

Neural Auction Mechanisms for Partial Fulfillment in Edge Computing

YUANYUAN ZHANG¹, XINPENG LU², MINGXUAN LIANG¹, JUNWU ZHU¹, YONGLONG ZHANG¹,
and YI JIANG¹

¹School of Information Engineering, Yangzhou University, Yangzhou, 225127, China

²Advanced Technology Research Institute, University of Science and Technology of China, Hefei, 230026, China

Corresponding author: Junwu Zhu (e-mail: jwzhu@yzu.edu.cn).

This work was supported in part by the National Natural Science Foundation of China (No.61872313), in part by the Yangzhou Science and Technology (No.YZU202103), and in part by the Postgraduate Research & Practice Innovation Program of Jiangsu Province (No.KYCX24_3741).

ABSTRACT In the edge computing market, application service providers typically purchase computing resources from edge service providers to serve end users efficiently. In order to motivate edge service providers to actively deploy computing services, researchers utilize auctions to design an effective incentive scheme. However, traditional auctions assume all-or-nothing demand fulfillment and ignore scenarios where partial fulfillment is feasible. To address this challenge, two models are proposed: PDNet, built on fully connected neural networks, and EANet, which combines exchangeable matrix layers with attention mechanisms to learn optimal auction. Unlike prevailing methods, our approaches support flexible allocation, enabling partial fulfillment rather than all-or-nothing outcomes. The auction design is formulated as a constrained learning problem, yielding scalable and incentive-compatible mechanisms through end-to-end training. Extensive experimental results demonstrate that both PDNet and EANet outperform traditional baselines. EANet also exhibits strong generalization capabilities in scenarios with invisible input sizes. This work provides a modular and extensible deep learning framework for designing practical high-performance auction mechanisms in resource-constrained markets.

INDEX TERMS Edge computing, Optimal Auction, Deep Learning.

I. INTRODUCTION

With the rapid advancement of information technology, explosive growth in mobile devices has laid a solid foundation for the widespread adoption of applications such as virtual reality, the Internet of Things, and autonomous driving [1]–[3]. These applications require greater computational resources and stricter latency due to their complex big data processing needs [4]. Traditional solutions typically rely on cloud-centric architectures where data need to be transferred from mobile devices to remote cloud servers for processing and computation, and then the results are returned to the devices [5]. However, long data transfers can lead to significant latency for the timely responses required by applications.

Edge computing, an emerging distributed paradigm, offers low latency and high bandwidth utilization and has attracted broad interest across applications [6]–[8]. In this paradigm, edge service providers (e.g., AT&T and Comcast) deploy a large number of servers near users and rent out their computational, communication, and storage resources. Application service providers (e.g., Netflix and Activision Blizzard)

provide resource-intensive services to users by renting the resources of these edge servers. In the edge computing resource trading market, application service providers act as bidders while edge service providers act as sellers. This market structure not only promotes efficient resource utilization but also reduces latency, thereby improving the user experience [9].

Due to the high costs of deploying and maintaining edge computing nodes, it is crucial to design an effective incentive mechanism to motivate edge service providers to actively deploy them. Auction, as an effective incentive mechanism design scheme, has been widely used in edge computing [10]–[12]. The framework for resource auction in edge computing is illustrated in **Figure 1**. Edge service providers sell services including computation and storage to application service providers. Application service providers provide end users (e.g., smartphones, vehicle computer systems, and smart home devices) with efficient data computation and processing services. Since end-users have varying application requirements, application service providers strategically purchase services from edge service providers. Both utilize an auction

mechanism to determine winner, allocation and payment results.

In early research, many scholars developed mechanisms based on traditional auction theory for the resource allocation problem of edge computing [13]–[16]. Habiba et al. [13] proposed a repeated auction model for load-aware dynamic resource allocation in multi-access edge computing. This model utilizes a modified Generalized Second Price (GSP) auction mechanism to dynamically allocate resources while ensuring individual rationality and symmetric Nash equilibrium. Su et al. [14] proposed a truthful combinatorial auction mechanism for mobile edge computing in the Industrial Internet of Things (IIoT), which solves the resource trading problem of personalized computing offloading services through a three-phase process. However, TCA is mainly designed for single-round auctions and does not fully consider changes in resource demand in dynamic environments. In addition, Sun et al. [17] proposed a resource allocation mechanism based on bilateral auctions, which balances fairness and efficiency of resource allocation by allowing bidders and sellers to engage in iterative bidding, but the method is less scalable in multi-user scenarios. These traditional auction mechanisms are usually designed based on fixed rules or mathematical models, which cannot adapt well to highly dynamic environments and complex demand patterns.

In auction mechanism design, there are two main objectives. One is maximizing social welfare, which is the total valuations of the bidders who win. The other, known as the optimal auction, is maximizing sellers' revenue, which is the total payments collected from the winners. To better reflect practical settings, we focus on maximizing the revenue of edge service providers as our design objective. Although designing optimal auctions is challenging, Myerson solved the optimal auction for a single item in 1981 [18]. However, the multi-item optimal auction problem remains has not been fully solved. In recent years, deep learning has gained great attention due to its powerful feature extraction and non-linear modeling capabilities. Deep learning based auction methods usually frame the auction design as a continuous optimization problem, and seek the optimal solution through a machine learning pipeline. Dütting et al. [19] proposed the first neural network framework, RegretNet, which finds near-optimal solutions by modeling the auction as a multi-layer neural network. Subsequent works [20]–[22] advanced this line of research along orthogonal directions, leading to fresh advances in optimal mechanism design. Kuo et al. [20] extended RegretNet with fairness constraints to effectively address proportional fairness in allocation while maintaining competitive revenue. Peri et al. [21] encoded human preferences over socially desirable allocations and proposed PCA to quantify constraint adherence. Inspired by privacy risks in ad auctions, Stein et al. [22] extended RegretNet with information bottlenecks to limit bidder leakage, preserving incentive compatibility and strong revenue. In addition, Rahme et al. [23] design permutation-equivariant architectures that recover optimal symmetric mechanisms, improving sample

efficiency. Duan et al. [24] introduced a Transformer-driven auction framework that models complex bidder-item interactions to approach optimal mechanisms.

Although substantial progress has been made, RegretNet-based extensions [20]–[22] primarily broaden objectives (e.g., fairness, human preferences, and privacy) without altering the underlying allocation paradigm. In parallel, architectural advances [23], [24] emphasize permutation-equivariant and context-integrated modeling to improve sample efficiency and cross-scale generalization. These methods typically require bidders to commit fixed quantities *ex ante*, leading to an all-or-nothing outcome where demands are either fully met or entirely unmet. In practical edge computing markets, resources are divisible and application service providers often accept partial fulfillment in exchange for lower cost or faster response. This all-or-nothing constraint limits overall utilization and makes it difficult for application service providers to obtain the needed resources.

This paper leverages recent advances in deep learning for mechanism design to build revenue-maximizing neural auctions with partial fulfillment for edge computing. Our approach utilizes neural networks to overcome the limitations of traditional all-or-nothing resource allocation, providing more adaptive solutions. The main contributions of this paper are summarized as follows:

- We introduce a data-driven approach for learning optimal auction mechanisms that maximize revenue while allowing partial fulfillment for application service providers. Unlike classic methods, our approach accommodates flexible and individualized resource allocation rather than being restricted to binary allocation outcomes. The mechanisms we learn are empirically shown to be approximately incentive compatible.
- We design two representative neural architectures: PDNet, a fully connected model, and EANet, an attention-based model, thereby extending the design space beyond standard deep networks. In particular, EANet achieves permutation equivariance and enables strong generalization to varying participant and resource configurations.
- Extensive experiments on synthetic edge auction datasets show that our neural mechanisms consistently and significantly outperform conventional auction baselines, with clear revenue improvements and strong adaptability to dynamic market scenarios. Our results provide empirical guidance and new benchmarks for practical auction design with partial fulfillment.

The subsequent sections of the paper are organized as follows: we introduce the problem setting of an optimal auction in Section 2. In Section 3, we describe the proposed method in detail. Then, we validate the effectiveness of the proposed method through a series of experiments in Section 4. Finally, Section 5 gives a summary and future work.

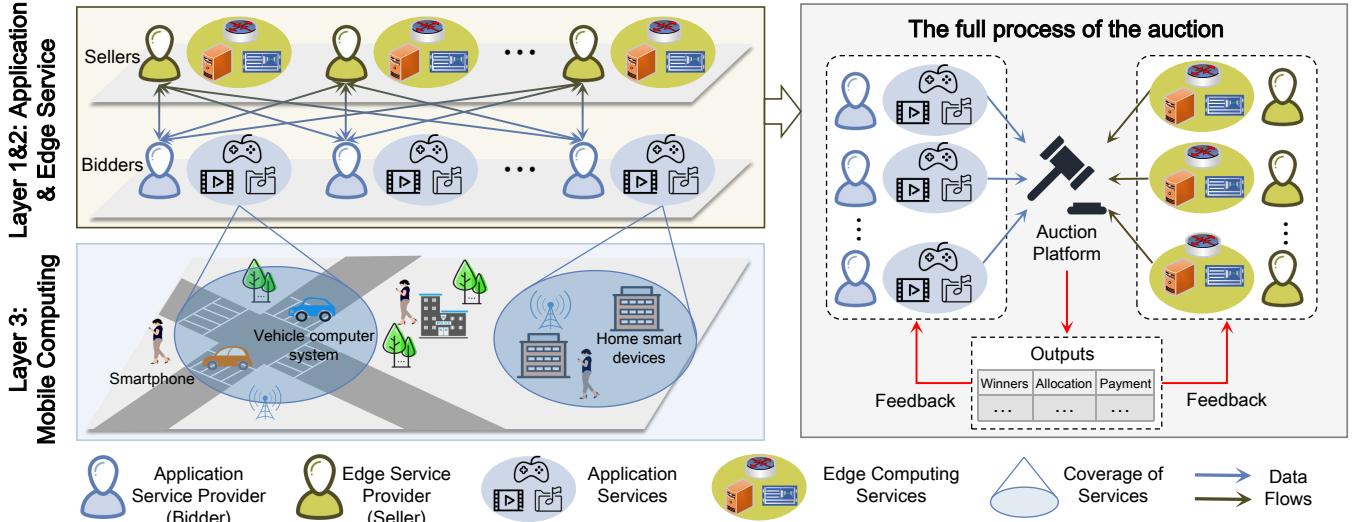


FIGURE 1: The resource auction framework in edge computing, where application service providers act as bidders and edge service providers as sellers.

II. PROBLEM SETTING

We first describe the auction model. Then, we define the desirable properties of auctions and formulate the optimal auction as an optimization problem.

A. AUCTION MODEL WITH PARTIAL FULFILLMENT

We consider an auction with *partial fulfillment*, meaning that each bidder's demand quantity may be satisfied only in part (i.e., not fully met). The auction involves n application service providers, denoted $N = \{1, \dots, n\}$, and m resources, denoted $M = \{1, \dots, m\}$. Each application service provider requires a distinct quantity of resources, and the corresponding edge service provider offers a varying amount of resources. Considering that the quantity of resources can be quantified, we denote the demand matrix as $\mathbf{o} = (o_{ij})_{i \in N, j \in M}$, where o_{ij} is the quantity of the resource j demanded by the application service provider i . The supply vector is $\mathbf{q} = (q_j)_{j \in M}$, where q_j is the quantity of the resource j supplied by the edge service providers. The demand of each application service provider can be partially fulfilled, i.e., the quantity d_{ij} of the resource j that is actually allocated to the application service provider i does not exceed o_{ij} .

Let $\mathbf{b} = (b_{ij})_{i \in N, j \in M}$ denote the bidding profile, where b_{ij} represents the bid submitted by the application service provider i for the resource j . Each application service provider i possesses a valuation v_{ij} for the resource j , and $v_i = \sum_{j \in S} v_{ij}$ for a subset of resources $S \subseteq M$. The valuation profile $\mathbf{v} = (v_{ij})_{i \in N, j \in M}$ is sampled from the distribution D . Such valuations, known as *additive*, have been extensively studied in the field of multi-unit auction design [25]–[27]. The additive valuations reduce the input space of v_{ij} from a size of 2^M to M , i.e., the application service providers only need to bid for each resource. Since the valuation information is private, the bid b_{ij} reported by the application service provider i may not be their true valuation v_{ij} .

We define $\mathcal{V} := \times_i \mathcal{V}_i$ as the set of domains of the joint valuation profile, where \mathcal{V}_i represents the set of domains of all possible valuation profiles $v_i = (v_{i1}, v_{i2}, \dots, v_{im})$ for the application service provider i . Let $\mathcal{V}_{-i} := \times_{j \neq i} \mathcal{V}_j$ denote the set of domains of joint valuation profiles except v_i . Similarly, we define v_{-i} as a valuation profile except i . Throughout this work, we assume that $b_i \in \mathcal{V}_i$ for each $i \in N$, without loss of generality.

Definition 1. *The auction mechanism is defined as a triple $\mathcal{M}(g, d, p)$, where $g : \mathbb{R}^{n \times m} \rightarrow [0, 1]^{n \times m}$ is the allocation rule, $d : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}_{\geq 0}^{n \times m}$ is the demand rule, $p : \mathbb{R}^{n \times m} \rightarrow \mathbb{R}_{\geq 0}^n$ is the payment rule.*

Given the bidding profiles \mathbf{b} , the allocation rule g_{ij} calculates the probability of allocating the resource j to the application service provider i , the demand rule d_{ij} outputs the number of the resource j that are actually allocated to the application service provider i , and the payment rule p_i determines the price that the application service provider i must pay.

B. DESIRABLE PROPERTIES

The following definitions are presented to elucidate the desirable properties of an optimal auction:

Definition 2. *The utility of the application service provider i is defined as follows:*

$$u_i(v_i, b) = \sum_{j=1}^m g_{ij}(b)v_{ij}d_{ij} - p_i(b), \forall v_i \in \mathcal{V}_i, b \in \mathcal{V} \quad (1)$$

Definition 3. *The $\mathcal{M}(g, d, p)$ is Dominant Strategy Incentive Compatibility (DSIC) if each application service provider achieves maximum utility by bidding truthfully, regardless of how others bid, i.e.,*

$$u_i(v_i, (v_i, b_{-i})) \geq u_i(v_i, (b_i, b_{-i})), \forall b_i \in \mathcal{V}_i \quad (2)$$

DSIC means that, regardless of how others bid, no deviation from truthful bidding can increase utility, eliminating the incentive to game the auction. Since the characterization of DSIC in the multi-unit auction mechanisms remains unexplored, we adopt a weaker notion of *ex-post regret*.

Definition 4. *The ex-post regret is the maximum utility gain that the application service provider i can attain by considering all possible non-truthful bids, while keeping the bids of other participants unchanged.*

$$rgt_i(v) = \mathbb{E}_{v \sim D} \left[\max_{b_i \in \mathcal{V}_i} u_i(v_i, (b_i, v_{-i})) - u_i(v_i, (v_i, v_{-i})) \right] \quad (3)$$

Following the perspective of Dütting et al. [19], DSIC can be characterized by the vanishing regret condition $rgt_i(v) = 0, \forall i \in N, v \in \mathcal{V}$.

Definition 5. *The $\mathcal{M}(g, d, p)$ is Individual Rationality (IR) if any application service provider $i \in N$ receives non-negative utility. Formally, we have*

$$u_i(v_i, (v_i, b_{-i})) \geq 0, \forall b_{-i} \in \mathcal{V}_{-i} \quad (4)$$

IR dictates that a bidder will never be charged more than its valuation of the resource she receives. In addition, we want $\mathcal{M}(g, d, p)$ to be Supply-Demand Equilibrium, defined as follows:

Definition 6. *The $\mathcal{M}(g, d, p)$ is Supply-Demand Equilibrium (SE) if the total demand for the resource j from all application service providers does not exceed the supply of the resource j provided by the seller. Formally, we have*

$$\sum_{i=1}^n g_{ij} d_{ij} \leq q_j, \forall i \in N, \forall j \in M \quad (5)$$

where g_{ij} represents the probability of allocating the resource j to the application service provider i .

In edge computing markets, the fundamental goal of the auction mechanism is to maximize the revenue of edge computing service providers while satisfying the necessary constraints. Given a valuation profile v , we approach the revenue maximization challenge as an optimization problem over allocation and payment functions. This task is formally written as:

$$\max_{g, p} \sum_{i \in N} p_i(v) \quad (6)$$

subject to the auction constraints of DSIC, IR, and SE, for all $i \in N$.

C. FORMALIZING AUCTION DESIGN AS A LEARNING TASK

We recast the problem of auction mechanism design as a learning task. Instead of optimizing a loss function defined by prediction error, we use the negative expected revenue—computed over valuations sampled from distribution D —as our objective.

In this framework, the aim is to learn an auction mechanism that minimizes the negative expected revenue, i.e., $-\sum_{i \in N} p_i(v)$. We re-formulate the objective as minimizing expected loss, i.e., the negative expected revenue:

$$\min -\mathbb{E}_{v \sim D} \left[\sum_{i=1}^n p_i(v) \right] \quad (7)$$

Let $S = \{v^{(\ell)}\}_{\ell=1}^L$ be L valuation profiles drawn from the distribution D . The empirical ex-post regret for bidder i is defined by:

$$\widehat{rgt}_i(v) = \frac{1}{L} \sum_{\ell=1}^L \left(\max_{v'_i \in \mathcal{V}_i} u_i(v_i^{(\ell)}; (v'_i, v_{-i}^{(\ell)})) - u_i(v_i^{(\ell)}; v^{(\ell)}) \right) \quad (8)$$

The objective of our auction design is to find an optimal auction that maximizes expected revenue while satisfying the constraints of DSIC, IR, and SE. Therefore, the problem of finding the optimal auction is reformulated as follows:

$$\begin{aligned} & \min -\frac{1}{L} \sum_{\ell=1}^L \sum_{i=1}^n p_i(v^{(\ell)}) \\ \text{s.t. } & \begin{cases} \text{DSIC : } \widehat{rgt}_i(v) = 0 & \forall i \in N, \forall v \in \mathcal{V} \\ \text{IR : } u_i(v_i, (v_i, b_{-i})) \geq 0 & \forall i \in N, \forall v \in \mathcal{V}, b_{-i} \in \mathcal{V}_{-i} \\ \text{SE : } \sum_{i=1}^n s_{ij} d_{ij} \leq q_j & \forall i \in N, \forall j \in M \end{cases} \end{aligned} \quad (9)$$

Theorem 1. *Assume D has full support on $\mathcal{V} := \times_i \mathcal{V}_i$. For a learned mechanism $\mathcal{M}(g, d, p)$, each bidder's expected ex-post regret satisfies*

$$rgt_i(v) = 0 \quad \text{for all } i \in N.$$

Then $\mathcal{M}(g, d, p)$ is DSIC.

The proof details can be found in Appendix A.

III. METHOD

Once the auction design problem is cast as a learning task, we develop models that enable end-to-end optimization using gradient-based methods.

A. DESIGN RATIONALE

We provide two complementary instantiations of $\mathcal{M}(g, d, p)$ to match different deployment regimes and generalization needs:

- **PDNet.** PDNet uses multilayer perceptrons to implement the allocation, quantity, and payment mappings. This design offers a strong function approximation for

fixed auction sizes and serves as a transparent baseline. It is efficient to train, and its simplicity clarifies the contribution of partial fulfillment modeling.

- **EANet.** EANet integrates exchangeable layers with row/column self-attention to capture bidder–resource interactions while preserving the permutation equivariance. Exchangeable layers share parameters across bidders and resources so that the outcomes are invariant to index permutations. The attention blocks contextualize each bidder–resource decision with signals from competing bids, which enhances expressiveness and supports transfer to unseen numbers of bidders and resources.

Given the optimization problem in Section 2, we perform end-to-end training with stochastic gradients. Both PDNet and EANet share the same loss—revenue maximization regularized by regret (IC) and capacity (SE) penalties—while IR is guaranteed by the payment construction. This modular design: (i) aligns directly with the constraints of the problem, including partial requests and finite capacities; (ii) separates concerns across allocation, quantity, and payment while allowing end-to-end learning; (iii) scales from fixed-size markets (PDNet) to variable-size, interaction-rich settings (EANet).

B. PDNET ARCHITECTURE

The proposed PDNet consists of an allocation network, a demand network, and a payment network. In the allocation network, the input is the bidding profile b and the output is the corresponding allocation probability $g_{ij} \in [0, 1]$. The demand network takes the bidding profile b as input and outputs the quantity $d_{ij} \in [0, o_{ij}]$ of the resource j that is actually allocated to the application service provider i . In the payment network, the bidding profile b is also used as input and its output is the expected payment $p_i \in \mathbb{R}^n$. The architecture of PDNet is depicted in **Figure 2**.

In the allocation network g , we first calculate the hidden allocation feature $\mathbf{h}^{(l)}$ through a fully connected neural network:

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{c}^{(l)}) \quad (10)$$

where $\mathbf{W}^{(l)}$ is the weight matrix of the l -th layer, $\mathbf{c}^{(l)}$ is the bias vector of the l -th layer, $\mathbf{h}^{(0)} = \mathbf{b}$ denotes the initial input and σ is the Tanh activation function.

Then, we take the output from the last layer of the fully connected neural network as $\mathbf{s} = \{s_{11}, s_{12}, \dots, s_{(n+1)m}\}$. To allow for the possibility that resources are unallocated, the output of the last layer has a size of $(n+1) \times m$ and is intended to mimic an additional virtual participant with zero valuation.

To ensure feasibility, we employ a sigmoid function in the output layer to calculate the probability g_{ij} that resource j is allocated to the application service provider i .

$$g_{ij} = \text{Sigmoid}(s_{ij}) \quad (11)$$

In the demand network d , we apply a fully connected neural network to calculate the hidden demand feature $\mathbf{h}^{(l)}$:

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{c}^{(l)}) \quad (12)$$

where $\mathbf{h}^{(0)} = \mathbf{b}$ denotes the initial input. We take the output from the last layer as $\tilde{\mathbf{d}} = \{\tilde{d}_{11}, \tilde{d}_{12}, \dots, \tilde{d}_{nm}\}$.

Additionally, we expect the mechanism to provide partial fulfillment for the application service providers. To achieve this, we calculate the quantity d_{ij} of the resource j via multiplication:

$$d_{ij} = o_{ij} \cdot \text{Sigmoid}(\tilde{d}_{ij}) \quad (13)$$

where o_{ij} represents the quantity demanded by the application service provider i for the resource j . By doing so, the quantity of resource j actually allocated to the application service provider i does not exceed the quantity of their demand.

We further denote the element-wise product of the allocation probability and the fulfilled quantity by:

$$t_{ij} = g_{ij} \cdot d_{ij} \quad (14)$$

where g_{ij} and d_{ij} are the outputs of the allocation network and the demand network, respectively.

In the payment network p , we first calculate the hidden feature $\mathbf{h}^{(l)}$ using a fully connected layer with the Tanh activation function:

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}) \quad (15)$$

where $\mathbf{h}^{(0)} = \mathbf{b}$ represents the initial input of the fully connected layer.

Then, we calculate the payment probability $\tilde{p}_i \in [0, 1]$ for the application service provider i using the sigmoid function, which ensures that the payment is non-negative, i.e., the auction mechanism is IR.

$$\tilde{p}_i = \text{Sigmoid}(\mathbf{h}^{(l)}) \quad (16)$$

Finally, we perform a summation operation to obtain the total expected payment p_i for the application service provider i .

$$p_i = \tilde{p}_i \sum_{j=1}^m t_{ij} b_{ij} \quad (17)$$

C. EANET ARCHITECTURE

Compared to traditional fully connected layers with limited expressive power, we propose PDNet by combining exchangeable matrix layers [28] and self-attention mechanisms [29] to enhance model expressiveness and ensure permutation equivariance in auction design (see **Figure 3**). For details on self-attention, please refer to Appendix B.

We first perform an unsqueeze operation on the bidding profile $b \in \mathbb{R}^{n \times m}$, transforming it into $X \in \mathbb{R}^{n \times m \times 1}$. Next, a fully connected layer is applied to project the input along the channel dimension, resulting in the feature $H \in \mathbb{R}^{n \times m \times d}$:

$$H = f^{\text{bid}}(X) \quad (18)$$

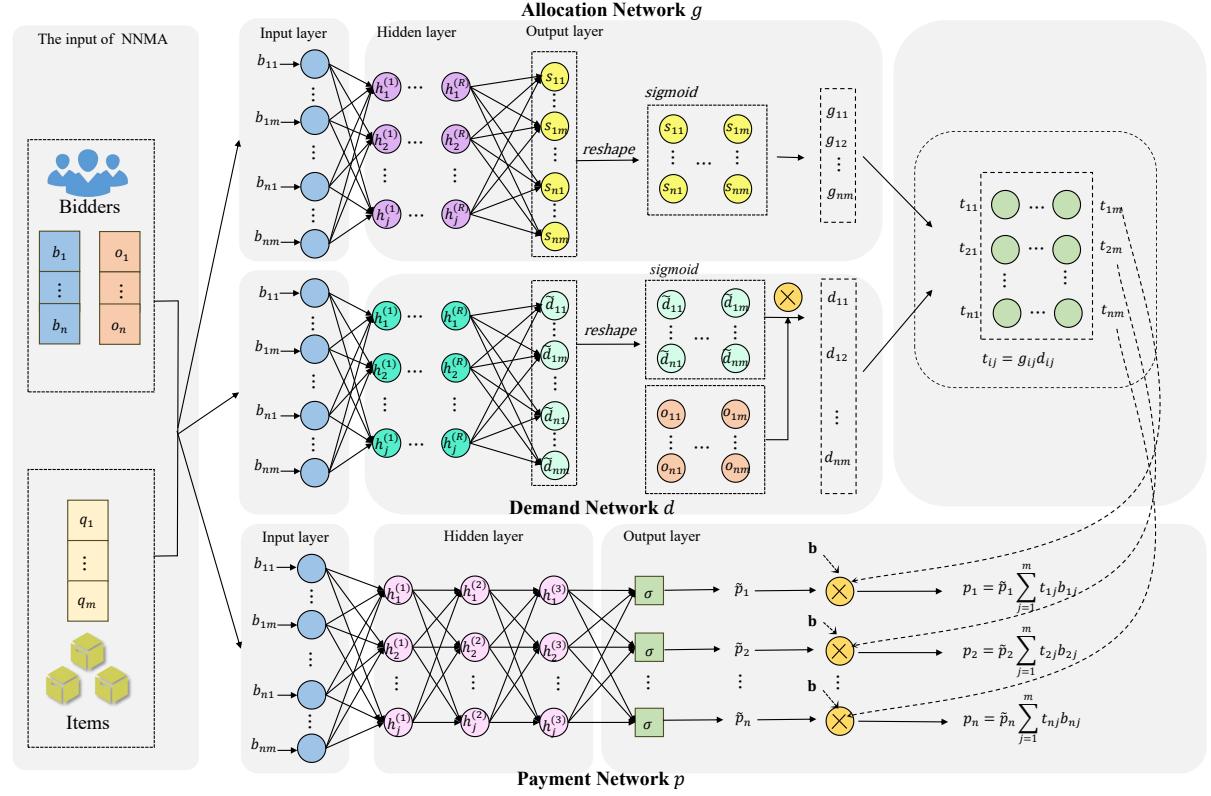


FIGURE 2: The architecture of the PDNet, which takes the bidding profile $b \in \mathbb{R}^{n \times m}$. The allocation network outputs the allocation probability g_{ij} , while the demand network predicts the actually allocated resource quantity d_{ij} for each application service provider. The two outputs are multiplied element-wise and fed into the payment network, which finally produces the payment.

Subsequently, the exchangeable matrix layer converts H into the feature vector E , which incorporates information from other bids:

$$E_{i,j}^{(k)} = \sigma \left(\sum_{k=1}^K \left(\theta_1^{(k)} H_{i,j}^{(k)} + \frac{\theta_2^{(k)}}{n} \left(\sum_{i'} H_{i',j}^{(k)} \right) + \frac{\theta_3^{(k)}}{m} \left(\sum_{j'} H_{i,j'}^{(k)} \right) + \frac{\theta_4^{(k)}}{nm} \left(\sum_{i',j'} H_{i',j'}^{(k)} \right) + \theta_5^{(k)} \right) \right) \quad (19)$$

where n is the number of bidders, m is the number of resources, $\theta_1^{(k)} \dots \theta_4^{(k)}$ are learnable weights for the k -th channel, and $\theta_5^{(k)}$ is a bias term.

In order to effectively capture the complex interactions between bidder–resource pairs, we independently apply self-attention layers to each i -th row of E (denoted as $E_{i,\cdot} \in \mathbb{R}^{m \times d_h}$) and each j -th column of E (denoted as $E_{\cdot,j} \in \mathbb{R}^{n \times d_h}$).

$$\begin{aligned} E_{i,\cdot}^{\text{row}} &= \text{SelfAttention}(E_{i,\cdot}) \\ E_{\cdot,j}^{\text{column}} &= \text{SelfAttention}(E_{\cdot,j}) \end{aligned} \quad (20)$$

For each bid, we concatenate the two self-attention outputs to form the joint bidder–resource representation $E'_{ij} \in \mathbb{R}^{2d_h}$:

$$E'_{ij} = [E_{ij}^{\text{row}} \| E_{ij}^{\text{column}}] \quad (21)$$

where $\|$ denotes the concatenation operation. Similarly, we apply another attention mechanism to E to obtain the joint demand representation, denoted as $Z'_{ij} = [Z_{ij}^{\text{row}} \| Z_{ij}^{\text{column}}]$. It is worth mentioning that each layer shares its parameters across all bids, which enables the model to remain equivariant to the ordering of bids and to adapt to varying input scales.

Subsequently, fully connected layers map $E' \in \mathbb{R}^{n \times m \times 2d_h}$ and $Z' \in \mathbb{R}^{n \times m \times 2d_h}$ to the original dimension 1, yielding the two sets of matrix $s = (s_{ij})_{i \in N, j \in M}$ and $\tilde{d} = (\tilde{d}_{ij})_{i \in N, j \in M}$.

$$\begin{aligned} s &= f^{\text{resource}}(E') \\ \tilde{d} &= f^{\text{agent}}(Z') \end{aligned} \quad (22)$$

Next, we apply the sigmoid function to compute the final allocation probability g_{ij} , thereby ensuring the feasibility of the allocation:

$$g_{ij} = \text{Sigmoid}(s_{ij}) \quad (23)$$

To allow for partial fulfillment, we calculate the actual quantity of the resource j allocated to the service provider i using an element-wise product:

$$d_{ij} = o_{ij} \cdot \text{Sigmoid}(\tilde{d}_{ij}), \quad (24)$$

where o_{ij} represents the demand of the service provider i for resource j .

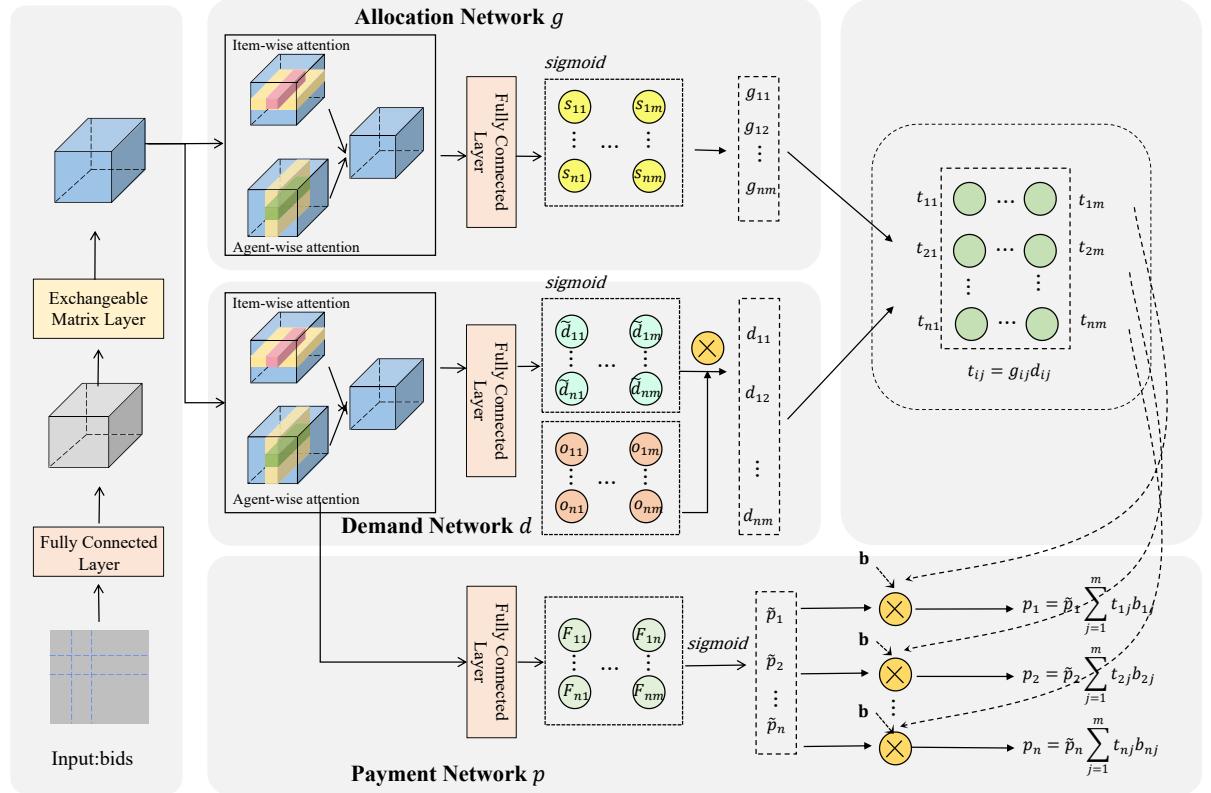


FIGURE 3: The architecture of the EANet, which takes an $n \times m$ bid matrix as input. It employs an attention mechanism to learn permutation-equivariant representations for bidder–resource pairs, and then produces a feasible allocation matrix g together with a demand matrix d . The two are combined via element-wise multiplication and fed into the payment module, which outputs the final payments.

Next, we define the product of the allocation probability and the fulfilled quantity as:

$$t_{ij} = g_{ij} \cdot d_{ij} \quad (25)$$

For payment calculation, the output Z^{row} from the attention mechanism is mapped to dimension 1 through a fully connected layer, resulting in a matrix $F \in \mathbb{R}^{n \times m}$:

$$F = f^{\text{pay}}(Z^{\text{row}}). \quad (26)$$

Next, we use a sigmoid unit to compute the probability of payment $\hat{p}_i \in [0, 1]$ for each bidder i :

$$\tilde{p}_i = \text{Sigmoid} \left(\frac{1}{m} \sum_{j=1}^m F_{ij} \right) \quad (27)$$

Finally, the total expected payment for each bidder i is calculated as:

$$p_i = \hat{p}_i \sum_{j=1}^m t_{ij} b_{ij} \quad (28)$$

Theorem 2. Suppose the mechanism $M(g, d, p)$ satisfies:

- (i) $\tilde{p}_i(b) \in [0, 1]$ for all i and b (sigmoidal fractional payment),
- (ii) $g_{ij}(b) \in [0, 1]$ and $d_{ij}(b) \in [0, o_{ij}]$ for all i, j, b .

Then the mechanism is IR, i.e.,

$$u_i(v_i, (v_i, b_{-i})) \geq 0 \quad \text{for all } i \in N \text{ and } b_{-i} \in V_{-i}.$$

The proof details can be found in Appendix A.

D. MODEL TRAINING

Similar to Duan et al. [24], we formulate the problem of finding an optimal auction as a learning task. The auction mechanism is parameterized as $\mathcal{M}(g^w, d^w, p^w)$, where w represents the parameters to be optimized. Therefore, the expected revenue is given by the following:

$$\min_{w \in \mathbb{R}^d} -\frac{1}{L} \sum_{\ell=1}^L \sum_{i=1}^n p_i^w(v^{(\ell)}) \quad (29)$$

During the training process, we need to ensure that the auction satisfies the DSIC and SE constraints. Therefore, we introduce two indicators to measure the extent to which $\mathcal{M}(g^w, d^w, p^w)$ complies with these constraints.

We define $ic_i(w)$ to measure the extent to which the auction complies with DSIC:

$$ic_i(w) = \widehat{rgt}_i^w(v) \quad (30)$$

where $ic_i = 0$ means that $rgt_i^w(v) = 0$, ensuring the DSIC property.

Then, we define se_j for the resource j :

$$se_j(w) = \frac{1}{L} \sum_{l=1}^L \left[\alpha \left| \sum_{i=1}^n d_{ij} - q_j \right| + (1-\alpha) \max \left\{ 0, \sum_{i=1}^n d_{ij} - q_j \right\} \right] \quad (31)$$

where α is a trade-off coefficients that controls the weights of the constraint terms. The first constraint term aims to allocate resources to bidders as much as possible, and the second term aims to ensure that the mechanism is SE.

Given the challenges of optimization problems with constraints, we utilize the augmented Lagrangian approach to convert the constrained optimization problem into an unconstrained one.

$$\begin{aligned} \mathcal{L}(w; \lambda) &= -\frac{1}{L} \sum_{l=1}^L \sum_{i \in N} p_i^w(v^{(l)}) + \mathcal{L}_{ic} + \mathcal{L}_{se} \\ \mathcal{L}_{ic} &= \sum_{i \in N} \lambda_{(ic,i)} ic_i(w) + \frac{\rho}{2} ic_i^2(w) \\ \mathcal{L}_{se} &= \sum_{j \in M} \lambda_{(se,j)} se_j(w) + \frac{\rho}{2} se_j^2(w) \end{aligned} \quad (32)$$

where $\lambda_{(ic,i)}$ and $\lambda_{(se,j)}$ represent Lagrange multipliers with DSIC and SE constraints, respectively. The ρ is a quadratic penalty term that controls the weight. The parameters w are optimized by solver, while the Lagrange multipliers are updated iteratively. The iteration process is as follows:

$$\begin{aligned} w^{t+1} &\in \operatorname{argmin}_w \mathcal{L}(w^t; \lambda_{ic}^t, \lambda_{se}^t) \\ \lambda_{(ic,i)}^{t+1} &= \lambda_{(ic,i)}^t + ic_i(g^{t+1}, p^{t+1}) \rho, \forall i \in N \\ \lambda_{(se,j)}^{t+1} &= \lambda_{(se,j)}^t + se_j(g^{t+1}, p^{t+1}) \rho, \forall j \in M \end{aligned} \quad (33)$$

During the training process, the setting of parameters ρ affects the convergence speed. The optimal solution can be quickly converged if the parameters ρ are appropriately set. However, if they are too large, they can easily fall into local optimal. To enhance the readability of our method, we present pseudocode for model training in Algorithm 1.

IV. EXPERIMENTS

In this section, we first present the experimental setup and evaluation metrics. Next, we demonstrate the superiority of our proposed models through comparisons with the baselines. Finally, we focus on analyzing the effectiveness of the models with respect to partial fulfillment and their generalization capabilities.

A. EXPERIMENTAL SETUP

In our experiment, we train models with 640,000 samples and run the augmented Lagrangian solver for a maximum of 80 epochs. The allocation and pricing networks in PDNet each consist of three layers, with 100 hidden units per layer. In comparison, EANet comprises two self-attention layers, each with 2 heads and 64 hidden features. We update the parameter ρ every two epochs and λ_{ic} and λ_{ir} every 100

Algorithm 1: Training process

```

Input: Minibatches  $S_1, \dots, S_T$  of size  $B$ 
Parameters:  $\rho^t, \gamma, \eta \in \mathbb{R}, \Gamma \in \mathbb{N}$ 
Initialize:  $w^0 \in \mathbb{R}^d, \lambda^0 \in \mathbb{R}^n, o \in \mathbb{R}^{n \times m}, q \in \mathbb{R}^m$ 
for  $t \leq T$  do
    Receive minibatch  $S_t = \{v^{(1)}, \dots, v^{(B)}\}$ ;
    Initialize  $w^0 \in \mathbb{R}^d, \lambda^0 \in \mathbb{R}^n, q \in \mathbb{R}^m$ ;
    for  $\ell \leq B$  do
         $v'^\ell \leftarrow v^\ell + \gamma \nabla_{v'} u_i^w(v_i^{(\ell)}; (v_i^{(\ell)}, v_{-i}^{(\ell)}))$ ;
        Compute and update regret gradient;
         $w^{t+1} \leftarrow w^t - \eta \nabla_w \mathcal{L}(w^t, \lambda^t)$ ;
        Update Lagrange multipliers every iteration;
        if  $t$  is a multiple of  $Q_{ic}$  then
             $\lambda_{ic}^{t+1} \leftarrow \lambda_{ic}^t + \rho^t ic_i(w^{t+1}), \forall i \in N$ ;
        else
             $\lambda_{ic}^{t+1} \leftarrow \lambda_{ic}^t$ ;
        if  $t$  is a multiple of  $Q_{se}$  then
             $\lambda_{se}^{t+1} \leftarrow \lambda_{se}^t + \rho^t se_j(w^{t+1}), \forall j \in M$ ;
        else
             $\lambda_{se}^{t+1} \leftarrow \lambda_{se}^t$ ;

```

iterations. The trade-off coefficient $\alpha \in [0, 1]$. Finally, we utilize the Adam optimizer with a learning rate of 0.001 to update w^t every minibatch. The experiments are conducted on a Linux platform equipped with an NVIDIA GeForce RTX 4090GPU. By default, we conduct each experiment three times and report the average outcome.

B. EVALUATION METRICS

In order to comprehensively evaluate the performance of the proposed models in edge computing resource auctions, We employ the following four metrics, all averaged across all bidders: empirical revenue (rev), empirical regret (rgt), individual rationality measure (irm), and supply-demand equilibrium measure (sem).

$$rev = \frac{1}{|L|} \sum_{\ell=1}^{|L|} \sum_{i=1}^n p_i \quad (34)$$

$$rgt = \frac{1}{n} \sum_{i=1}^n \widehat{rgt}_i \quad (35)$$

$$irm = \frac{1}{|L|} \sum_{\ell=1}^{|L|} \sum_{i=1}^n \max\{0, -u_i(v^\ell)\} \quad (36)$$

$$sem = \frac{1}{|L|} \sum_{\ell=1}^{|L|} \sum_{j=1}^m se_j \quad (37)$$

C. BASELINES

To illustrate the efficacy of our proposed models, we perform a comparative analysis with mainstream approaches, including

- **GSP** [30]: This is a typical auction mechanism, where bidders are ranked according to their bids, but each winning bidder pays the price of the next bidder after her ranking.
- **VCG** [31]–[33]: This is a mechanism design approach that maximizes social welfare by incentivizing bidders to reveal their valuations truthfully.
- **RegretNet** [19]: This is a neural network-based auction mechanism that learns allocation and pricing rules in an end-to-end manner.

D. EXPERIMENTAL RESULTS

1) Performance Comparison

We design a series of experimental environments to comprehensively evaluate the potential of models in multi-unit auction scenarios. Specifically, we consider competitive auction settings in which the total demand exceeds the total supply:

(I) $n = 2$ bidders and $m = \{1, 2, 3, 4\}$ resources, with $v_{ij} \sim U[0, 1]$. The quantity demanded by a bidder for a resource is drawn independently from $U[1, 4]$, and the quantity of a resource supplied is drawn independently from $U[1, 7]$.

(II) $n = 3$ bidders and $m = \{1, 2, 3, 4\}$ resources. The quantity of a resource supplied is drawn independently from $U[1, 9]$. The quantities demanded and the corresponding values are drawn similarly to Setting I.

(III) $n = 4$ bidders and $m = \{1, 2, 3, 4\}$ resources. The quantity of a resource supplied is drawn independently from $U[1, 11]$. The quantities demanded and the corresponding values are drawn similarly to Setting I.

The comparison results of various methods in different settings are presented in **Table 1**. The experimental data shows that the VCG has the worst performance, which is due to the fact that the goal of VCG is to maximize social welfare rather than the seller's revenue. RegretNet exhibits better performance compared to GSP. In contrast to GSP, which employs fixed pricing rules, the neural network-based architecture demonstrates greater adaptive ability to better cope with complex market environments. Our learned mechanisms (PDNet, EANet) achieve the highest revenue with negligible regret in all settings. The advantage expands with market complexity (larger n or m), where fine-grained rationing across bidder–resource pairs and data-driven pricing better exploit competition and supply heterogeneity—particularly in 4×3 , 4×4 blocks of Table 1. Joint learning of allocation probabilities and fulfilled quantities enables reallocating marginal units toward higher-value matches, explaining the widening gap over GSP/VCG and the gains over RegretNet when partial-quantity decisions are pivotal.

EANet consistently surpasses PDNet, with a larger margin as the auction scale increases. Self-attention captures non-local bidder–resource interactions (cross-bidder externalities and cross-resource substitution/complementarity), while exchangeable layers enforce permutation equivariance and parameter sharing across bidders/resources. This inductive bias scales effectively with participants and resources, yielding

stronger performance in settings with dense interactions and heterogeneous supply.

2) Verification of Desirable Properties

To measure the extent to which the models comply with desirable properties, we evaluated their performance with respect to DSIC, IR and SE across different environmental settings.

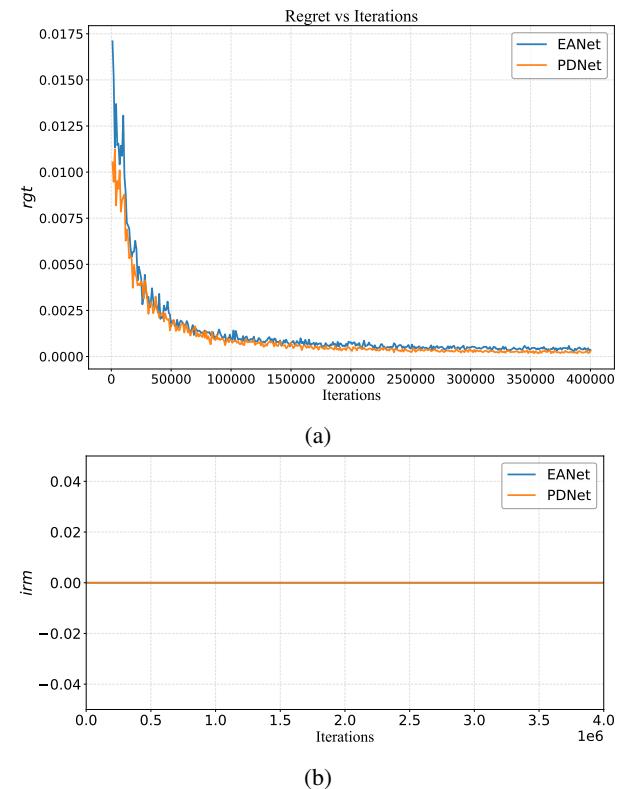


FIGURE 4: The variations of desirable properties during training. (a) The variation of rgt with the number of iterations; (b) The variation of irm with the number of iterations.

Table 2 presents the evaluation results for the proposed models across various settings. The experimental results show that rgt and sem for both models approach zero, while the irm consistently remains zero. **Figure 4** illustrates the trends of rgt and irm of the models throughout the training process. It is worth noting that the irm always remains 0. This phenomenon can be effectively elucidated from the perspective of model design: since the payment probabilities output from the payment network are constrained within the range of 0 to 1, the utility of the bidder is necessarily kept non-negative, thus satisfying IR. Overall, these results indicate that the proposed models possess desirable properties.

3) Empirical Validation of Partial Fulfillment

To evaluate the capability of our models to achieve partial fulfillment under different supply and demand conditions, we consider two scenarios: supply scarcity and supply abundance. All experimental results reported in this section are

TABLE 1: Experimental results of the proposed models compared to baselines across different settings.

Settings	GSP		VCG		RegretNet		PDNet		EANet	
	rev	r _{gt}	rev	r _{gt}	rev	r _{gt}	rev	r _{gt}	rev	r _{gt}
(I)	2 × 1	0.83	-	0.73	-	1.04	< 0.001	1.127	< 0.001	1.142
	2 × 2	1.67	-	1.38	-	1.81	< 0.001	2.052	< 0.001	2.071
	2 × 3	2.53	-	2.05	-	2.69	< 0.001	2.828	< 0.001	2.842
	2 × 4	3.38	-	2.83	-	3.47	< 0.001	3.651	< 0.001	3.668
(II)	3 × 1	1.19	-	1.14	-	1.41	< 0.001	1.677	< 0.001	1.701
	3 × 2	2.38	-	2.26	-	2.64	< 0.001	2.985	< 0.001	3.011
	3 × 3	3.64	-	3.34	-	3.98	< 0.001	4.111	< 0.001	4.136
	3 × 4	4.90	-	4.55	-	5.05	< 0.001	5.267	< 0.002	5.294
(III)	4 × 1	1.22	-	1.37	-	1.62	< 0.001	2.196	< 0.001	2.228
	4 × 2	2.81	-	2.68	-	3.55	< 0.001	3.943	< 0.001	3.975
	4 × 3	4.11	-	3.88	-	4.66	< 0.002	5.475	< 0.002	5.507
	4 × 4	5.22	-	5.22	-	5.87	< 0.002	6.892	< 0.003	6.924

TABLE 2: Evaluation results for desirable properties.

Settings	PDNet			EANet		
	r _{gt}	irm	sem	r _{gt}	irm	sem
(I) 2 × 2	0.000479	0.000	0.0023	0.000842	0.000	0.0019
(II) 3 × 2	0.000542	0.000	0.0019	0.001648	0.000	0.0024
(III) 4 × 2	0.000538	0.000	0.0016	0.002654	0.000	0.0018

averaged over all testing epochs to ensure statistical reliability.

First, we examine settings with both symmetric and asymmetric valuations in supply scarcity:

(IV) $n = 2$ bidders and $m = 4$ resources, with $v_{ij} \sim U[0, 1]$. The quantity demanded by a bidder for a resource is drawn independently from $U[1, 4]$, and the quantity of a resource supplied is drawn independently from $U[1, 7]$.

(V) $n = 2$ bidders and $m = 4$ resources, with $v_{1,\{1,2,3,4\}} \sim U[0, 1]$ and $v_{2,\{1,2,3,4\}} \sim U[1, 2]$. The quantities demanded and supplied are drawn similarly to Setting IV.

The experimental results presented in **Figure 5a** and **Figure 6a** illustrate the allocation of resources across various valuation scenarios. The experimental results show that there is a significant positive correlation between the probability of obtaining resources and the bidders' valuations. Specifically, higher valuations lead to a greater probability of resource acquisition and an increased quantity of resources obtained. This allocation pattern is highly consistent with economic intuition. It is worth noting that due to the introduction of virtual nodes to represent cases where resources may not be allocated, this leads to the total allocation probabilities for all bidders being less than one. In addition, Figure 5b and 6b indicate that the model exhibits a strong capability for partial fulfillment. Specifically, the quantity of resources actually allocated to each bidder remains strictly within their specified requirements. Meanwhile, the total quantities acquired by all bidders for a given type of resource approach the supply ceiling for that resource.

Subsequently, we examine both symmetric and asymmetric valuation settings under supply abundant conditions:

(VI) $n = 2$ bidders and $m = 4$ resources, with $v_{ij} \sim U[0, 1]$.

The quantity demanded by a bidder for a resource is drawn independently from $U[1, 4]$, and the quantity of a resource supplied is drawn independently from $U[1, 11]$.

(VII) $n = 2$ bidders and $m = 4$ resources, with $v_{1,\{1,2,3,4\}} \sim U[0, 1]$ and $v_{2,\{1,2,3,4\}} \sim U[1, 2]$. The quantities demanded and supplied are drawn similarly to Setting IV.

As illustrated by the results in **Figure 7** and **Figure 8**, the model is able to fully exploit the potential of resources under supply-abundant conditions. Specifically, while ensuring that the total resources allocated to all bidders do not exceed the total supply, each bidder's actual allocation approaches their declared demand. This demonstrates that the model effectively balances supply and demand, enabling efficient allocation and utilization of resources.

4) Impact of Supply and Demand Relationships

To further investigate how demand-supply affects auction outcomes, we introduce the demand-supply ratio $\beta = \frac{\sum_{i=1}^n o_{ij}}{q_j}$ to simulate a range of market environments. The experimental settings are as follows:

(VIII) $n = 2$ bidders and $m = \{1, 2, 3, 4\}$ resources, with $v_{ij} \sim U[0, 1]$.

1) Demand exceeds supply ($\beta > 1$): The quantity demanded by a bidder for a resource is drawn independently from $U[1, 4]$, and the quantity of a resource supplied is drawn independently from $U[1, 4]$.

2) Demand matches supply ($\beta = 1$): The quantity of a resource supplied is drawn independently from $U[1, 9]$. The quantity demanded is drawn similarly to 1).

3) Demand falls below supply ($\beta < 1$): The quantity of a resource supplied is drawn independently from $U[1, 11]$. The

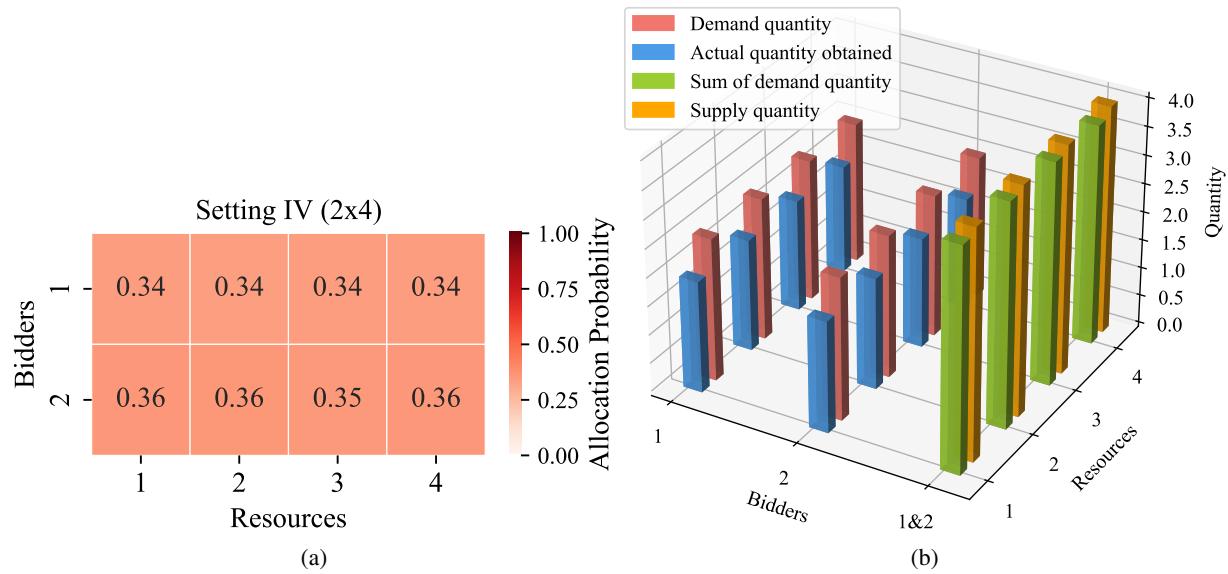


FIGURE 5: Experimental results of PDNet for symmetric valuation in Setting IV. (a) Allocation-probability heatmap: the x-axis denotes resources, the y-axis denotes bidders. Each cell shows the allocation probability, with darker color indicating higher probability. (b) Partial-fulfillment view: the x-axis denotes bidders, the y-axis denotes resources, and the z-axis denotes quantities. Red bars denote demanded quantities, blue bars the quantities actually allocated to bidders, green bars the combined demand of both bidders, and yellow bars the available supply.

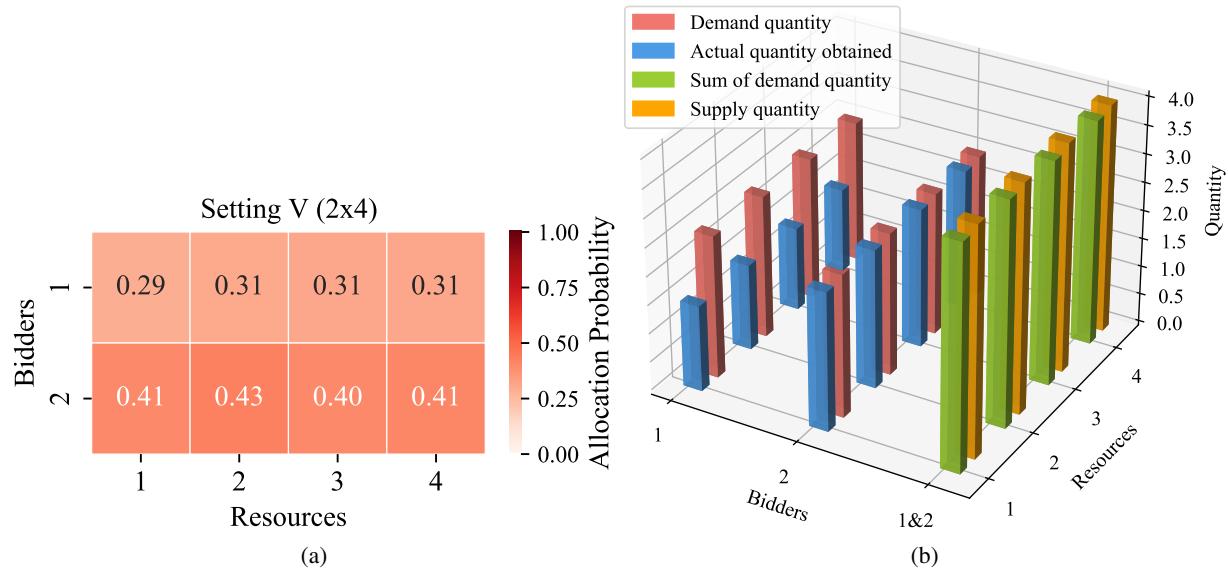


FIGURE 6: Experimental results of PDNet for asymmetric valuation in Setting V. (a) Allocation-probability heatmap: the x-axis denotes resources, the y-axis denotes bidders. Each cell shows the allocation probability, with darker color indicating higher probability. (b) Partial-fulfillment view: the x-axis denotes bidders, the y-axis denotes resources, and the z-axis denotes quantities. Red bars denote demanded quantities, blue bars the quantities actually allocated to bidders, green bars the combined demand of both bidders, and yellow bars the available supply.

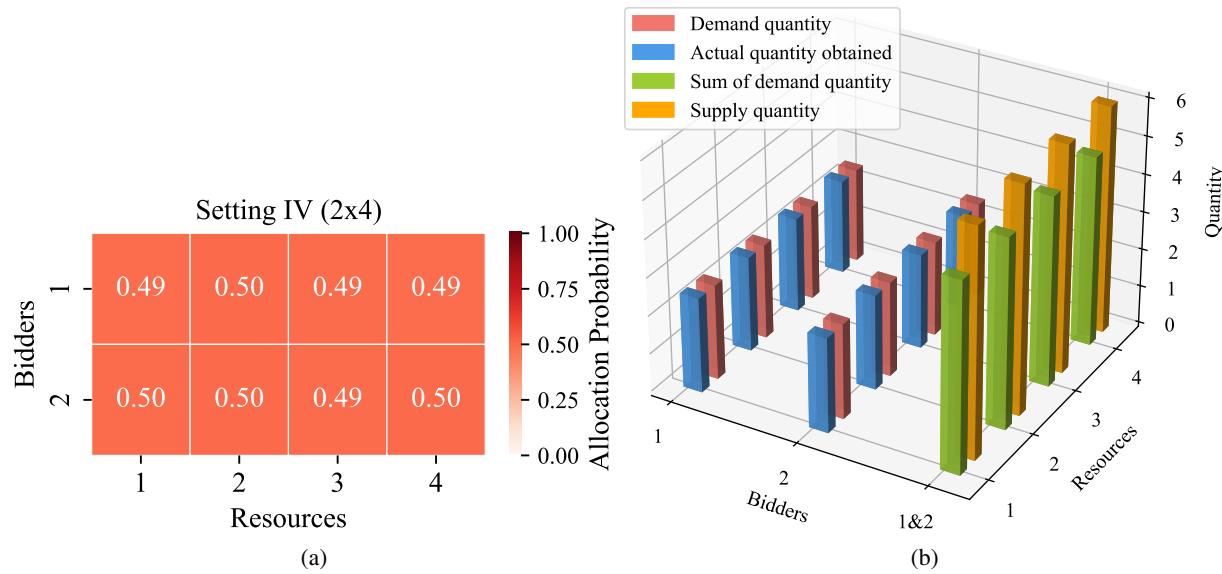


FIGURE 7: Experimental results of EANet for symmetric valuation in Setting IV. (a) Allocation-probability heatmap: the x-axis denotes resources, the y-axis denotes bidders. Each cell shows the allocation probability, with darker color indicating higher probability. (b) Partial-fulfillment view: the x-axis denotes bidders, the y-axis denotes resources, and the z-axis denotes quantities. Red bars denote demanded quantities, blue bars the quantities actually allocated to bidders, green bars the combined demand of both bidders, and yellow bars the available supply.

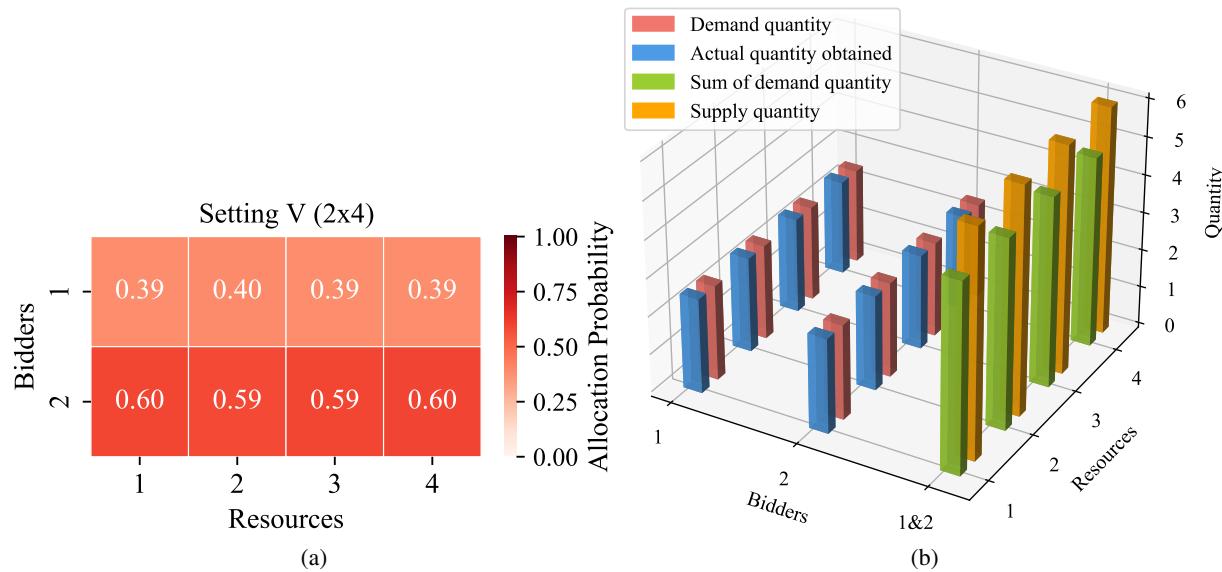


FIGURE 8: Experimental results of EANet for asymmetric valuation in Setting V. (a) Allocation-probability heatmap: the x-axis denotes resources, the y-axis denotes bidders. Each cell shows the allocation probability, with darker color indicating higher probability. (b) Partial-fulfillment view: the x-axis denotes bidders, the y-axis denotes resources, and the z-axis denotes quantities. Red bars denote demanded quantities, blue bars the quantities actually allocated to bidders, green bars the combined demand of both bidders, and yellow bars the available supply.

quantity demanded is drawn similarly to 1).

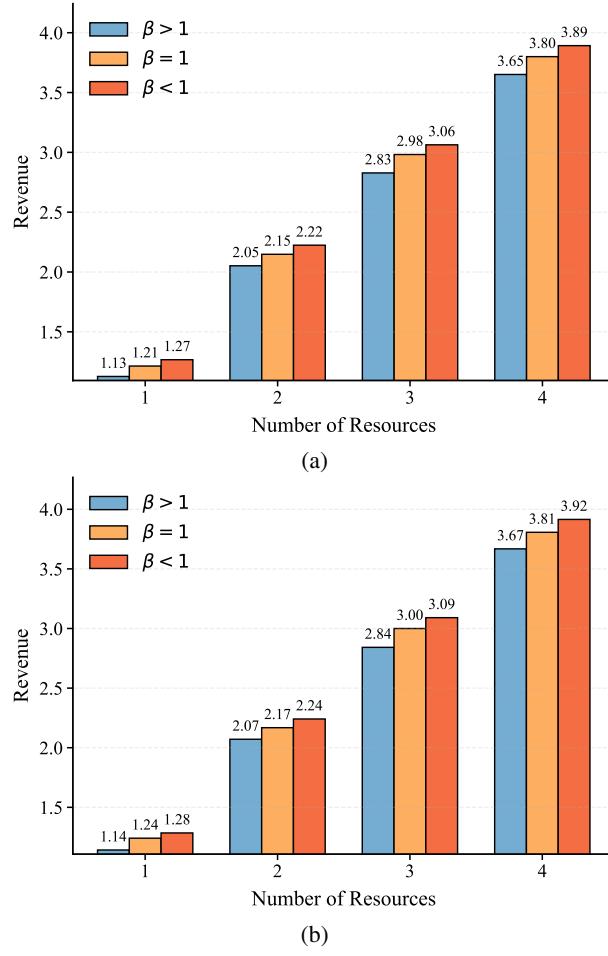


FIGURE 9: Performance of models in different market environments. (a) Revenue performance of PDNet in Setting VIII; (b) Revenue performance of EANet in Setting VIII.

Figure 9 illustrates how demand–supply conditions affect auction outcomes. When demand exceeds supply, intense competition induces aggressive bidding but also amplifies allocative frictions, leading to the lowest revenue among the three environments as capacity binds and feasible reallocations are exhausted. As demand gradually aligns with supply, the market approaches a better state: allocations better match bidder valuations, penalty terms are rarely activated, and revenue improves steadily. When demand falls below supply, the seller’s revenue continues to grow but at a decreasing rate. This occurs because, under ample capacity, competition for marginal units weakens, allowing them to clear at lower prices, which compresses incremental revenue. In conclusion, the proposed models exhibit strong adaptability and effectiveness across market environments.

5) Generalization Beyond Training Configurations

To further demonstrate the generalization of our models, we design experiments in which the models are evaluated under configurations different from those used during training. In

particular, we train our models in a fixed auction environment and evaluate their performance when the number of bidders or resources varies. We consider the following two experimental settings:

(IX) The model is trained in an environment with $n = 2$ bidders and $m = 1$ resource, where each bidder’s demand for the resource is independently drawn from $U[1, 4]$, the resource’s supply is independently drawn from $U[1, 7]$, and each bidder’s value for the resource is independently drawn from $U[0, 1]$.

(X) The model is trained in an environment with $n = 3$ bidders and $m = 3$ resources, where the demand for each bidder, the supply of each resource, and the valuation of each bidder are sampled in the same way as in Setting IX.

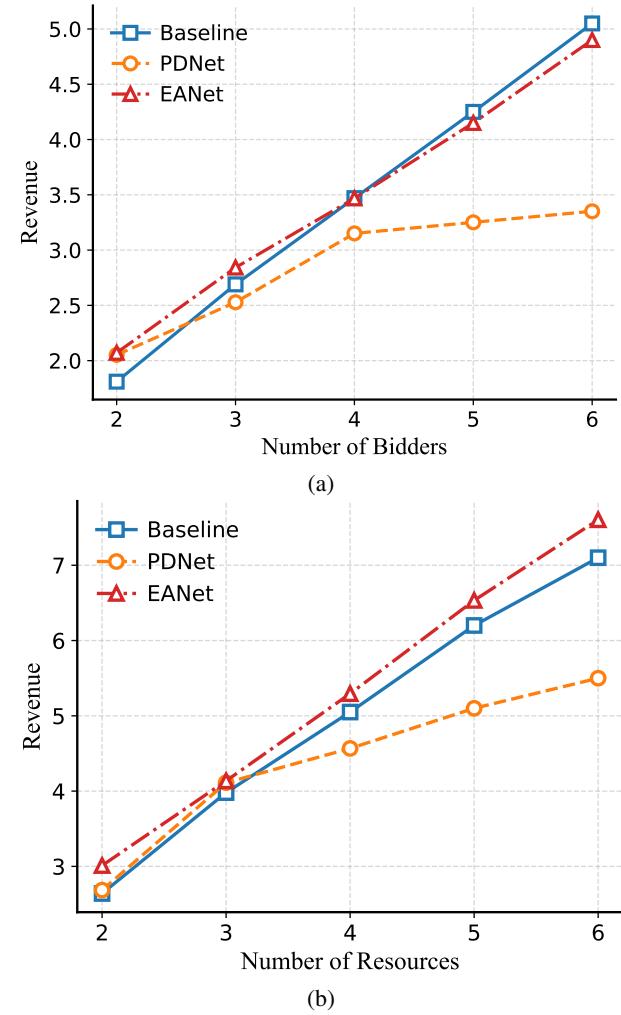


FIGURE 10: Generalization results of the models beyond training configurations. We select RegretNet as the baseline, where each data point is obtained through its complete training. (a) Model trained on Setting IX and evaluated with different numbers of bidders. (b) Model trained on Setting X and evaluated under different numbers of resources.

Figure 10 presents the test revenue achieved as the number of bidders and resources changes. The experimental results

indicate that PDNet exhibits limited generalization ability. This is primarily due to the fact that the dimensionality of the weights in its feed-forward architecture is dependent on the input size, leading to poor generalization when the input size changes. In contrast, the EANet demonstrates strong generalization beyond training configurations, benefitting from its parameterization being independent of the input size. Specifically, EANet exhibits excellent generalization performance when the number of resources varies and the number of bidders is fixed. However, its generalization ability is slightly compromised when the number of bidders changes. This may be attributed to the more complex interactions arising among bidders in such scenarios. Additional experiments can be found in Appendix C.

6) Ablation and Sensitivity Analysis

a: Ablation study on self-Attention

To quantify the contribution of the self-Attention module in EANet, we perform an ablation in which we replace all self-Attention blocks with multilayer perceptrons, yielding the variant EANet-MLP. All other components, training schedules, and hyperparameters are kept identical. As summarized in **Figure 11**, EANet-MLP exhibits a drop in revenue across different settings, with the largest degradation observed on the high-competition settings. These findings indicate that self-attention—by adaptively aggregating bidder and resource information to capture interactions—plays a central role in EANet’s performance.

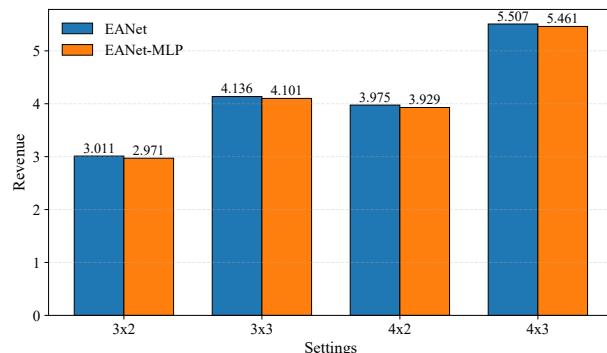


FIGURE 11: Revenue comparison between EANet and EANet-MLP across auction settings where 3×2 and 3×3 correspond to Setting I and 4×2 and 4×3 correspond to Setting II.

b: Impact of hidden sizes

In PDNet, we denote an architecture by (H, K) (with H hidden layers and K nodes per layer). **Figure 12a** reports revenue and regret under Setting IV on a validation set of 5,000 profiles. The $(5, 100)$ network attains the lowest regret among 100-node models. We observe that increasing the number of layers generally improves performance. We attribute this gain to the deeper model’s greater capacity to approximate the complex, piecewise-linear allocation and payment rules

inherent in auction mechanisms. However, extremely deep architectures may not yield additional gains due to optimization difficulties and overfitting, which explains why performance saturates at around five layers in our setting.

c: Impact of exchangeable layer depth

We vary the depth of the exchangeable layers in $\{16, 32, 64, 128\}$ to evaluate whether EANet benefits from increased depth. Figure 12b reports the results. Increasing the depth from 16 to 64 delivers a substantial performance gain, while pushing to 128 yields only marginal improvement at a much higher computational cost. Thus, a moderate depth strikes a better balance between performance and efficiency.

d: Impact of trade-off coefficient

We conduct a sensitivity analysis on the trade-off coefficient α . As shown in Figure 12c, performance varies non-monotonically with α , reaching a peak around 0.4 for PDNet and around 0.6 for EANet. Both extremes hurt revenue: with too small α , the optimizer over-enforces feasibility, yielding conservative allocations and under-exploited revenue; with too large α , training operates near constraint boundaries, triggering noisy penalties and unstable convergence. A mid-range α balances feasibility and exploration, yielding the best performance.

V. CONCLUSION

In this paper, we propose a deep learning-based auction mechanism that supports partial fulfillment in edge computing resource trading markets. Specifically, we design two novel models, PDNet and EANet, to address the challenge of optimizing resource allocation under complex application demands. Unlike the traditional all-or-nothing resource allocation mechanism, our approach not only flexibly satisfies partial demands from application service providers, but also ensures economic properties such as incentive compatibility, individual rationality, and supply-demand equilibrium. Through an end-to-end neural network framework, the proposed models automatically learn optimal allocation and payment strategies, achieving efficient resource utilization.

Extensive experimental results demonstrate that both PDNet and EANet significantly enhance the revenue of edge service providers, outperforming the existing mainstream mechanisms. Especially, EANet not only outperforms in terms of revenue, but also exhibits additional advantages, including permutation equivariance and strong generalization to unseen settings with variable input sizes. In addition, the experiments also verify the effectiveness of the proposed models for partial fulfillment and market regulation under different supply and demand conditions.

Our model currently assumes additive, i.i.d. valuations, and we evaluate it on synthetic data. Although we have not yet modeled correlated preferences or hard budget constraints, the framework is flexible and can be extended. As natural next steps, we plan to incorporate budget constraints, extend to multi-round/dynamic markets with intertemporal dependence

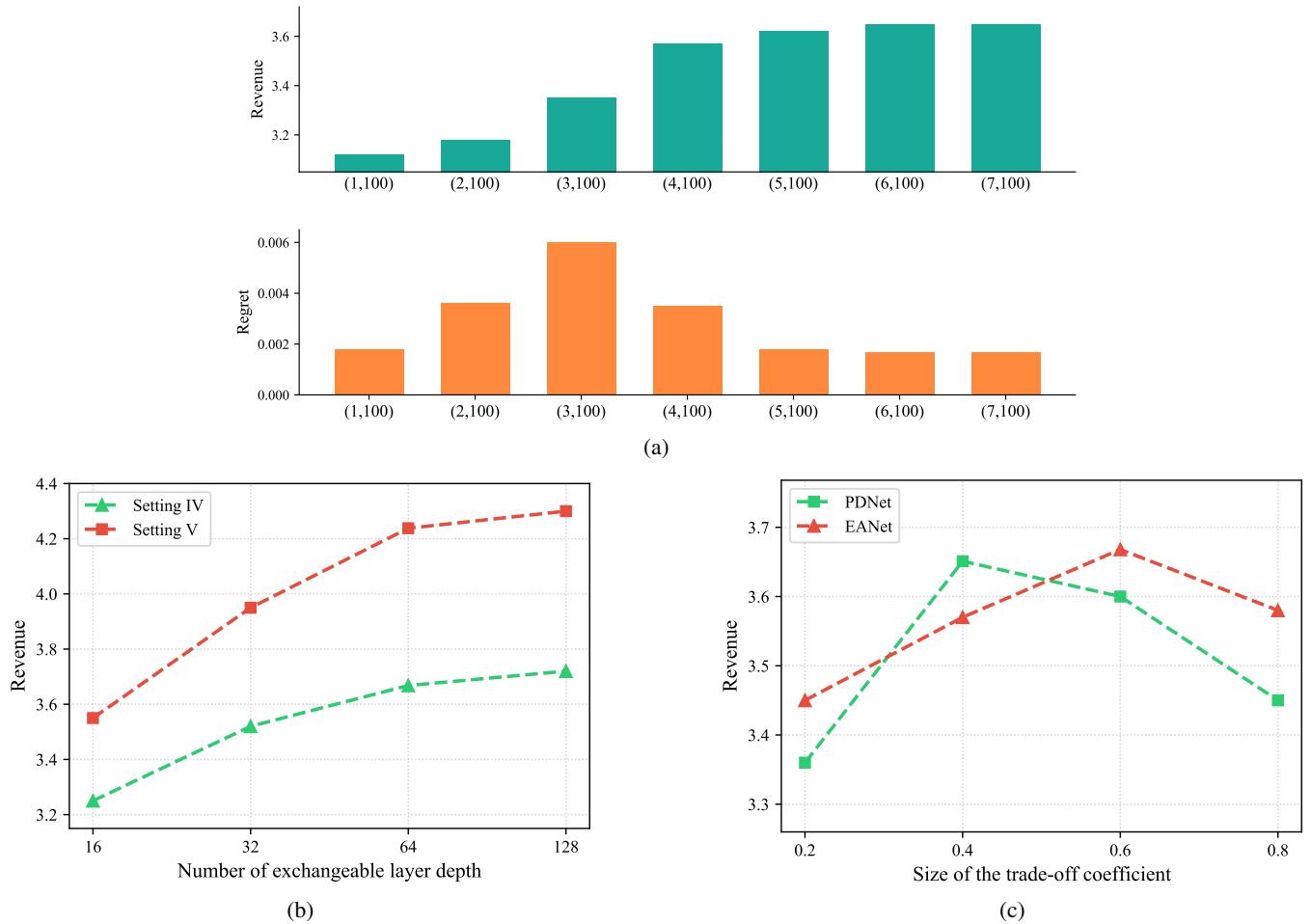


FIGURE 12: Sensitivity analysis results. (a) Revenue and regret for PDNet using various architectures for Setting IV; (b) The impact of different exchangeable layer depths in EANet for Setting IV-V; (c) The impact of trade-off coefficient for Setting IV.

and strategic behavior, and adopt privacy-preserving training and deployment (e.g., differential privacy, federated learning, secure aggregation) to protect bidder data.

APPENDIX A OMITTED PROOFS

A. PROOF OF THEOREM 1

Proof. The proof of Theorem 1 is carried out in two steps:

(i) Sufficient Condition:

If $\mathcal{M}(g, d, p)$ is DSIC, then for any bidder i , any profile $(v_i, v_{-i}) \in \mathcal{V}$, and any deviation $v'_i \in \mathcal{V}_i$,

$$u_i(v_i; (v_i, v_{-i})) \geq u_i(v_i; (v'_i, v_{-i})).$$

Equivalently, for each (v_i, v_{-i}) ,

$$\max_{v'_i \in \mathcal{V}_i} \{u_i(v_i; (v'_i, v_{-i})) - u_i(v_i; (v_i, v_{-i}))\} \leq 0.$$

The maximand is nonnegative by definition of the maximum, hence the maximum must be exactly 0 pointwise. Taking expectation over $v \sim D$ yields $rgt_i(v) = 0$ for all i .

(ii) Necessary Condition:

Fix $i \in N$ and define the nonnegative function

$$R_i(v) = \max_{v'_i \in \mathcal{V}_i} \{u_i(v_i; (v'_i, v_{-i})) - u_i(v_i; (v_i, v_{-i}))\} \geq 0.$$

By assumption,

$$\mathbb{E}_{v \sim F}[R_i(v)] = rgt_i(v) = 0.$$

A nonnegative random variable has zero expectation if and only if it is zero almost surely, hence $R_i(v) = 0$ for D -almost every $v \in \mathcal{V}$.

Full support of D then implies $R_i(v) = 0$ for every $v \in \mathcal{V}$. Consequently, for all $(v_i, v_{-i}) \in \mathcal{V}$ and all $v'_i \in \mathcal{V}_i$,

$$u_i(v_i; (v_i, v_{-i})) \geq u_i(v_i; (v'_i, v_{-i})),$$

which is precisely the DSIC condition. Hence, the mechanism is DSIC. \square

B. PROOF OF THEOREM 2

Proof. By the definition of payment and the range of $\tilde{p}_i(b)$, we have for each i :

$$p_i(b) = \tilde{p}_i(b) \sum_{j=1}^m g_{ij}(b) d_{ij}(b) b_{ij} \leq \sum_{j=1}^m g_{ij}(b) d_{ij}(b) b_{ij},$$

since $0 \leq \tilde{p}_i(b) \leq 1$ and each summand is nonnegative.

Under additive valuations and $b_{ij} \leq v_{ij}$, we have for any allocation and demand realization:

$$\sum_{j=1}^m g_{ij}(b) d_{ij}(b) v_{ij} \geq \sum_{j=1}^m g_{ij}(b) d_{ij}(b) b_{ij}.$$

Therefore, bidder i 's utility satisfies

$$\begin{aligned} u_i(v_i, b) &= \sum_{j=1}^m g_{ij}(b) d_{ij}(b) v_{ij} - p_i(b) \\ &\geq \sum_{j=1}^m g_{ij}(b) d_{ij}(b) b_{ij} - \tilde{p}_i(b) \sum_{j=1}^m g_{ij}(b) d_{ij}(b) b_{ij} \\ &= (1 - \tilde{p}_i(b)) \sum_{j=1}^m g_{ij}(b) d_{ij}(b) b_{ij} \geq 0, \end{aligned}$$

because $1 - \tilde{p}_i(b) \geq 0$ and each term in the sum is nonnegative. Taking $b_i = v_i$ yields

$$u_i(v_i, (v_i, b_{-i})) \geq 0,$$

which establishes individual rationality. \square

APPENDIX B SELF-ATTENTION ARCHITECTURE

We model bidder–resource interactions with a multi-head self-Attention encoder that aggregates information over a set of tokens while preserving permutation equivariance.

Let the token sequence be

$$Z = (z_1, \dots, z_n)^\top \in \mathbb{R}^{n \times d},$$

where each $z_i \in \mathbb{R}^d$ encodes either a bidder (when attending over resources) or a resource (when attending over bidders).

Choose a hidden size d_h and number of heads H . For each head $h \in [H]$, project queries, keys, and values:

$$q_i^{(h)} = W_Q^{(h)} z_i, \quad k_j^{(h)} = W_K^{(h)} z_j, \quad v_j^{(h)} = W_V^{(h)} z_j,$$

where $W_Q^{(h)}, W_K^{(h)}, W_V^{(h)} \in \mathbb{R}^{d \times d'}$ denote the learnable matrices that convert embeddings from the dimension d into the subspace with dimension d' .

Then, we compute attention weights for token i :

$$\alpha_{i,j}^{(h)} = \frac{\exp(\langle q_i^{(h)}, k_j^{(h)} \rangle)}{\sum_{k=1}^n \exp(\langle q_i^{(h)}, k_k^{(h)} \rangle)},$$

where $\langle \cdot, \cdot \rangle$ denotes inner product.

The head output for token i is

$$u_i^{(h)} = \sum_{j=1}^n \alpha_{i,j}^{(h)} v_j^{(h)} \in \mathbb{R}^{d'}.$$

After computing all heads, we concatenate their outputs:

$$\tilde{z}_i = u_i^{(1)} \oplus \dots \oplus u_i^{(H)} \in \mathbb{R}^{d_h}.$$

Next, the concatenated representation is processed by a token-level feed-forward network:

$$z'_i = \text{MLP}(\tilde{z}_i) \in \mathbb{R}^{d_h}.$$

Finally, aggregating the per-token outputs gives:

$$Z' = (z'_1, \dots, z'_n)^\top \in \mathbb{R}^{n \times d_h}.$$

Notably, the learnable components include the projection matrices $W_{\text{query}}^{(h)}, W_{\text{key}}^{(h)}, W_{\text{value}}^{(h)} \in \mathbb{R}^{d' \times d}$ for each head, along with the parameters of the token-wise MLP. None of these depend on the token count n .

APPENDIX C

ADDITIONAL EXPERIMENTS

A. CONSTRAINT-COMPLIANCE DIAGNOSTICS

We report constraint-compliance diagnostics beyond point estimates by analyzing the across-batch distribution of regret (rgt). **Figure 13** summarizes rgt with boxplots over training iterations, grouped into 50k-step buckets. Phase-wise, the 50k-bucket boxplot shows that high- rgt outliers are concentrated in the earliest bucket (0–50k), which also exhibits the widest spread. Subsequent buckets display tight boxes near the lower bound with few or no outliers, indicating rapid stabilization and sustained compliance as training proceeds. Thus, the diagnostics reveal: (i) the majority of batches satisfy the constraint, (ii) non-compliant episodes are primarily early-training transients, and (iii) once past the initial phase, both central tendency and variability of rgt remain low.

Additionally, we quantify constraint compliance by aggregating violation rates across epochs for IR and SE. The violation rate is defined as the fraction of evaluation instances that violate a constraint within a reporting window. Specifically, for each 10-epoch window $[e, e + 9]$, we compute $\text{ViolationRate} = \sum_{t=e}^{e+9} N_{\text{viol}}^{(t)} / \sum_{t=e}^{e+9} N_{\text{total}}^{(t)}$ for IR (irm) and SE (sem). Results in **Table 3** show that IR remains at 0% throughout, while SE violations concentrate in early epochs and fall below 1% in later epochs.

B. SENSITIVITY TO LAGRANGE MULTIPLIERS AND PENALTY COEFFICIENT

We study the joint effect of the initial Lagrange multiplier λ_0 and the penalty term ρ on final performance under augmented Lagrangian training. We take λ_{ic} as the representative instance of λ_0 (the analysis for λ_{se} is analogous). As illustrated in **Figure 14**, we sweep (λ_0, ρ) and report two 2D heatmaps for final revenue and mean regret. The default ($\lambda_0 = 5.0$, $\rho = 1.0$) lies near the center of a broad plateau where revenue is high and regret is low, consistent with the training dynamics implied by the update rule (Equation 32). Moving away from this region yields predictable patterns: when ρ is too small, regret remains under-penalized and the mechanism may exhibit slightly inflated nominal revenue due to

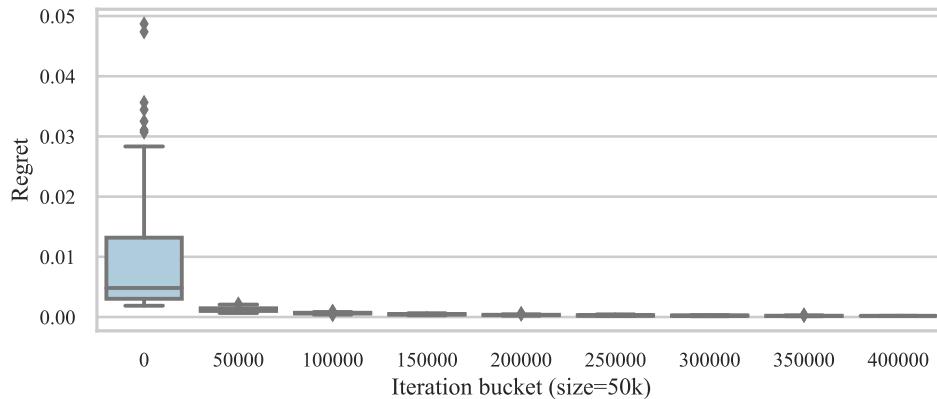


FIGURE 13: Regret distribution of EANet under Setting IV.

TABLE 3: IR/SE violation rates per 10 epochs (lower is better).

Setting		PDNet		EANet	
		IR viol./10epoch	SE viol./10epoch	IR viol./10epoch	SE viol./10epoch
IV	1–10	0.000	0.075	0.000	0.085
	11–20	0.000	0.025	0.000	0.030
	21–30	0.000	0.008	0.000	0.015
	31–80	0.000	0.001	0.000	0.001

residual regret; when ρ is too large, the quadratic penalty and induced multiplier updates dominate, leading to conservative or oscillatory optimization that lowers revenue at comparable regret levels. Similarly, extreme λ_0 either underprices regret early (too small) or overconstraints the search (too large), both of which can impede learning high-revenue structure before regret is tightened. Overall, once regret is driven near zero, final revenue varies weakly across a sizeable neighborhood of (λ_0, ρ) .

APPENDIX D TINY WORKED EXAMPLE

We present a minimal numerical example with two bidders and two resources. We illustrate allocation with partial fulfillment, compute payments using the module $p_i = \tilde{p}_i \sum_{j \in M} g_{ij} d_{ij} b_{ij}$, and verify IC informally, IR, and SE. The construction uses intuitive numbers and mirrors a second-price-like intuition through the choice of g_{ij} .

We consider bidders $N = \{1, 2\}$ and resources $M = \{A, B\}$. The resource capacities are $q_A = 3$ and $q_B = 2$. The maximum consumable quantities (demands) are $o_{1A} = 2$, $o_{1B} = 2$, $o_{2A} = 2$, and $o_{2B} = 1$. We assume truthful bidding with per-unit bids $b_{1A} = 10$, $b_{1B} = 6$, $b_{2A} = 8$, and $b_{2B} = 9$. Partial fulfillment is allowed, i.e., $d_{ij} \in [0, o_{ij}]$, subject to capacity constraints $\sum_i d_{ij} \leq q_j$ for each $j \in M$.

We construct the allocation by prioritizing higher per-unit bids on each resource, respecting both capacity and individual demand. For resource A with capacity $q_A = 3$, bidder 1 has the higher bid (10) and is allocated $d_{1A} = 2$ units (saturating o_{1A}), leaving a residual capacity of 1. Bidder 2 then receives the residual $d_{2A} = 1$. For resource B with capacity $q_B = 2$, bidder 2 has the higher bid (9) and is allocated $d_{2B} = 1$

unit (saturating o_{2B}), leaving a residual capacity of 1 that is allocated to bidder 1 as $d_{1B} = 1$. The final allocation is therefore

$$d = \begin{bmatrix} d_{1A} & d_{1B} \\ d_{2A} & d_{2B} \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix},$$

$$\sum_i d_{iA} = 3 \leq q_A, \quad \sum_i d_{iB} = 2 \leq q_B, \quad d_{ij} \leq o_{ij}.$$

This allocation illustrates partial fulfillment: e.g., $o_{2A} = 2$ but $d_{2A} = 1$, and $o_{1B} = 2$ but $d_{1B} = 1$.

Payments are computed using the module

$$p_i = \tilde{p}_i \sum_{j \in M} g_{ij} d_{ij} b_{ij},$$

where $\tilde{p}_i \geq 0$ is a bidder-level multiplier and $g_{ij} \in [0, 1]$ is a per-resource scaling factor. To preserve intuitive, rival-indexed pricing while remaining within the module, we set $\tilde{p}_1 = \tilde{p}_2 = 1$ and choose $g_{iA} = 0.8$ for all units allocated in A (reflecting the rival's bid 8 relative to 10), and $g_{iB} = 2/3$ for all units allocated in B (reflecting the rival's bid 6 relative to 9). Under these factors, the payments specialize to

$$p_1 = 1 \cdot (0.8 \cdot 2 \cdot 10 + \frac{2}{3} \cdot 1 \cdot 6) = 16 + 4 = 20,$$

$$p_2 = 1 \cdot (0.8 \cdot 1 \cdot 8 + \frac{2}{3} \cdot 1 \cdot 9) = 6.4 + 6 = 12.4.$$

The total value of each bidder is additive in the allocated quantities: $V_1 = 2 \cdot 10 + 1 \cdot 6 = 26$ and $V_2 = 1 \cdot 8 + 1 \cdot 9 = 17$. Utilities are

$$u_1 = V_1 - p_1 = 26 - 20 = 6 \geq 0,$$

$$u_2 = V_2 - p_2 = 17 - 12.4 = 4.6 \geq 0.$$

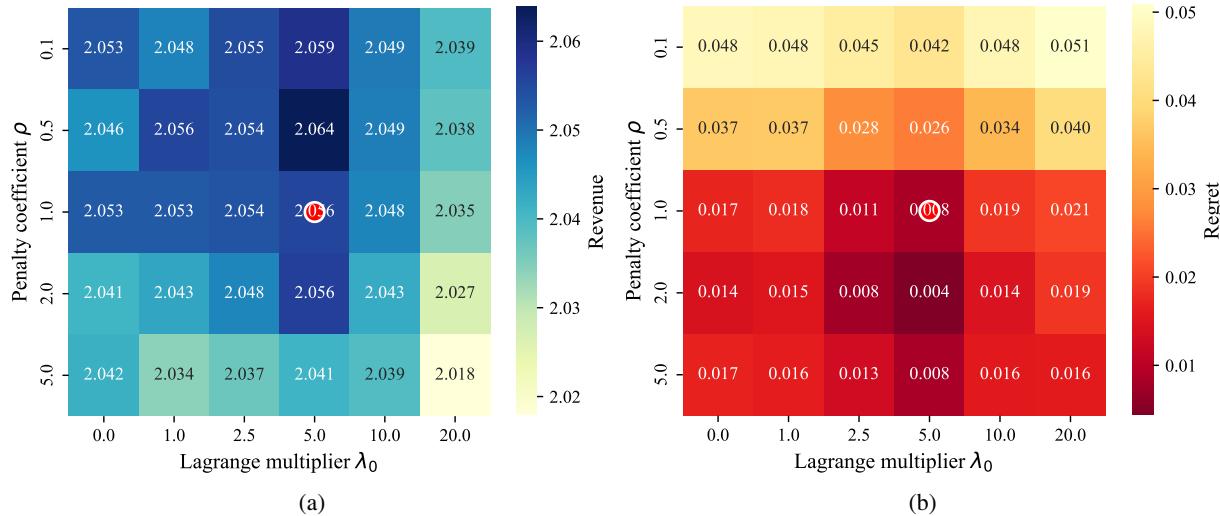


FIGURE 14: Joint sensitivity of performance to the initial Lagrange multiplier and the penalty coefficient under Setting I (2×2). (a) Revenue heatmap over (λ_0, ρ) . (b) Regret heatmap over (λ_0, ρ) .

In this example, IR holds because both bidders achieve non-negative utilities under the specified allocation and payment factors: bidder 1 has $u_1 = 26 - 20 = 6$ and bidder 2 has $u_2 = 17 - 12.4 = 4.6$. SE is satisfied by construction, with $\sum_i d_{iA} = 3 \leq q_A$ and $\sum_i d_{iB} = 2 \leq q_B$, and each allocation respects individual demand bounds $d_{ij} \leq o_{ij}$. Regarding IC, while the full mechanism in our manuscript evaluates IC through regret minimization rather than closed-form dominant strategies, the present instantiation offers clear intuition: with fixed bidder multipliers \tilde{p}_i and rival-indexed scaling factors g_{ij} , increasing a bid b_{ij} scales payments proportionally without improving allocation once demand or capacity is saturated, whereas decreasing b_{ij} risks forfeiting profitable units. Consequently, truthful bidding is weakly stable in this constructed instance, reflecting a second-price-like effect embedded within the payment module $p_i = \tilde{p}_i \sum_{j \in M} g_{ij} d_{ij} b_{ij}$.

REFERENCES

- [1] D. Kaltenecker, R. Al-Maskari, M. Negwer, L. Hoherer, F. Kofler, S. Zhao, M. Todorov, Z. Rong, J. C. Paetzold, B. Wiestler *et al.*, “Virtual reality-empowered deep-learning analysis of brain cells,” *Nature Methods*, vol. 21, no. 7, pp. 1306–1315, 2024.
- [2] O. Aouedi, T.-H. Vu, A. Sacco, D. C. Nguyen, K. Piamrat, G. Marchetto, and Q.-V. Pham, “A survey on intelligent internet of things: Applications, security, privacy, and future directions,” *IEEE communications surveys & tutorials*, 2024.
- [3] L. Kong, X. Xu, J. Ren, W. Zhang, L. Pan, K. Chen, W. T. Ooi, and Z. Liu, “Multi-modal data-efficient 3d scene understanding for autonomous driving,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [4] M. Shahnawaz and M. Kumar, “A comprehensive survey on big data analytics: Characteristics, tools and techniques,” *ACM Computing Surveys*, 2025.
- [5] E. Dritsas and M. Trigka, “A survey on the applications of cloud computing in the industrial internet of things,” *Big data and cognitive computing*, vol. 9, no. 2, p. 44, 2025.
- [6] M. M. Jaber, M. H. Ali, S. K. Abd, A. Alkhayyat, R. Malik *et al.*, “Application of edge computing-based information-centric networking in smart cities,” *Computer Communications*, vol. 211, pp. 46–58, 2023.
- [7] J. Anand and B. Karthikeyan, “Dynamic priority-based task scheduling and adaptive resource allocation algorithms for efficient edge computing in healthcare systems,” *Results in Engineering*, p. 104342, 2025.
- [8] A. A. Al-Bakhrani, M. Li, M. S. Obaidat, and G. A. Amran, “Moalf-uav-mec: Adaptive multi-objective optimization for uav-assisted mobile edge computing in dynamic iot environments,” *IEEE Internet of Things Journal*, 2025.
- [9] J. Tu, L. Yang, and J. Cao, “Distributed machine learning in edge computing: Challenges, solutions and future directions,” *ACM Computing Surveys*, vol. 57, no. 5, pp. 1–37, 2025.
- [10] R. Lu, W. Zhang, Y. Wang, Q. Li, X. Zhong, H. Yang, and D. Wang, “Auction-based cluster federated learning in mobile edge computing systems,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 34, no. 4, pp. 1145–1158, 2023.
- [11] X. Wang, X. Wang, C. Wang, R. Zeng, L. Ma, Q. He, and M. Huang, “Truthful online combinatorial auction-based mechanisms for task offloading in mobile edge computing,” *IEEE Transactions on Mobile Computing*, 2025.
- [12] D. Wu, X. Wang, S. Yan, Q. He, and M. Huang, “Privacy-preserving and truthful auction-based resource allocation mechanisms for task offloading in mobile edge computing,” *Computer Networks*, p. 111079, 2025.
- [13] U. Habiba, S. Maghsudi, and E. Hossain, “A repeated auction model for load-aware dynamic resource allocation in multi-access edge computing,” *IEEE Transactions on Mobile Computing*, 2023.
- [14] Y. Su, W. Fan, Y. Liu, and F. Wu, “A truthful combinatorial auction mechanism towards mobile edge computing in industrial internet of things,” *IEEE Transactions on Cloud Computing*, vol. 11, no. 2, pp. 1678–1691, 2022.
- [15] L. Wu, P. Sun, Z. Wang, X. Pang, J. Hu, H. Chen, and Y. Yang, “An incentive framework for task offloading in edge computing marketplaces under price competition,” *IEEE Transactions on Mobile Computing*, 2025.
- [16] H. Qiu, K. Zhu, N. C. Luong, C. Yi, D. Niyato, and D. I. Kim, “Applications of auction and mechanism design in edge computing: A survey,” *IEEE Transactions on Cognitive Communications and Networking*, vol. 8, no. 2, pp. 1034–1058, 2022.
- [17] W. Sun, J. Liu, Y. Yue, and H. Zhang, “Double auction-based resource allocation for mobile edge computing in industrial internet of things,” *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4692–4701, 2018.
- [18] R. B. Myerson, “Optimal auction design,” *Mathematics of operations research*, vol. 6, no. 1, pp. 58–73, 1981.
- [19] P. Düting, Z. Feng, H. Narasimhan, D. Parkes, and S. S. Ravindranath, “Optimal auctions through deep learning,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 1706–1715.
- [20] K. Kuo, A. Ostuni, E. Horishny, M. J. Curry, S. Dooley, P.-y. Chiang, T. Goldstein, and J. P. Dickerson, “Proportionnet: Balancing fair-

- ness and revenue for auction design with deep learning,” *arXiv preprint arXiv:2010.06398*, 2020.
- [21] N. Peri, M. Curry, S. Dooley, and J. Dickerson, “Preferencenet: Encoding human preferences in auction design with deep learning,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 17532–17542, 2021.
 - [22] A. Stein, A. Schwarzschild, M. Curry, T. Goldstein, and J. Dickerson, “Neural auctions compromise bidder information,” *arXiv preprint arXiv:2303.00116*, 2023.
 - [23] J. Rahme, S. Jelassi, J. Bruna, and S. M. Weinberg, “A permutation-equivariant neural network architecture for auction design,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 6, 2021, pp. 5664–5672.
 - [24] Z. Duan, J. Tang, Y. Yin, Z. Feng, X. Yan, M. Zaheer, and X. Deng, “A context-integrated transformer-based neural network for auction design,” in *International Conference on Machine Learning*. PMLR, 2022, pp. 5609–5626.
 - [25] K. Zhu, Y. Xu, Q. Jun, and D. Niyato, “Revenue-optimal auction for resource allocation in wireless virtualization: A deep learning approach,” *IEEE Transactions on Mobile Computing*, vol. 21, no. 4, pp. 1374–1387, 2020.
 - [26] Z. Duan, H. Sun, Y. Chen, and X. Deng, “A scalable neural network for dsic affine maximizer auction design,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 56169–56185, 2023.
 - [27] M. Liang, J. Zhu, Y. Zhang, X. Li, and M. Zhao, “Intermediary-supervised auction mechanism design with deep neural networks,” *IEEE Transactions on Network Science and Engineering*, 2025.
 - [28] J. Hartford, D. Graham, K. Leyton-Brown, and S. Ravanbakhsh, “Deep models of interactions across sets,” in *International Conference on Machine Learning*. PMLR, 2018, pp. 1909–1918.
 - [29] Z. Niu, G. Zhong, and H. Yu, “A review on the attention mechanism of deep learning,” *Neurocomputing*, vol. 452, pp. 48–62, 2021.
 - [30] B. Edelman, M. Ostrovsky, and M. Schwarz, “Internet advertising and the generalized second-price auction: Selling billions of dollars worth of keywords,” *American economic review*, vol. 97, no. 1, pp. 242–259, 2007.
 - [31] W. Vickrey, “Counterspeculation, auctions, and competitive sealed tenders,” *The Journal of finance*, vol. 16, no. 1, pp. 8–37, 1961.
 - [32] E. H. Clarke, “Multipart pricing of public goods,” *Public choice*, pp. 17–33, 1971.
 - [33] T. Groves, “Incentives in teams,” *Econometrica: Journal of the Econometric Society*, pp. 617–631, 1973.

YUANYUAN ZHANG is currently pursuing a PhD degree at Yangzhou University, with research interests in deep learning and game theory. He has published papers in international journals such as Applied Soft Computing and has also presented at conferences like the China Conference on Knowledge Graph and Semantic Computing. He has participated in several provincial and ministerial research funds.

XINPENG LU received his bachelor’s degree in Software Engineering from Yangzhou University. He is currently with the Advanced Technology Research Institute, University of Science and Technology of China (USTC), Hefei, China, pursuing a master’s degree in Computer Science at USTC. His research interests include reinforcement learning and multi-agent systems. He has published papers in international conferences and journals such as AAMAS, CSCWD, and TCE.

MINGXUAN LIANG is currently pursuing a PhD degree in Yangzhou University, with research interests in deep learning and game theory. He has published papers in international conferences such as International Conference on Computer Science, Engineering and Applications. He has participated in several provincial and ministerial research funds.

JUNWU ZHU is currently a professor at Yangzhou University. He served as a visiting professor at the IMAGO Laboratory at the University of Guelph in Canada in 2013 and 2015. He is a member of the Information System Committee of the Chinese Computer Society, primarily engaged in researching knowledge engineering and game theory. He has presided over a number of National Natural Science Foundation of China and Natural Science Foundation of Jiangsu Province.

YONGLONG ZHANG is currently a professor at Yangzhou University, primarily teaching computing and conducting research in cloud computing, mechanism design, and auctions. He has presided over one provincial and ministerial-level open project, one industry-university-research project in Yangzhou City, participated in four national and provincial-level scientific research funds, and published more than 20 papers at home and abroad.