild violations through two intrinsic mechanisms. *First*, each $L_{i,n}^t$ is averaged over M_i^t intra-slot samples, which reduces the sampling variance of $\bar{r}_{i,n}^t$ by the law of large numbers and keeps the Hoeffding bounds underlying Lemma 1 accurate at slot granularity when intra-slot non-stationarity is mild. *Second*, the exploration term $u_{i,n}^t \propto \sqrt{\log t/\beta_{i,n}^t}$ remains strictly positive throughout the horizon, so fresh observations gradually correct the empirical estimates of degraded models and allow AucLearn to revisit previously sub-optimal ones, keeping it responsive to slowly evolving distributions when the drift relative to τ_B is mild. *For more pronounced drift*, formal sub-linear dynamic regret guarantees have been established in the non-stationary bandit literature for both UCB-style and adversarial-bandit-style algorithms, characterizing the admissible drift regime via either a bounded number of abrupt change-points [67] or a variation budget of the form $\sum_t \|\mathscr{D}_t - \mathscr{D}_{t-1}\|_{\mathrm{TV}} = o(\tau_B)$ [68], and requiring explicit non-stationarity-aware machinery rather than standard UCB. In such regimes, AucLearn admits several low-cost upgrades that preserve its overall structure, including non-stationarity-aware UCB variants that replace the cumulative empirical mean with sliding-window or exponentially-discounted variants (e.g., SW-UCB and D-UCB [67]), drift detectors, and a re-derivation of guarantees under dynamic regret. We leave these formal extensions as a promising direction for future work.

VIII. Conclusion

Cost-performance-aware decentralized model trading in edge AI remains critical yet underexplored. To bridge this gap, we propose a blockchain-empowered model auction framework integrating online auctioning, validator selection, and inference under stochastic dynamics. We design online algorithms with provable performance guarantees and economic properties to jointly optimize blockchain operational risk, operational cost, and inference cost, subject to budget and capacity constraints. Extensive evaluations on real-world datasets validate the effectiveness of our approach across multiple dimensions. For future work, we intend to extend our framework to incorporating more security and privacy aspects.

References

[1] S. Deng, H. Zhao, W. Fang, J. Yin, S. Dustdar, and A. Y. Zomaya, "Edge intelligence: The confluence of edge computing and artificial intelligence," *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7457–7469, 2020.

[2] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[3] "Hugging Face," https://huggingface.co/, 2025.

[4] "AI marketplace," https://aimarketplace.co/, 2025.

[5] Amazon Web Services, Inc., "Amazon Web Services (AWS)," https://aws.amazon.com/cn/, 2025.

[6] J. P. Miller, R. Taori, A. Raghunathan, S. Sagawa, P. W. Koh, V. Shankar, P. Liang, Y. Carmon, and L. Schmidt, "Accuracy on the line: on the strong correlation between out-of-distribution and in-distribution generalization," in *ICML*, 2021.

[7] L. Chen, P. Koutris, and A. Kumar, "Towards model-based pricing for machine learning in a data marketplace," in *SIGMOD*, 2019.

[8] J. Liu, J. Lou, J. Liu, L. Xiong, J. Pei, and J. Sun, "Dealer: an end-to-end model marketplace with differential privacy," in *VLDB*, 2021.

[9] S. Chen, Z. Zhou, F. Liu, L. I. Zongpeng, and S. Ren, "Cloudheat: An efficient online market mechanism for datacenter heat harvesting," *ACM Transactions on Modeling and Performance Evaluation of Computing Systems*, vol. 3, no. 3, pp. 11.1–11.31, 2018.

[10] X. Qiu, L. Liu, W. Chen, Z. Hong, and Z. Zheng, "Online deep reinforcement learning for computation offloading in blockchain-empowered mobile edge computing," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 8050–8062, 2019.

[11] Y. Li, Z. Liu, Z. Kou, Y. Wang, G. Zhang, Y. Li, and Y. Sun, "Real-time adaptive partition and resource allocation for multi-user end-cloud inference collaboration in mobile environment," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 13 076–13 094, 2024.

[12] J.-A. Lim, J. Lee, J. Kwak, and Y. Kim, "Cutting-edge inference: Dynamic dnn model partitioning and resource scaling for mobile ai," *IEEE Transactions on Services Computing*, vol. 17, no. 6, pp. 3300–3316, 2024.

[13] K. Zhao, Z. Zhou, X. Chen, R. Zhou, X. Zhang, S. Yu, and D. Wu, "Edgeadaptor: Online configuration adaption, model selection and resource provisioning for edge dnn inference serving at scale," *IEEE Transactions on Mobile Computing*, vol. 22, no. 10, pp. 5870–5886, 2023.

[14] L. Zhou, Z. Xu, Q. Xia, Z. Xu, W. Ren, W. Qi, J. Ma, S. Yan, and Y. Yang, "Chasing common knowledge: Joint large model selection and pulling in mec with parameter sharing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 36, no. 3, pp. 437–454, 2025.

[15] X. Feng, L. Li, T. Wang, W. Xu, J. Zhang, B. Wei, and C. Luo, "Cobc: A blockchain-based collaborative inference system for internet of things," *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 21 389–21 400, 2023.

[16] M. Zhang, J. Cao, Y. Sahni, Q. Chen, S. Jiang, and L. Yang, "Blockchain-based collaborative edge intelligence for trustworthy and real-time video surveillance," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 2, pp. 1623–1633, 2022.

[17] Q. Ma, Y. Qin, C. Zhu, L. Gao, and X. Chen, "Joint resource trading and task scheduling in edge-cloud computing networks," *IEEE Transactions on Networking*, vol. 33, no. 3, pp. 994–1008, 2025.

[18] X. Ren, C. Qiu, Z. Chen, X. Wang, D. Niyato, and W. Wang, "Compcube: A space-time-request resource trading framework for edge-cloud service market," *IEEE Transactions on Services Computing*, vol. 16, no. 5, pp. 3252–3264, 2023.

[19] Y. Zhang, X. Lan, J. Ren, and L. Cai, "Efficient computing resource sharing for mobile edge-cloud computing networks," *IEEE/ACM Transactions on Networking*, vol. 28, no. 3, pp. 1227–1240, 2020.

[20] K. Liu, X. Qiu, W. Chen, X. Chen, and Z. Zheng, "Optimal pricing mechanism for data market in blockchain-enhanced internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9748–9761, 2019.

[21] W. Sun, J. Liu, Y. Yue, and P. Wang, "Joint resource allocation and incentive design for blockchain-based mobile edge computing," *IEEE Transactions on Wireless Communications*, vol. 19, no. 9, pp. 6050–6064, 2020.

[22] K. Xiao, W. Shi, Z. Gao, C. Yao, and X. Qiu, "Daer: A resource preallocation algorithm of edge computing server by using blockchain in intelligent driving," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9291–9302, 2020.

[23] L. D. Nguyen, S. R. Pandey, S. Beatriz, A. Broering, and P. Popovski, "A marketplace for trading ai models based on blockchain and incentives for iot data," *arXiv preprint arXiv:2112.02870*, 2021.

[24] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: online convex optimization with long term constraints," *Journal of Machine Learning Research*, vol. 13, no. 1, pp. 2503–2528, 2012.

[25] T. Chen, Q. Ling, and G. B. Giannakis, "An online convex optimization approach to proactive network resource allocation," *IEEE Transactions on Signal Processing*, vol. 65, no. 24, pp. 6350–6364, 2017.

[26] A. Koppel, F. Y. Jakubiec, and A. Ribeiro, "A saddle point algorithm for networked online convex optimization," *IEEE Transactions on Signal Processing*, vol. 63, no. 19, pp. 5149–5164, 2015.

[27] S. Paternain and A. Ribeiro, "Online learning of feasible strategies in unknown environments," *IEEE Transactions on Automatic Control*, vol. 62, no. 6, pp. 2807–2822, 2016.

[28] "The MNIST Database of Handwritten Digits," http://yann.lecun.com/exdb/mnist/, 1998.

[29] "The CIFAR-10 Dataset," https://www.cs.toronto.edu/∼kriz/cifar.html, 2009.

[30] "XRP Ledger: System Requirements (APEX 2025)," https://xrpl.org/docs/infrastructure/installation/system-requirements, 2025.

[31] C. Xia, J. Zhao, H. Cui, X. Feng, and J. Xue, "Dnntune: Automatic benchmarking dnn models for mobile-cloud computing," *ACM Transactions on Architecture and Code Optimization*, vol. 16, no. 4, pp. 1–26, 2019.

[32] L. L. Zhang, S. Han, J. Wei, N. Zheng, T. Cao, Y. Yang, and Y. Liu, "Nn-meter: Towards accurate latency prediction of deep-learning model inference on diverse edge devices," in *ACM MobiSys*, 2021.

[33] J. Chen, S.-h. Kao, H. He, W. Zhuo, S. Wen, C.-H. Lee, and S.-H. G. Chan, "Run, don't walk: chasing higher flops for faster neural networks," in *IEEE CVPR*, 2023.

[34] A. Khosravi and F. Säämäki, "Beyond bitcoin: evaluating energy consumption and environmental impact across cryptocurrency projects," *Energies*, vol. 16, no. 18, p. 6610, 2023.

[35] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for modern deep learning research," in *AAAI*, 2020.

[36] "Our Open Data – Transport for London," https://tfl.gov.uk/info-for/open-data-users/our-open-data/, 2023.

[37] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[38] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE CVPR*, 2016.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *IEEE CVPR*, 2017.

[41] G. Gao, H. Huang, M. Xiao, J. Wu, Y.-E. Sun, and S. Zhang, "Auction-based combinatorial multi-armed bandit mechanisms with strategic arms," in *IEEE INFOCOM*, 2021.

[42] M. Xiao, J. Wang, H. Zhao, and G. Gao, "Unknown worker recruitment in mobile crowdsensing using cmab and auction," in *IEEE ICDCS*, 2020.

[43] L. Tran-Thanh, A. Chapman, E. M. De Cote, A. Rogers, and N. R. Jennings, "Epsilon–first policies for budget–limited multi-armed bandits," in *AAAI*, 2010.

[44] M. Neely, *Stochastic network optimization with application to communication and queueing systems*. Morgan & Claypool Publishers, 2010.

[45] S. Shalev-Shwartz *et al.*, "Online learning and online convex optimization," *Foundations and Trends® in Machine Learning*, vol. 4, no. 2, pp. 107–194, 2012.

[46] J. Yan, J. Liu, H. Xu, Z. Wang, and C. Qiao, "Peaches: Personalized federated learning with neural architecture search in edge computing," *IEEE Transactions on Mobile Computing*, vol. 23, no. 11, pp. 10 296–10 312, 2024.

[47] L. Chamon and A. Ribeiro, "Probably approximately correct constrained learning," in *NeurIPS*, 2020.

[48] M. E. Ahsen and M. Vidyasagar, "An approach to one-bit compressed sensing based on probably approximately correct learning theory," *Journal of Machine Learning Research*, vol. 20, no. 11, pp. 1–23, 2019.

[49] L. Mauri, S. Cimato, and E. Damiani, "A formal approach for the analysis of the xrp ledger consensus protocol," in *ICISSP*, 2020.

[50] W. Li, C. Feng, L. Zhang, H. Xu, B. Cao, and M. A. Imran, "A scalable multi-layer pbft consensus for blockchain," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 5, pp. 1146–1160, 2020.

[51] L. Zhang, B. Zhang, and C. Li, "An efficient and reliable byzantine fault tolerant blockchain consensus protocol for single-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 23, no. 3, pp. 1974 – 1987, 2023.

[52] D. Zhou and C. Tomlin, "Budget-constrained multi-armed bandits with multiple plays," in *AAAI*, 2018.

[53] R. B. Myerson, "Optimal auction design." *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.

[54] S. Mizuno and F. Jarre, "Global and polynomial-time convergence of an infeasible-interior-point algorithm using inexact computation." *Mathematical Programming*, vol. 84, no. 1, 1999.

[55] R. Gandhi, S. Khuller, S. Parthasarathy, and A. Srinivasan, "Dependent rounding and its applications to approximation algorithms," *Journal of the ACM*, vol. 53, no. 3, pp. 324–360, 2006.

[56] AllBright & Ripple, "Get started with the xrp ledger," 2023.

[57] "CasADi: Symbolic Framework for Automatic Differentiation and Optimization," https://web.casadi.org/, 2022.

[58] G. Gao, H. Huang, J. Wu *et al.*, "You can trade your experience in distributed multi-agent multi-armed bandits," in *IEEE/ACM IWQoS*. IEEE, 2023, pp. 1–10.

[59] H. Wang, E. Wang, Y. Yang, B. Yang, and J. Liu, "A truthful pricing-based defending strategy against adversarial attacks in budgeted combinatorial multi-armed bandits," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 11, pp. 5529–5543, 2023.

[60] F. Li, Y. Chai, H. Yang, P. Hu, and L. Duan, "Incentivizing massive unknown workers for budget-limited crowdsensing: From off-line and on-line perspectives," *IEEE/ACM Transactions on Networking*, vol. 32, no. 3, pp. 2406–2421, 2024.

[61] Y. Xu, M. Xiao, J. Wu, and H. Sun, "Privacy preserving task push in spatial crowdsourcing with unknown popularity," *IEEE Transactions on Parallel and Distributed Systems*, vol. 35, no. 11, pp. 2039–2053, 2024.

[62] H. Zhang, N. Ti, D. Wang, X. Du, Q. Wang, and W. Xia, "Quality-aware incentive mechanism for efficient federated learning in mobile crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 12, pp. 19 696–19 707, 2024.

[63] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 700–10 714, 2019.

[64] H. Kasyap, A. Manna, and S. Tripathy, "An efficient blockchain assisted reputation aware decentralized federated learning framework," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 2771–2782, 2022.

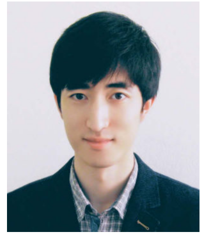
[65] W. Zhu, L. Shi, J. Li, B. Cao, K. Wei, Z. Wang, and T. Huang, "Trustworthy blockchain-assisted federated learning: Decentralized reputation management and performance optimization," *IEEE Internet of Things Journal*, vol. 12, no. 3, pp. 2890–2905, 2024.

[66] S. Gao, X. Chen, J. Zhu, X. Dong, and J. Ma, "Trustworker: A trustworthy and privacy-preserving worker selection scheme for blockchain-based crowdsensing," *IEEE Transactions on Services Computing*, vol. 15, no. 6, pp. 3577–3590, 2021.

[67] A. Garivier and E. Moulines, "On upper-confidence bound policies for switching bandit problems," in *ALT*, 2011.

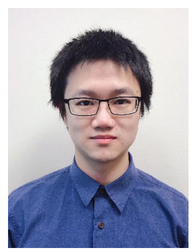
[68] O. Besbes, Y. Gur, and A. Zeevi, "Stochastic multi-armed-bandit problem with non-stationary rewards," in *NeurIPS*, 2014.

Yining Zhang received the Ph.D. degree in artificial intelligence from the Beijing University of Posts and Telecommunications. Her research interests include online network optimization and cloud-edge collaborative computing.



Lei Jiao received the Ph.D. degree in computer science from the University of Göttingen, Germany. He is currently a faculty member at the University of Oregon, USA and was previously a technical staff member at Nokia Bell Labs in Ireland. His research interests include machine learning systems, edge computing, cloud computing, network optimization, and network economics.

Konglin Zhu received the master's degree in computer science from the University of California, Los Angeles, CA, USA, and the Ph.D. degree from the University of Göttingen, Germany, in 2009 and 2014, respectively. He is now a Professor with the Beijing University of Posts and Telecommunications, Beijing, China. His research interests include Internet of Vehicles, Edge Computing and Distributed Learning.



Yebo Feng is a research fellow in the College of Computing and Data Science (CCDS) at Nanyang Technological University (NTU). He received his Ph.D. degree in Computer Science from the University of Oregon (UO) in 2023. His research interests include network security, blockchain security, and anomaly detection. He is the recipient of the Best Paper Award of 2019 IEEE CNS, Gurdeep Pall Graduate Student Fellowship of UO, and Ripple Research Fellowship. He has served as the reviewer of IEEE TDSC, IEEE TIFS, ACM TKDD, IEEE JSAC, IEEE COMST, etc. Furthermore, he has been a member of the program committees for international conferences including SDM, CIKM, and CYBER, and has also served on the Artifact Evaluation (AE) committees for USENIX OSDI and USENIX ATC.

Lin Zhang received the Ph.D. degrees in 2001, from the Beijing University of Posts and Telecommunications, Beijing, China. He joined Beijing University of Posts and Telecommunications in 2004 and has been a Professor since 2011. He is also the director of Beijing Big Data Center. His current research interests include mobile cloud computing.

Jiahua Xu is an Associate Professor in Financial Computing and Programme Director of the MSc Emerging Digital Technologies at UCL. Her research focuses on blockchain economics and decentralized finance, with publications in Usenix Security, ACM IMC, ACM ASIACCS, FC, IEEE ICDCS and IEEE COMST.