FL-Market: Trading Private Models in Federated Learning

Shuyuan Zheng*, Yang Cao†✉, Masatoshi Yoshikawa*, Huizhong Li‡, Qiang Yan§
*Kyoto University, †Hokkaido University, ‡WeBank Co., Ltd., §Singapore Management University
Email: *{caryzheng@db.soc., yoshikawa@}i.kyoto-u.ac.jp, †yang@ist.hokudai.ac.jp,
‡wheatli@webank.com, §qiang.yan.2008@smu.edu.sg

***Abstract*—Acquiring a sufficient amount of training data is a significant bottleneck for machine learning (ML) based data analytics. Recently, commoditizing ML models has been proposed as an economical and moderate solution to ML-oriented data acquisition. However, existing model marketplaces assume that the broker can access data owners' private training data, which may not be realistic in practice. In this paper, to promote trustworthy data acquisition for ML tasks, we propose FL-Market, a *locally* private model marketplace that protects privacy against not only model buyers but also an untrusted broker. FL-Market decouples ML from the need to centrally gather training data on the broker's side using *federated learning*, a privacy-preserving ML paradigm in which data owners collaboratively train an ML model by uploading local gradients (to be aggregated into a global gradient for model updating). Then, FL-Market enables data owners to locally perturb their gradients by *local differential privacy* and thus further prevents privacy risks. To drive FL-Market, we propose a deep learning-empowered auction mechanism for intelligently deciding the local gradients' perturbation levels and an optimal aggregation mechanism for aggregating the perturbed gradients. Our auction and aggregation mechanisms can jointly maximize the global gradient's accuracy, which optimizes model buyers' utility. Our experiments verify the effectiveness of the proposed mechanisms.**

***Index Terms*—data trading, incentive mechanism, federated learning, local differential privacy**

I. Introduction

Machine learning (ML) based data analytics has demonstrated great success in many domains. Acquiring a sufficient amount of private data to train ML models usually needs considerable expenses, especially as data owners are becoming increasingly aware of the value of their data and the severe risks from uncontrolled data usage after sharing the data. Consequently, recent efforts have proposed *model marketplaces* [1]–[5] where a data broker commercializes data owners' private data in the form of ML models to facilitate ML-oriented data acquisition. Since model buyers do not contact training data directly, this category of business models can relieve data owners' concerns about losing control over their data and thus incentivize data sharing to some extent.

However, data owners still face notable privacy risks in the existing model marketplaces, which may make them hesitate to contribute data. Although some works (e.g., [1], [4], [5]) reduce privacy leakage to model buyers by injecting random noise into ML models using central differential privacy (CDP) [6], existing works assume that the broker is trusted and authorized to access and control the raw data. This assumption is unrealistic, considering that many giant companies have been involved in user data breaches or privacy scandals. Therefore, we demand a model marketplace that protects privacy against not only model buyers but also its broker.

Federated learning (FL) [7] has emerged as a promising paradigm for privacy-preserving ML. Unlike traditional ML that requires training data to be stored on a centralized server (e.g., a broker in a model marketplace), FL enables the clients (i.e., data owners) to collaboratively train a model by uploading local updates (e.g., gradients) and, meanwhile, to keep their own training data on the local sides. Since FL decouples ML from the need to centrally gather training data, it can largely restrict an untrusted server's ability to acquire private information. Even though the local gradients trained on the raw data can be sensitive [8], many works [9]–[13] suggest that *local differential privacy* (LDP) [14] can be combined with FL to perturb the gradients on the local sides and thus protect privacy.

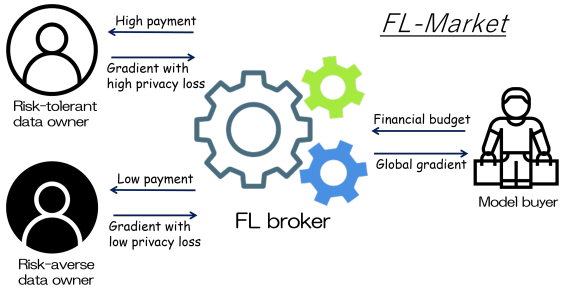

Fig. 1: FL-Market allows data owners to control the perturbation level of their gradients in each round of FL training. Those data owners who contribute more accurate gradients (i.e., with less noise) will receive higher payments.

In this paper, for the first time, we propose a locally private model marketplace empowered by FL and LDP, called *FL-Market* (Federated Learning Based Locally Private Model Market), to promote trustworthy data acquisition for ML-based data analytics. Figure 1 depicts the three parties in FL-Market: data owners, model buyers, and an FL broker. The FL broker coordinates FL-based model training and trading between data owners and model buyers. A model buyer attempts to purchase ML models with a financial budget. Data owners do not sell their raw data; instead, they sell locally private gradients perturbed by LDP in the training process coordinated by the FL broker. The perturbation level is controlled by a privacy parameter ϵ, which LDP formally defines as a metric of privacy loss.

978-1-6654-8045-1/22/$31.00 ©2022 IEEE

To incentivize contribution, we follow seminal differentially private data marketplaces [15]–[20] to employ an auction-based method for pricing gradients. Concretely, we allow each owner to report (bid) her valuation of privacy loss, named *privacy valuation*, and report the maximum tolerable privacy loss, called *privacy budget*. Then, the broker uses an *auction mechanism* to decide each owner's privacy parameter and compensate for the corresponding privacy loss according to her privacy valuation. The auction should guarantee *truthfulness*, which means each data owner (i.e., a bidder) will never obtain a higher utility by reporting an untruthful privacy valuation and budget. Finally, the perturbed local gradients are aggregated into a global gradient by an *aggregation mechanism* to update the buyer's model.

Building this model marketplace calls for an elaborate mechanism design that enables the auction and aggregation mechanisms to jointly optimize the global gradient's utility. First, in FL-Market, the broker has to aggregate the locally private gradients considering their various accuracy levels. Consequently, the aggregation mechanism should factor in the privacy losses decided by the auction mechanism when making a decision. Second, the auction mechanism should properly purchase local gradients to maximize the aggregated gradient's utility, which implies that the aggregation decision feeds back into the auction decision. However, the aggregation mechanism may fail to provide an analytical solution. In this case, the utility-maximizing objective of our auction problem also cannot be expressed in an analytic form, which makes it extremely challenging to characterize and design an optimal truthful mechanism. In a nutshell, the need for joint optimization dramatically increases the complexity of optimal mechanism design.

Our main contributions are threefold.

- We design a novel privacy-preserving model trading framework, *FL-Market*, for acquiring locally private ML models via FL (Section III). In FL-Market, data owners maintain control of their raw data by FL and enjoy the desired level of privacy against both the broker and model buyers using LDP. To the best of our knowledge, FL-Market is the first *locally* private model marketplace. On the other end, we formulate optimization problems for designing the auction and aggregation mechanisms with the objective of maximizing the global gradient's accuracy, which optimizes model buyers' utility.
- We propose an optimal aggregation mechanism *OptAggr* for FL with personalized LDP parameters (Section IV). The conventional practice of FL aggregates gradients with weights proportional to clients' data sizes (i.e., all samples are uniformly weighted), which may not be optimal when the gradients are perturbed to different extents. We transform the problem of designing an optimal aggregation mechanism under personalized privacy losses into an equivalent *quadratic programming problem*. We prove that the equivalent problem is convex and thus can be solved by off-the-shelf optimizers. Supported by the optimizers, OptAggr decides the optimal way to aggregate the gradients.
- We propose a novel auction mechanism, *DM-RegretNet*, to incentivize data owners to contribute accurate gradients (Section V). Concretely, to design an optimal mechanism that jointly optimizes the gradient's utility with the aggregation mechanism, we seek support from RegretNet, the state-of-the-art deep learning-empowered automated mechanism design technique [21]. However, RegretNet always generates *randomized* allocation results for auction items (i.e., the privacy losses in our case), which makes it tough to maximize the global gradient's accuracy. On the contrary, DM-RegretNet (Deterministic Multi-Unit RegretNet) yields *deterministic* auction decisions jointly with OptAggr and thus can significantly improve the global gradient's utility. Our extensive experiments demonstrate that DM-RegretNet can achieve better model accuracy and approximate the truthfulness constraint more closely than RegretNet.

II. Preliminary

Federated learning: FL is a privacy-preserving framework for collaborative ML. In a typical FL architecture, n data owners $\{1, ..., n\}$ collaboratively train an ML model $h_w(\cdot)$ using their datasets $\{D_1, ..., D_n\}$ under the coordination of an FL server (e.g., the FL broker in FL-Market), where w is a set of model parameters. The training process consists of multiple training rounds $1, ..., R$. We show a training round $r \in [R]$ of the widely-used FedSGD algorithm [7] as follows.

1) **Model broadcasting**: The server broadcasts model parameters w^r with a loss function $l(\cdot)$.
2) **Local training**: Each data owner i computes a local gradient g_i using her local dataset $D_i = [r_{i,j}]_{j \in [d_i]}$ consisting of d_i records. The gradient g_i is the mean gradient of the records, i.e., $g_i = \mathbb{E}_{r \in D_i}[\nabla l(w^r; r)]$.
3) **Gradients aggregation**: The server collects all the local gradients and aggregates them into a global gradient g^* by averaging, i.e., $g^* = \sum_{i=1}^{n} \frac{d_i}{d_1 + ... + d_n} g_i$ where d_i denotes the size of D_i.
4) **Model updating**: The server updates the model parameters w^r by the global gradient, i.e., $w^{r+1} = w^r - \eta \cdot g^*$ where $\eta \in R^+$ is a learning rate.

In addition, gradient clipping is a widely used method for avoiding the exploding gradient problem [22] where unacceptably large gradients make the training process unstable. In this paper, we adopt the gradient clipping method $clip$ [23] that rescales a gradient g_i if its norm cannot be covered by a threshold L, i.e., $clip(g_i, L) = g_i \cdot \min(1, \frac{L}{||g_i||_1})$. To reduce notational overload, we let each g_i denote the clipped version in the rest of this paper, i.e.,

$$g_i = \mathbb{E}_{r \in D_i}[\nabla l(w^r; r)] \cdot \min(1, \frac{L}{||\mathbb{E}_{r \in D_i}[\nabla l(w^r; r)]||_1}) \quad (1)$$

Local differential privacy: LDP [14] is a de facto data privacy definition. In FL, even if data owners maintain their datasets on the local sides, their private information still can be inferred from the uploaded gradients by the server [8]. To prevent privacy leakage, data owners can use an LDP perturbation mechanism $\mathcal{M}$, such as the Laplace mechanism

[6], to perturb the gradients before uploading them, which ensures that any change to the mechanism's input does not significantly affect the output. The protection level of LDP for owner i is parameterized by ϵ_i, which also quantifies her privacy loss. A smaller ϵ_i corresponds to a higher protection level and a more randomized perturbation. We let $\mathcal{M}_{\epsilon_i}$ denote a perturbation mechanism that satisfies ϵ_i-LDP. Note that if we perturb a gradient g_i by $\mathcal{M}_{\epsilon_i}$, releasing the perturbed gradient also satisfies ϵ_i-LDP for each record $r \in D_i$.

Definition 1 (ϵ_i-Local Differential Privacy [14])**.** *Given a privacy loss $\epsilon_i \geq 0$, a randomized mechanism $\mathcal{M}$ satisfies ϵ_i-LDP if for any two inputs $x, x' \in Domain(\mathcal{M})$ and any output $o \in Range(\mathcal{M})$, we have:*

$$Pr[\mathcal{M}(x) = o] \leq \exp(\epsilon_i) \cdot Pr[\mathcal{M}(x') = o]$$

III. FL-Market Framework

A. Market Setup

Participants: As shown in Figure 2, there are three parties in FL-Market: data owners, model buyers, and an FL broker. A *model buyer* enters FL-Market to purchase a global gradient with a financial budget B at each FL training round r to train her target model h_{w^r}. We assume that the buyer already knows that data owners' data attributes meet her needs. *Data owners* $\mathcal{N} = \{1, ..., n\}$ possess local datasets $D = \{D_1, ..., D_n\}$ that can be used to compute local gradients $g_1, ..., g_n$ for training h_{w^r}. To prevent privacy leakage against the FL broker and model buyers, each owner i perturbs her local gradient g_i using a perturbation mechanism $\mathcal{M}_{\epsilon_i}$ that satisfies ϵ_i-LDP. The *broker* mediates between the model buyer and data owners in the FL process: it arranges the training tasks among data owners, collects their perturbed local gradients, and aggregates them into a perturbed global gradient for the buyer. In addition, the broker sets the payments $p_1, ..., p_n$ to data owners within the buyer's budget B.

Privacy valuation: Inspired by [15], [16], FL-Market requires data owners to report their privacy valuations to price perturbed gradients. Concretely, each owner i has a *valuation function* $v_i(\epsilon_i, d_i)$ that reflects her valuation of her privacy loss ϵ_i for her d_i-sized dataset: she will accept a privacy loss ϵ_i for d_i records if she obtains a payment $p_i \geq v_i(\epsilon_i, d_i)$. However, in [15], [16], data owners cannot set the upper bounds of their privacy losses. To provide better privacy protection as an incentive, we follow Zheng et al. [24] to allow each owner i to set a *privacy budget* $\bar{\epsilon}_i$ that denotes the maximum tolerable privacy loss. In practice, the broker can provide some instructions to help data owners decide privacy valuations and budgets, e.g., questionnaires for figuring out privacy preferences, typical choices for different preferences, and some analysis of historical transaction data.

Threat model: We assume that all the participants are honest-but-curious, which means they will not deviate from the protocol but will attempt to learn information from received messages. Note that in an auction, reporting a *fake bid* that does not represent the bidder's real preference is not a malicious behavior that violates the protocol since the auction allows bidders to submit arbitrary bids.

B. Trading Framework

We depict the trading framework in Fig. 2 and Alg. 1. Initially, a model buyer enters FL-Market and specifies a target model $h_w(\cdot)$ with a loss function $l(\cdot)$ for FL. Then, in each FL training round r, the buyer purchases a global gradient for model updating by the following steps:

Algorithm 1 Trading Framework of FL-Market

1: A buyer specifies a model $h_w(\cdot)$ with a loss function $l(\cdot)$.
2: **for** each FL training round r **do**
3: The buyer announces an auction with a financial budget B and model parameters w^r.
4: Data owners report their bids $\boldsymbol{b}' = (b'_1, ..., b'_n)$.
5: The broker runs $\mathsf{Auc}(\boldsymbol{b}', B) \rightarrow \boldsymbol{\epsilon}, \boldsymbol{p}$.
6: The broker broadcasts w^r and data owners compute perturbed local gradients $\tilde{g}_1, ..., \tilde{g}_n$.
7: The broker runs $\mathsf{Aggr}(\boldsymbol{\epsilon}, \boldsymbol{d}) \rightarrow \boldsymbol{\lambda}$.
8: The broker delivers a global gradient $\tilde{g}_{\boldsymbol{\lambda}} = \sum_{i=1}^{n} \lambda_i \cdot \tilde{g}_i$ to the buyer for model updating.

1) **Auction announcement**: The buyer asks the FL broker to announce a procurement auction (where bidders are sellers) for purchasing gradients, specifying a financial budget B and model parameters w^r.
2) **Bidding**: Data owners report their bids $b'_1, ..., b'_n$ in the auction. We assume that each owner i has a *real bid* $b_i = (v_i, \bar{\epsilon}_i, \bar{d}_i)$ in mind consisting of her valuation function v_i, the maximum privacy budget $\bar{\epsilon}_i$, and the maximum size of her dataset $\bar{d}_i$. Then, each i reports to the broker a valuation function v'_i, a privacy budget $\bar{\epsilon}'_i$ and a data size d_i as a *reported bid* $b'_i = (v'_i, \bar{\epsilon}'_i, d_i)$. If the reported bid b'_i is *truthful*, then $b'_i = b_i$; otherwise, it is a *fake bid*, i.e., $b'_i \neq b_i$. We simplify "reported bid" as "bid" and denote the collection of all the bids as a bid profile $\boldsymbol{b}' = [b'_1, ..., b'_n]$.
3) **Auction decision**: The broker runs an *auction mechanism* Auc to decide data owners' privacy losses and payments. Formally, an auction mechanism given a bid profile $\boldsymbol{b}'$ and a financial budget B yields an allocation of privacy losses $\boldsymbol{\epsilon} = [\epsilon_1, ..., \epsilon_n]$ and payments $\boldsymbol{p} = [p_1, ..., p_n]$.
4) **Local gradient computing**: Given model parameters w^r, each data owner i computes and submits a noisy gradient $\tilde{g}_i = \mathcal{M}_{\epsilon_i}(g_i)$ to the broker.
5) **Gradients aggregation and model delivery**: The FL broker runs an *aggregation mechanism* Aggr to aggregate those noisy gradients into a perturbed global gradient $\tilde{g}_{\boldsymbol{\lambda}}$. Finally, the broker returns $\tilde{g}_{\boldsymbol{\lambda}}$ to the model buyer.

Gradients aggregation: In step (5), the broker needs a "good" strategy to aggregate the collected noisy gradients. To study the optimality of the aggregation mechanism in our setting, we generalize the problem as follows. Formally, given data owners' perturbed gradients $\tilde{g}_1, ..., \tilde{g}_n$, the broker sets the

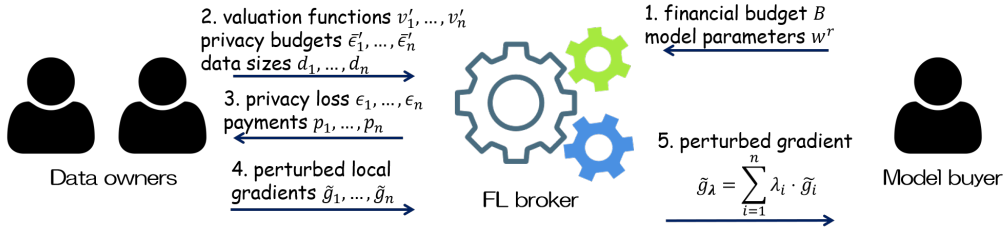

Fig. 2: FL-Market Trading Framework.

aggregation weights $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_n]$ with $\sum_{i=1}^{n} \lambda_i = 1, \lambda_i \geq 0, \forall i$ and then computes the perturbed global gradient as:

$$\tilde{g}_{\boldsymbol{\lambda}} = \sum_{i=1}^{n} \lambda_i \cdot \tilde{g}_i. \tag{2}$$

We note that Equation (2) is a generalization of the *weighted aggregation* [7], [25] in the literature. Then, we attempt to design an optimal aggregation mechanism under personalized privacy losses. Specifically, we define the aggregation mechanism as a function $\mathsf{Aggr} : R^{2n} \to R^n$ that given privacy losses $\boldsymbol{\epsilon} = [\epsilon_1, ..., \epsilon_n]$ and data sizes $\boldsymbol{d} = [d_1, ..., d_n]$ outputs aggregation weights $\boldsymbol{\lambda} = [\lambda_1, ..., \lambda_n]$ for weighted aggregation.

C. Mechanism Design

In this section, we formulate the problems of designing the auction mechanism Auc and aggregation mechanism Aggr (Lines 5 and 7 in Alg. 1, respectively) to instantiate the trading protocol of FL-Market. The mechanism design should achieve the following two goals: (1) to provide *utility-optimal* global gradients and (2) to prevent *untruthful* privacy valuations.

Aggregation mechanism: The aggregation mechanism should optimally aggregate perturbed local gradients to provide highly usable global gradients for model buyers. Concretely, given local gradients $\tilde{g}_1, ..., \tilde{g}_n$ with privacy losses $\boldsymbol{\epsilon}$ and data sizes $\boldsymbol{d}$, Aggr should yield optimal aggregation weights that minimize the error of the global gradient:

$$\min_{\boldsymbol{\lambda}} err(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}, \boldsymbol{d}) = ||\tilde{g}_{\boldsymbol{\lambda}} - g^*||_2 = ||\sum_{i=1}^{n} \lambda_i \cdot \mathcal{M}_{\epsilon_i}(g_i) - g^*||_2$$

where $g^* = \sum_{i=1}^{n} \frac{d_i}{\sum_{j=1}^{n} d_j} g_i$ is the raw global gradient without any perturbation. The lower the error $err(\tilde{g}_{\boldsymbol{\lambda}})$ is, the smaller the difference between $\tilde{g}_{\boldsymbol{\lambda}}$ and g^*, which also implies that the buyer will obtain a more accurate global model.

However, the broker cannot calculate the ground-truth error $err(\tilde{g}_{\boldsymbol{\lambda}})$ under LDP without the access to $g_1, .., g_n$. Hence, we turn to the error bound $ERR(\tilde{g}_{\boldsymbol{\lambda}})$ and design the aggregation mechanism by solving the following problem:

Problem 1 (Error Bound-Minimizing Aggregation)**.**

$$\min_{\boldsymbol{\lambda} = \mathsf{Aggr}(\boldsymbol{\epsilon}, \boldsymbol{d})} ERR(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}, \boldsymbol{d}) = \sup_{g_1, ..., g_n} err(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}, \boldsymbol{d})$$

$$\textit{S.t.: } \forall i, \lambda_i \in [0, 1], \textit{ and } \sum_{i=1}^{n} \lambda_i = 1$$

Auction mechanism: Solving Problem 1 alone is still insufficient to determine a utility-optimal global gradient since the utility is also affected by the privacy losses purchased for perturbing the local gradients. That is, the auction mechanism Auc should take the aggregation mechanism into account to *jointly* optimize the (expected) error bound of the global gradient over all possible bid profiles and financial budgets:

$$\min_{\boldsymbol{\epsilon}, \boldsymbol{p} = \mathsf{Auc}(\boldsymbol{b}', B)} \mathbb{E}_{(\boldsymbol{b}', B)}[ERR(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\lambda} = \mathsf{Aggr}(\boldsymbol{\epsilon}, \boldsymbol{d}))]$$

$$= \mathbb{E}_{(\boldsymbol{b}', B)}[\sup_{g_1, ..., g_n} ||\sum_{i=1}^{n} \lambda_i \cdot \mathcal{M}_{\epsilon_i}(g_i) - g^*||_2]$$

$$= \mathbb{E}_{(\boldsymbol{b}', B)}[\sup_{g_1, ..., g_n} ||\mathsf{Aggr}(\boldsymbol{\epsilon}, \boldsymbol{d}) \cdot [\mathcal{M}_{\epsilon_1}(g_1), ..., \mathcal{M}_{\epsilon_n}(g_n)] - g^*||_2]$$

Then, Auc needs to determine appropriate auction results that prevent untruthful privacy valuations. Concretely, by trading a global gradient, each data owner i obtains a utility

$$u_i(b'_i; \boldsymbol{b}'_{-i}, B) = \begin{cases} p_i - v_i(\epsilon_i, d_i), & \epsilon_i \leq \bar{\epsilon}_i, d_i \leq \bar{d}_i \\ -\infty, & \text{otherwise} \end{cases}$$

where $\boldsymbol{b}'_{-i} = (b'_1, ..., b'_{i-1}, b'_{i+1}, ..., b'_n)$ denotes the other bidders' bids. Then, Auc should ensure the following incentives:

- Truthfulness: With the other bidders' bids $\boldsymbol{b}'_{-i}$ fixed, each bidder i never obtains a higher utility by reporting a fake bid $b'_i \neq b_i$, i.e., $\forall i, \forall b'_i, \forall B, u_i(b'_i; \boldsymbol{b}'_{-i}, B) \leq u_i(b_i; \boldsymbol{b}'_{-i}, B)$.
- Individual rationality (IR): Each bidder i never obtains a negative utility, i.e., $u_i(b'_i) \geq 0, \forall b'_i, \forall i$.
- Budget feasibility (BF): The payments should be within the financial budget, i.e., $\sum_i p_i \leq B$.

Therefore, we can design the auction mechanism by solving the following problem.

Problem 2 (Budget-Limited Multi-Unit Multi-Item Procurement Auction)**.**

$$\min_{\boldsymbol{\epsilon}, \boldsymbol{p} = \mathit{Auc}(\boldsymbol{b}', B)} \mathbb{E}_{(\boldsymbol{b}', B)}[ERR(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\lambda} = \mathit{Aggr}(\boldsymbol{\epsilon}, \boldsymbol{d}))]$$

$$\textit{S.t.: } \forall i, \epsilon_i \in [0, \bar{\epsilon}'_i], \textit{ truthfulness, IR, and BF.}$$

Problem 2 is a budget-limited multi-unit multi-item procurement auction problem [26] because (1) each data owner's privacy loss ϵ_i can be seen as a divisible item for procurement with $\bar{\epsilon}'_i$ units available, and (2) the buyer purchases privacy losses under her financial budget B. To the best of our knowledge, such a problem has yet to be generally solved in the literature. Moreover, we have to involve the aggregation

mechanism in minimizing the global gradient's error bound, which increases the complexity of optimal mechanism design. Concretely, the privacy losses affect the aggregation weights in Problem 1, but the latter also feeds back into the former in Problem 2, which calls for joint optimization. By solving this problem, we can obtain an auction mechanism that maximizes the global gradient's utility jointly with Aggr.

Computational efficiency: We additionally require that the auction and aggregation mechanisms (designed by solving Problems 1 and 2) should finish in polynomial time, which ensures the efficiency of FL-Market. Note that we design the mechanisms offline before executing Algorithm 1 rather than during each FL training round therein.

IV. Aggregation Mechanism: OptAggr

In this section, we propose an error-optimal aggregation mechanism *OptAggr* by solving a *convex quadratic programming problem* that we prove is equivalent to Problem 1.

Error bound decomposition: It is well known that the MSE error of a random variable consists of its variance and squared bias. Let σ_i denote the variance of the local gradient $\tilde{g}_i$, and let $W_i = \frac{d_i}{\sum_{j\in[n]} d_j}, \forall i$. We can decompose the error $err(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}, \boldsymbol{d})$ as $err(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}, \boldsymbol{d}) = var(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}) + bias^2(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}, \boldsymbol{d})$ where

$$var(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}) = var(\sum_{i=1}^{n} \lambda_i \tilde{g}_i; \boldsymbol{\epsilon}) = \sum_{i=1}^{n} (\lambda_i)^2 \sigma_i,$$

$$bias(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}, \boldsymbol{d}) = ||E[\tilde{g}_{\boldsymbol{\lambda}}] - E[g^*]||_2 = ||\sum_{i=1}^{n} \lambda_i g_i - \sum_{i=1}^{n} W_i g_i||_2$$

$$= ||\sum_{i=1}^{n} (\lambda_i - W_i) g_i||_2 \leq \sum_{i=1}^{n} |\lambda_i - W_i| \cdot ||g_i||_2$$

$$= \sum_{i=1}^{n} |\lambda_i - W_i| \cdot ||\mathbb{E}_{r\in D_i}[\nabla l(w^r; r)] \cdot \min(1, \frac{L}{||\mathbb{E}_{r\in D_i}[\nabla l(w^r; r)]||_1})||_2$$

Because $\sup_{g_1,...,g_n} bias(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}, \boldsymbol{d}) = \sum_{i=1}^{n} |\lambda_i - W_i| L$, the objective function of Problem 1 is equal to

$$\min_{\boldsymbol{\lambda}=\mathsf{Aggr}(\boldsymbol{\epsilon},\boldsymbol{d})} ERR(\tilde{g}_{\boldsymbol{\lambda}}; \boldsymbol{\epsilon}, \boldsymbol{d}) = \sum_{i=1}^{n} (\lambda_i)^2 \sigma_i + (\sum_{i=1}^{n} |\lambda_i - W_i| L)^2.$$

Problem transformation: We further transform Problem 1 into a convex quadratic programming problem. First, to minimize the error bound, any data owner i with $\epsilon_i = 0$ must be allocated a zero-valued weight $\lambda_i = 0$ by an optimal solver because its gradient $\tilde{g}_i$ has an infinite variance σ_i. For simplicity, we assume that only the first $k \leq n$ data owners have positive privacy losses without loss of generality. Then, we let $\boldsymbol{x} = [\lambda_1, ..., \lambda_k]$ and replace the terms $|\lambda_i - W_i|, \forall i \in [k]$ with auxiliary variables $\boldsymbol{y} = [y_1, ..., y_k]$ with the constraints $y_i \geq -(\lambda_i - W_i), y_i \geq \lambda_i - W_i, \forall i \in [k]$. Consequently, we have the following quadratic programming problem [27].

Problem 3 (Equivalent problem of Problem 1)**.**

$$\min_{\boldsymbol{x},\boldsymbol{y}} \frac{1}{2} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}^T \begin{bmatrix} Diag([\sigma_1, ..., \sigma_k]) & 0 \\ 0 & Uni(L^2)_{k\times k} \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}$$

$$\textit{S.t.:} \begin{bmatrix} Uni(1)_{k\times 1} \\ Uni(0)_{k\times 1} \end{bmatrix}^T \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} = 1, \begin{bmatrix} \boldsymbol{I}_k & -\boldsymbol{I}_k \\ -\boldsymbol{I}_k & -\boldsymbol{I}_k \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix} \leq \begin{bmatrix} \boldsymbol{W} \\ -\boldsymbol{W} \end{bmatrix}$$

Algorithm 2 Aggregation Mech.: OptAggr

Input: privacy losses $\epsilon_1, ..., \epsilon_n$, data sizes $d_1, ..., d_n$
Output: aggregation weights $\lambda_1, ..., \lambda_n$
1: **return** $\lambda_i = W_i, \forall i$ if $\epsilon_i = 0, \forall i$
2: For each data owner i with $\epsilon_i = 0$, let $\lambda_i = 0$
3: For each data owner j with $\epsilon_j > 0$, calculate the variance σ_j; then compute $\lambda_j, \forall j$ using an optimizer that solves Problem 3.
4: **return** $\lambda_1, ..., \lambda_n$

where $\boldsymbol{I}_k$ is a $k \times k$ identity matrix, $Uni(a)_{m\times n}$ is an $m \times n$ matrix where all the elements are equal to $a \in R$, $Diag([\sigma_1, ..., \sigma_k])$ is a $k \times k$ diagonal matrix with $Diag([\sigma_1, ..., \sigma_k])[i][i] = \sigma_i, \forall i \in [k]$, and $\boldsymbol{W} = [W_1, ..., W_k]$.

Because Problem 3 is a convex quadratic programming problem, it can be well solved by many existing solvers in polynomial time, e.g., the SCS solver [28] to be used in our experiments. Note that there is no existing analytical solution to Problem 3 to the best of our knowledge. Hence, we propose the OptAggr mechanism that (1) allocates zero-valued aggregation weights to those data owners with zero-valued privacy losses and (2) then computes other data owners' aggregation weights by solving Problem 3 with a polynomial-time optimizer, as depicted in Algorithm 2.

Proposition 1. *Problem 3 is a convex quadratic programming problem and is equivalent to Problem 1.*

Proof. Let $\mathcal{Q} = \begin{bmatrix} Diag([\sigma_1, ..., \sigma_k]) & 0 \\ 0 & Uni(L^2)_{k\times k} \end{bmatrix}$ and $A = \begin{bmatrix} Diag([\sqrt{\sigma_1}, ..., \sqrt{\sigma_k}]) & 0 \\ 0 & Uni(\frac{L}{\sqrt{k}})_{k\times k} \end{bmatrix}$. Because $\mathcal{Q} = A^T A$, $\mathcal{Q}$ is a positive semidefinite matrix. Therefore, Problem 3 is a convex quadratic programming problem.

For each y_i, a solver for Problem 3 will find the lowest value of y_i as possible. Therefore, if $\lambda_i - W_i \geq 0$, the constraint $y_i \geq \lambda_i - W_i$ is equivalent to $y_i = \lambda_i - W_i$ and implies $y_i \geq -(\lambda_i - W_i)$; if $\lambda_i - W_i \leq 0$, the constraint $y_i \geq -(\lambda_i - W_i)$ is equivalent to $y_i = -(\lambda_i - W_i)$ and implies $y_i \geq \lambda_i - W_i$. Therefore, the constraints $y_i \geq \lambda_i - W_i$ and $y_i \geq -(\lambda_i - W_i)$ are equivalent to $y_i = |\lambda_i - W_i|$. Therefore, we conclude that Problem 3 is equivalent to Problem 1. □

V. Auction Mechanism: DM-RegretNet

In this section, we design a truthful mechanism that maximizes the global gradient's utility jointly with the OptAggr mechanism. Since OptAggr does not provide an analytical solution to Problem 3, the objective function also cannot be expressed in an analytic form, which makes it extremely difficult to characterize and design an optimal truthful mechanism. To design a truthful mechanism that optimizes the nonanalytical objective, we turn to an automated mechanism design approach that achieves an auction objective by ML. We also propose a traditional auction mechanism in Appendix B.

RegretNet: We seek support from RegretNet [21], the state-of-the-art automated mechanism design framework for multi-item auctions. As depicted in Figure 5, RegretNet consists of two deep learning networks: an allocation network and a payment network. Both the networks take as input data owners'

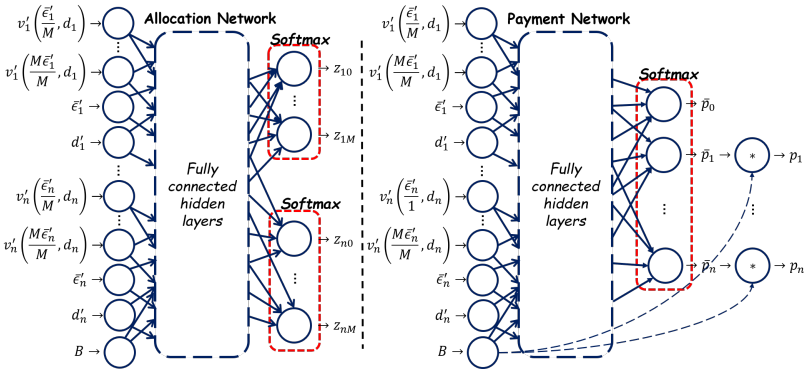

Fig. 3: M-RegretNet.

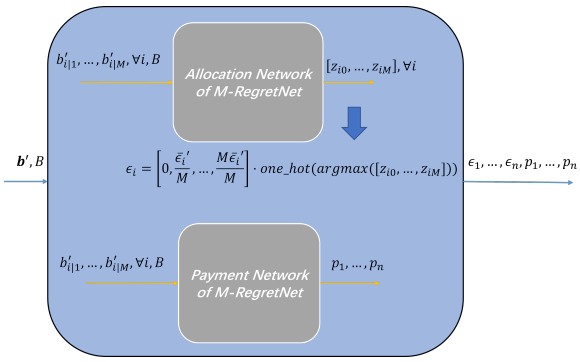

Fig. 4: DM-RegretNet.

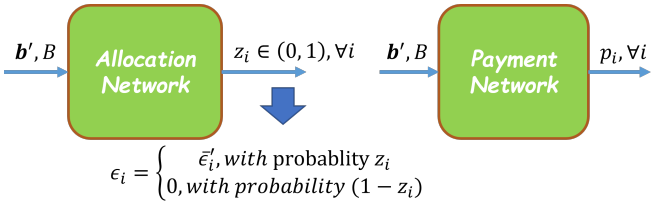

Fig. 5: RegretNet.

bid profile $\boldsymbol{b}'$ and the buyer's financial budget B but output allocation probabilities $z_i \in (0,1), \forall i$ and payments $p_1, ..., p_n$, respectively. Therefore, RegretNet is a randomized auction mechanism: the allocation result of each data owner's privacy loss is a binary random variable ϵ_i with $Pr[\epsilon_i = \bar{\epsilon}_i'] = z_i$ and $Pr[\epsilon_i = 0] = 1 - z_i$.[1] Then, the truthfulness constraint is approximately guaranteed by model training: the violation degree of truthfulness is quantified as a regret penalty in the training objective function to be minimized.

Problems with RegretNet: RegretNet may perform poorly in our auction problem. First, RegretNet can only auction single-unit items and output binary auction results. That is, under RegretNet, the allocation result of each data owner i's privacy loss (i.e., the item ϵ_i) is either to purchase the whole unit (i.e., $\epsilon_i = \bar{\epsilon}_i'$) or not to purchase any privacy loss (i.e., $\epsilon_i = 0$). However, we should support trading a portion of the privacy budget $\bar{\epsilon}_i'$ to flexibly optimize the global gradient's utility. Second, some extra variance from the randomness of the allocation results by RegretNet might be introduced into the perturbed local gradients. Third, also because of the allocation randomness, RegretNet cannot treat the (expected) error bound minimization function as the objective function for model training. Intuitively, RegretNet always allocates zero-valued privacy losses for all data owners with probability $Pr[\epsilon_1 = ... = \epsilon_n = 0] = \prod_{i=1}^{n}(1 - z_i)$, which means that the expected error bound of the perturbed global gradient $\tilde{g}_{\boldsymbol{\lambda}}$ remains infinite and cannot be minimized.

M-RegretNet: To solve the first problem with RegretNet, we extend the allocation network of RegretNet and propose *M-RegretNet* (Multi-Unit RegretNet). As shown in Figure 3, like RegretNet, M-RegretNet has an allocation (payment) network with multiple fully connected hidden layers, each of which has multiple hidden nodes with *tanh* activations. However, it does not take the reported bids as input. Instead, for each data owner i, we transform her reported bid $b_i' = (v_i', \bar{\epsilon}_i', d_i)$ into M sub-bids $b_{i|1}', ..., b_{i|M}'$ and then input them into M-RegretNet, where $b_{i|m}' = (v_i'(\frac{m \cdot \bar{\epsilon}_i'}{M}, d_i), \frac{m \cdot \bar{\epsilon}_i'}{M}, d_i), \forall m \in [M]$. Regarding the payment network, it first generates *budget fractions* $\bar{p}_0, ..., \bar{p}_n$ and then output payments $p_i = \bar{p}_i \cdot B, \forall i \in [n]$. Because the budget fractions are output by a *softmax* activation function, the sum of the payments $\sum_{i \in [n]} p_i$ never exceeds the financial budget B, which ensures BF. Then, the allocation network outputs $\boldsymbol{z}_i = [z_{i0}, z_{i1}, ..., z_{iM}]$ for each data owner i, where z_{i0} denotes the probability of data owner i losing the auction, and z_{im} is the probability of data owner i winning with her sub-bid $b_{i|m}'$. Since each owner i should win with at most one sub-bid, we apply softmax activation functions to ensure that $\sum_{m=0}^{M} z_{im} = 1, \forall i$. Therefore, the allocation result for each data owner i is a random variable ϵ_i with $M+1$ possible values, i.e., $Pr[\epsilon_i = \frac{m \cdot \bar{\epsilon}_i'}{M}] = z_{im}, \forall m \in \{0, 1, ..., M\}$. When $M = 1$, M-RegretNet reduces to a budget-feasible version of RegretNet; when $M \geq 2$, it enables the buyer to only purchase a part of each data owner's privacy budget. In addition, when M increases, it becomes easier for M-RegretNet to approximate the truthfulness and IR guarantees since it has more possible values to allocate as privacy losses.

DM-RegretNet: To address the second and third problems with RegretNet, we further propose *DM-RegretNet* that outputs deterministic allocation results. DM-RegretNet deploys M-RegretNet as a module to determine allocation probabilities $\boldsymbol{z}_1, ..., \boldsymbol{z}_n$ and payments $p_1, ..., p_n$. Then, it realizes deterministic allocation results by processing the vector of allocation probabilities $\boldsymbol{z}_i = [z_{i0}, ..., z_{iM}]$ into a one-hot vector; by such a process, there is only one one-valued allocation probability for each data owner i, and thus each privacy loss ϵ_i is deterministic. Formally, it is

$$\epsilon_i = [0, \frac{1 \cdot \bar{\epsilon}_i'}{M}, ..., \frac{M \cdot \bar{\epsilon}_i'}{M}] \cdot one\_hot(argmax(\boldsymbol{z}_i)) \quad (3)$$

where $one\_hot(\cdot)$ is a function that takes as input an integer $m \in [0, M]$ and outputs an $(M+1)$-length one-hot vector where the m-th element equals 1 and the others are zero-valued. However, the function $one\_hot(argmax(\cdot))$ is non-differentiable, which makes the networks untrainable.

[1]The concrete privacy loss ϵ_i to be used to perturb the local gradient is a sample of the random variable. To reduce notational overload, we use ϵ_i to denote the random variable in Section V.

Algorithm 3 Auction Mech.: DM-RegretNet

Input: (reported) bid profile $\boldsymbol{b}' = (b'_1, ..., b'_n)$, financial budget B, the number of sub-bids M, training=False
Output: privacy losses $\epsilon_1, .., \epsilon_n$, payments $p_1, ..., p_n$
1: Transform each data owner i bid $b'_i = (v'_i, \bar{\epsilon}'_i, d_i)$ into M sub-bids $b'_{i|1}, ..., b'_{i|M}$, where $b'_{i|m} = (v'_i(\frac{m \cdot \bar{\epsilon}'_i}{M}, d_i), \frac{m \cdot \bar{\epsilon}'_i}{M}, d_i), \forall m \in [M]$
2: Input sub-bids, privacy budgets, and financial budget into M-RegretNet to obtain $\boldsymbol{z}_1, ..., \boldsymbol{z}_n, p_1, ..., p_n$ where $\boldsymbol{z}_i = [z_{i0}, ..., z_{iM}]$
3: **if** training == True **then**
4: $\hat{\epsilon}_i = [0, \frac{1 \cdot \bar{\epsilon}'_i}{M}, ..., \frac{M \cdot \bar{\epsilon}'_i}{M}] \cdot softmax(\frac{\boldsymbol{z}_i}{\tau}), \forall i$
5: **return** $\hat{\epsilon}_1, ..., \hat{\epsilon}_n, p_1, ..., p_n$
6: **else**
7: $\epsilon_i = [0, \frac{1 \cdot \bar{\epsilon}'_i}{M}, ..., \frac{M \cdot \bar{\epsilon}'_i}{M}] \cdot one\_hot(argmax(\boldsymbol{z}_i)), \forall i$
8: **return** $\epsilon_1, ..., \epsilon_n, p_1, ..., p_n$

To realize deterministic allocation results while ensuring trainable networks, we apply the soft argmax trick [29] to DM-RegretNet. Then, as shown in Alg. 3, for the model inference phase, DM-RegretNet obtains deterministic allocation results by Equation (3); for the model training phase, it uses the following differentiable estimator to approximate Equation (3):

$$\hat{\epsilon}_i = [0, \frac{1 \cdot \bar{\epsilon}'_i}{M}, ..., \frac{M \cdot \bar{\epsilon}'_i}{M}] \cdot softmax(\boldsymbol{z_i}/\tau)$$

where τ is a smoothing parameter that controls the tradeoff between the estimator's approximation accuracy and smoothness. If we use a smaller τ, the estimator $\hat{\epsilon}_i$ will approach the truth but become harder to optimize.

Then, to further promote the approximation accuracy, we introduce the *deterministic allocation* constraint when training DM-RegretNet, which requires that $softmax(\boldsymbol{z_i}/\tau)$ should be a one-hot vector. Consider a vector $\boldsymbol{z}^U = [z_0^U, ..., z_M^U]$ with uniform allocation probabilities, i.e., $z_m^U = \frac{1}{M+1}, \forall m \in [0, M]$. Obviously, for a vector $\boldsymbol{z} = [z_0, ..., z_M]$ of allocation probabilities, the squared Euclidean distance between $\boldsymbol{z}$ and $\boldsymbol{z}^U$ is maximized only when $\boldsymbol{z}$ is a one-hot vector:

$$\sup_{\boldsymbol{z}} \sum_{m \in [0,M]} (z_m - z_m^U)^2 = (1 - \frac{1}{M+1})^2 + M(0 - \frac{1}{M+1})^2 = \frac{M}{M+1}$$

Then, we formalize the deterministic allocation constraint over the vector $\boldsymbol{z}'_i = [z'_{i0}, ..., z'_{iM}] = softmax(\boldsymbol{z_i}/\tau)$ as:

$$dav_i(\theta) = \mathbb{E}_{(\boldsymbol{b},B)}[\frac{M}{M+1} - \sum_{m \in [0,M]} (z'_{im} - z_m^U)^2] = 0.$$

where θ is the network parameters of DM-RegretNet. We note that $\boldsymbol{z}'_i$ is determined by the network parameters θ and the input (b, B) to DM-RegretNet.

Training DM-RegretNet: We train DM-RegretNet by solving Problem 2. Concretely, given a (real) bid profile $\boldsymbol{b}$ and a financial budget B, we can obtain a global gradient:

$$\tilde{g}_{\boldsymbol{\lambda},\theta} = \mathsf{Aggr}(\hat{\boldsymbol{\epsilon}}, \boldsymbol{d}) \cdot [\mathcal{M}_{\hat{\epsilon}_1}(g_1), ..., \mathcal{M}_{\hat{\epsilon}_n}(g_n)]$$

where the estimated privacy losses $\hat{\boldsymbol{\epsilon}} = [\hat{\epsilon}_1, ..., \hat{\epsilon}_n]$ are affected by the network parameters θ. The training objective thus is to find the optimal network parameters that minimize the expected error bound $\mathbb{E}_{(\boldsymbol{b},B)}[ERR(\tilde{g}_{\boldsymbol{\lambda},\theta}; \boldsymbol{\lambda} = \mathsf{Aggr}(\hat{\boldsymbol{\epsilon}}, \boldsymbol{d}))]$.

Then, we relax the truthfulness constraint and quantify the violation degree of truthfulness for data owner i by the *expected regret* (normalized by the expected valuation of her allocated privacy loss $c_i^\theta(b_i; \boldsymbol{b}_{-i}, B) = \sum_{m=1}^M z'_{im} \cdot v_i(\frac{m \cdot \bar{\epsilon}_i}{M}, \bar{d}_i)$ under parameters θ):

$$rgt_i(\theta) = \mathbb{E}_{(\boldsymbol{b},B)}[\frac{\max(0, \max_{b'_i} u_i^\theta(b'_i; \boldsymbol{b}_{-i}, B) - u_i^\theta(b_i; \boldsymbol{b}_{-i}, B))}{c_i^\theta(b_i; \boldsymbol{b}_{-i}, B)}]$$

where u_i^θ is data owner i's utility function under network parameters θ. Similarly, the violation degree of the IR constraint can be measured by the *expected IR violation*:

$$irv_i(\theta) = \mathbb{E}_{(\boldsymbol{b},B)}[\frac{max(0, -u_i^\theta(b_i; \boldsymbol{b}_{-i}, B))}{c_i^\theta(b_i; \boldsymbol{b}_{-i}, B)}]$$

Therefore, we have the following optimization problem.

Problem 4 (DM-RegretNet Training Problem)**.**

$$\min_\theta \mathbb{E}_{(\boldsymbol{b},B)}[ERR(\tilde{g}_{\boldsymbol{\lambda},\theta}; \boldsymbol{\lambda} = \mathsf{Aggr}(\hat{\boldsymbol{\epsilon}}, \boldsymbol{d}))]$$
$$\text{S.t.: } rgt_i(\theta) = 0, \forall i \quad \textit{(Truthfulness)}$$
$$irv_i(\theta) = 0, \forall i \quad \textit{(Individual Rationality)}$$
$$dav_i(\theta) = 0, \forall i \quad \textit{(Deterministic Allocation)}$$

We can empirically estimate the expected error bound and those violation degrees from some training data and solve an empirical version of Problem 4 to train DM-RegretNet. The details can be checked in Appendix A. The training data can be drawn from a known distribution or historical data. Note that DM-RegretNet is trained offline before the execution of Algorithm 1; in each FL training round, the trained auction model makes a model inference to decide the auction result, which efficiently finishes in polynomial time.

VI. Evaluation

A. Setup

Research questions: We investigate the following research questions through experiments.

- RQ1: How does the proposed auction mechanism DM-RegretNet perform compared with the baselines (explained below) in terms of minimizing the error bound?
- RQ2: Can OptAggr outperform the conventional aggregation method in FL?
- RQ3: How does DM-RegretNet approximately guarantee the truthfulness and IR constraints?
- RQ4: Does increasing M benefit approximating the truthfulness and IR guarantees?

Baselines: We compare OptAggr with the conventional aggregation method *ConvlAggr* [7], which allocates positive aggregation weights only to those data owners with nonzero privacy losses, and the weights are proportional to their data sizes. Regarding auction, we compare DM-RegretNet with RegretNet [21] and M-RegretNet.[2]

[2]Our code, data, and trained models are available at https://github.com/teijyogen/FL-Market. We use the *CVXPY* [30] and *cvxpylayers* [31] libraries to implement the OptAggr aggregation mechanism.

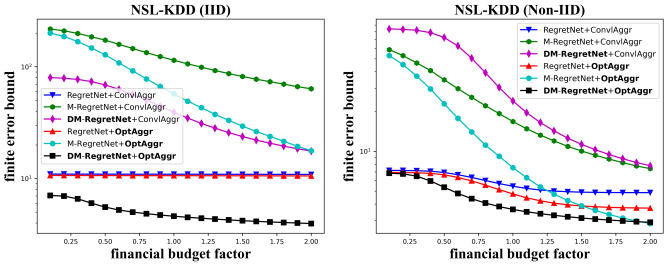

Fig. 7: Effect of financial budget on error bound.

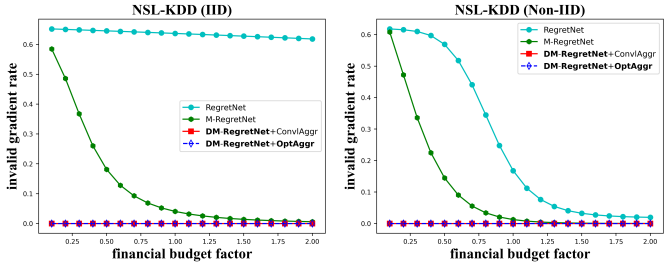

Fig. 8: Invalid gradient rate.

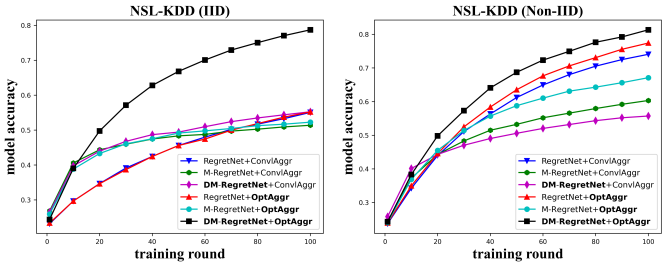

Fig. 9: Model accuracy over FL training rounds.

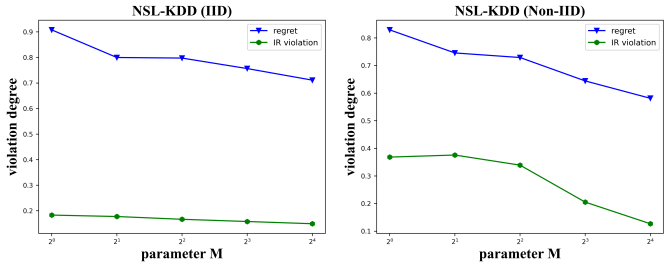

Fig. 10: Effect of parameter M.

FL settings: We use real data to train FL models. We choose logistic regression classifiers as FL models and use the NSL-KDD [32] datasets for 5-class classification with 125973 training samples and 22544 test samples. We distribute the training samples among 1000 data owners to form their local datasets using the following partition methods:

- **IID**: We follow [33] to draw all the local datasets from the same distribution, and their sizes follow a power law.
- **Non-IID**: We follow [34] to allocate each class of samples among clients according to the Dirichlet distribution.

We set the learning rate $\eta = 0.01$ and the threshold $L = 1.0$ for gradient clipping and perturb local gradients by the Laplace mechanism [6].

Auction settings: For each run of the experiment, we simulate 100 rounds of FL and generate 1000 data owners; in each round, we randomly select 10 data owners as bidders in the auction. To simulate various types of bids, we let each bidder randomly select a basic valuation function from four provided: a linear function $v^L(\epsilon_i, d_i) = 2 \cdot d_i \cdot \epsilon_i$, a quadratic function $v^Q(\epsilon_i, d_i) = d_i \cdot (\epsilon_i)^2$, a square-root function $v^S(\epsilon_i, d_i) = 2 \cdot d_i \cdot \sqrt{\epsilon_i}$, and an exponential function $v^E(\epsilon_i, d_i) = d_i \cdot (\exp(\epsilon_i) - 1)$, which are natural choices considered in [16]; these functions are directly proportional to the data size d_i because it is natural to model the valuation of a dataset as the sum of the valuations of the data records therein. Then, we consider each owner's valuation function to be a randomly selected rate $\alpha \in [0.5, 1.5]$ of the selected function, e.g., $v_i(\epsilon_i, d_i) = \alpha \cdot v^L(\epsilon_i, d_i)$. Finally, we randomly generate each data owner's privacy budget $\bar{\epsilon}_i \in [0.5, 2.0]$, which is in line with those commonly used in the differential privacy research community. For DM-RegretNet and M-RegretNet, we set $M = 8$ by default. We train all the auction models on $102,400$ bid profiles with 50 epochs.

Evaluation metric: To evaluate the utilities of the global gradients, we use as evaluation metrics the expected empirical error bound $E\hat{R}R$, the model accuracy (i.e., the percentage of correctly predicted examples), and the *invalid gradient rate* (i.e., the frequency of sampling zero-valued privacy losses for all data owners). Then, to evaluate the truthfulness and IR guarantees of the auction mechanisms, we use the empirical regret $\hat{rgt}_i$ and empirical IR violation $\hat{irv}_i$ as the metrics. The definitions of the above metrics can be found in Appendix A.

B. Experimental Results

Incentive mechanisms comparison (RQ1): First, we test the auction mechanisms' performance in minimizing the error bound. We vary the financial budget factor $\bar{B}$ and let the budget $B = \bar{B} \cdot \sum_{i \in [n]} v_i(\bar{\epsilon}_i, \bar{d}_i)$. As shown in Figure 7, our DM-RegretNet can generate global gradients with a lower error bound in expectation. When the financial budget factor increases and exceeds 1.0, which means the budget covers the gross valuation of the bidders' privacy budgets, the error bound may still be able to decrease since the payments made by a truthful auction mechanism are usually much higher than the winners' valuations. We note that since the randomized mechanisms RegretNet and M-RegretNet may sample zero-valued privacy losses for all data owners, which results in *invalid global gradients* with infinite error, we only take the error bound of valid gradients into account. That means that even if Figure 7 shows that RegretNet results in low error bounds, it actually frequently generates invalid gradients with infinite error, while our DM-RegretNet based mechanisms never do, which is depicted in Figure 8. For the rest experiments, we sample the budget factor uniformly at random from $[0.1, 2.0]$. We also test the model accuracy over 100 FL training rounds. As shown in Figure 9, in both cases, DM-RegretNet makes better auction decisions that result in more accurate models.

TABLE I: Comparisons of the violation degrees of truthfulness and IR of RegretNet-based mechanisms. At each box, the two numbers are the empirical regret and IR violation, respectively.

	IID	Non-IID
RegretNet	0.9351, 0.1684	0.8164, 0.3864
M-RegretNet	0.7715, 0.1508	0.6652, 0.2020
DM-RegretNet+ConvlAggr	0.0617, 0.0251	0.0516, 0.0210
DM-RegretNet+OptAggr	0.0556, 0.0265	0.0428, 0.0259

Aggregation mechanisms comparison (RQ2): As depicted in Figure 7, under each auction mechanism, our OptAggr aggregation mechanism can always generate global gradients with a lower error bound in expectation than ConvlAggr. In addition, Figure 9 shows that model buyers can obtain more accurate models using global gradients aggregated by OptAggr. Therefore, OptAggr outperforms ConvlAggr.

Incentive guarantees (RQ3).: Table I illustrates the violation degrees of truthfulness and IR of those RegretNet-based auction mechanisms. The empirical regrets and IR violations under DM-RegretNet are significantly lower than those under RegretNet and M-RegretNet, which means that DM-RegretNet has stronger abilities to approximate the truthfulness and IR constraints. DM-RegretNet has this advantage because it is a deterministic mechanism that universally guarantees truthfulness and IR, while RegretNet and M-RegretNet are randomized mechanisms that approximate the two constraints by expectation.

Parameter effects (RQ4): We vary the value of parameter $M \in \{1, 2, 4, 8, 16\}$ to test its effects on the truthfulness and IR guarantees. For each value, we train 10 instances of M-RegretNet and test them to obtain the average result. Figure 10 shows that under M-RegretNet, an increase in M decreases both the regret and IR violation, which demonstrates our intuition that a larger M can enhance the abilities of M-RegretNet to approximate the truthfulness and IR guarantees as it has more possible values to allocate as privacy losses.

VII. Related Work

Incentive mechanisms for FL: Many incentive mechanisms [35]–[42] have been proposed to encourage participation in FL by providing appropriate rewards for data owners' contributions. The contributions can be evaluated in various ways. For example, Zhan et al. [41] consider the data size, the most basic measurement of data, for contribution evaluation. Then, from a cost-covering perspective, Jiao et al. [35] propose an auction mechanism where data owners can bid their computational and communication costs in providing their FL training services. Similarly, Sarikayar et al. [38] regard the CPU computational costs as their contributions. Then, Richardson et al. [36] evaluate data owners' influences on the model accuracy to decide their rewards. The Shapley value is also adapted into an FL version by Wang et al. [37] to value data owners' influence. Data quality is another natural choice. Since the data quality is known only to data owners, to ensure the contribution of high-quality data, Kang et al. [39] design different types of rewarding contracts to distinguish data owners such that the FL server can infer the data quality based on the contracts they select. In this way, the rewards are essentially determined by the data quality. In addition, both the works of Kang et al. [40] and Zhang et al. [42] employ some reputation metric to remove unreliable data owners from FL. However, none of the above mechanisms considers privacy protection, which is also a critical incentive. To fill this gap, we propose an auction-based incentive mechanism that protects data owners' privacy and compensates them according to their privacy preferences.

FL under LDP: Some efforts [9]–[13], [43]–[45] have devoted to designing FL frameworks under LDP. Since the data perturbation under LDP may substantially reduce the utility of FL models, these authors mainly focus on how to reduce the perturbation level while still providing appropriate privacy guarantees. Concretely, to relieve the utility problem that the noise that LDP injects into a gradient should be proportional to its size, Liu et al. [11] propose an FL framework to perturb only the top-k important dimensions of the gradient and thus can its utility. Then, Liu et al. [12] and Girgis et al. [45] employ the shuffle model [46] in their FL frameworks to amplify the privacy guarantee under the same level of noise injection. Then, Sun et al. [10] propose a more secure LDP mechanism that can extend the difference between the perturbed data and its original value while introducing lower variance. There are also works on designing LDP-based FL frameworks for specific ML tasks [9], [13], [43]. While prior works address the utility problem under LDP by relaxing the privacy guarantee or elaborately injecting noise, we tackle it from an incentive perspective, i.e., by incentivizing data owners to contribute more privacy loss, which can also increase utility. In addition, Zhao et al. [44] propose an LDP-based FedSGD algorithm, which is similar to our protocol in privacy protection; however, they assume uniform privacy losses for all data owners and thus do not consider different perturbation levels when aggregating gradients.

VIII. Conclusion and Future Work

In this paper, we propose FL-Market to facilitate trustworthy data acquisition for ML-based data analytics. Our mechanisms can incentivize data sharing by providing preferred levels of local privacy and compensation for data owners and optimizing model buyers' utility. FL-Market opens up new possibilities for ML-oriented data acquisition and initiates a new direction toward designing locally private model marketplaces. There are several interesting future directions. One question is how to guarantee that the auction decisions are arbitrage free against strategic buyers. Another question is how to apply and optimize FL-Market in specific learning tasks.

IX. Acknowledgment

We thank the Japan Society for the Promotion of Science (JSPS) for its generous and continued support for the first author who conducted this research as a JSPS Research Fellow. In addition, this work was partially supported by JST CREST (No. JPMJCR21M2), JST SICORP (No. JPMJSC2107), and JSPS KAKENHI (No. 21J23090, 21K19767, 22H03595).

REFERENCES

[1] L. Chen, P. Koutris, and A. Kumar, "Towards model-based pricing for machine learning in a data marketplace," in *ACM SIGMOD International Conference on Management of Data*, 2019, p. 1535–1552.

[2] R. Jia, D. Dao, B. Wang, F. A. Hubis, N. M. Gurel, B. Li, C. Zhang, C. Spanos, and D. Song, "Efficient task-specific data valuation for nearest neighbor algorithms," *Proceedings of the VLDB Endowment*, vol. 12, no. 11, p. 1610–1623, 2019.

[3] A. Agarwal, M. Dahleh, and T. Sarkar, "A marketplace for data: An algorithmic solution," in *ACM Conference on Economics and Computation*, 2019, p. 701–726.

[4] J. Liu, J. Lou, J. Liu, L. Xiong, J. Pei, and J. Sun, "Dealer: An end-to-end model marketplace with differential privacy," *Proceedings of the VLDB Endowment*, vol. 14, no. 6, pp. 957–969, 2021.

[5] X. Jiang, C. Niu, C. Ying, F. Wu, and Y. Luo, "Pricing GAN-based data generators under Rényi differential privacy," *Information Sciences*, vol. 602, pp. 57–74, 2022.

[6] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of Cryptography*, 2006, pp. 265–284.

[7] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *International Conference on Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[8] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," in *Annual Conference on Neural Information Processing Systems*, 2019, pp. 14 747–14 756.

[9] Y. Wang, Y. Tong, and D. Shi, "Federated Latent Dirichlet Allocation: A local differential privacy based framework," in *AAAI Conference on Artificial Intelligence*, 2020, pp. 6283–6290.

[10] L. Sun, J. Qian, and X. Chen, "LDP-FL: Practical private aggregation in federated learning with local differential privacy," in *International Joint Conference on Artificial Intelligence*, 2021, pp. 1571–1578.

[11] R. Liu, Y. Cao, M. Yoshikawa, and H. Chen, "FedSel: Federated SGD under local differential privacy with top-k dimension selection," in *International Conference on Database Systems for Advanced Applications*, 2020, pp. 485–501.

[12] R. Liu, Y. Cao, H. Chen, R. Guo, and M. Yoshikawa, "FLAME: Differentially private federated learning in the shuffle model," in *AAAI Conference on Artificial Intelligence*, 2021, pp. 8688–8696.

[13] C. Wu, F. Wu, L. Lyu, Y. Huang, and X. Xie, "FedCTR: Federated native ad CTR prediction with cross-platform user behavior data," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, no. 4, 2022.

[14] A. Evfimievski, J. Gehrke, and R. Srikant, "Limiting privacy breaches in privacy preserving data mining," in *ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 2003, p. 211–222.

[15] A. Ghosh and A. Roth, "Selling privacy at auction," in *ACM Conference on Electronic Commerce*, 2011, pp. 199–208.

[16] ——, "Selling privacy at auction," *Games and Economic Behavior*, vol. 91, pp. 334–346, 2015.

[17] A. Roth, "Buying private data at auction: The sensitive surveyor's problem," *SIGecom Exch.*, vol. 11, no. 1, p. 1–8, 2012.

[18] K. Nissim, C. Orlandi, and R. Smorodinsky, "Privacy-aware mechanism design," in *ACM Conference on Electronic Commerce*, 2012, p. 774–789.

[19] L. K. Fleischer and Y.-H. Lyu, "Approximately optimal auctions for selling privacy when costs are correlated with data," in *ACM Conference on Electronic Commerce*, 2012, pp. 568–585.

[20] K. Nissim, S. Vadhan, and D. Xiao, "Redrawing the boundaries on purchasing data from privacy-sensitive individuals," in *Innovations in Theoretical Computer Science*, 2014, p. 411–422.

[21] P. Duetting, Z. Feng, H. Narasimhan, D. C. Parkes, and S. S. Ravindranath, "Optimal auctions through deep learning," in *ICML*, 2019, pp. 1706–1715.

[22] Y. Bengio, P. Y. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[23] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *ICML*, 2013, pp. 1310–1318.

[24] S. Zheng, Y. Cao, and M. Yoshikawa, "Money cannot buy everything: Trading mobile data with controllable privacy loss," in *IEEE International Conference on Mobile Data Management*, 2020, pp. 29–38.

[25] M. Mohri, G. Sivek, and A. T. Suresh, "Agnostic federated learning," in *ICML*, 2019, pp. 4615–4625.

[26] H. Chan and J. Chen, "Truthful multi-unit procurements with budgets," in *Web and Internet Economics*, 2014, pp. 89–105.

[27] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, "OSQP: An operator splitting solver for quadratic programs," *Mathematical Programming Computation*, vol. 12, no. 4, pp. 637–672, 2020.

[28] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd, "Conic optimization via operator splitting and homogeneous self-dual embedding," *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 1042–1068, 2016.

[29] O. Chapelle and M. Wu, "Gradient descent optimization of smoothed information retrieval metrics," *Information Retrieval*, vol. 13, no. 3, pp. 216–235, 2010.

[30] S. Diamond and S. Boyd, "CVXPY: A Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[31] A. Agrawal, B. Amos, S. Barratt, S. Boyd, S. Diamond, and Z. Kolter, "Differentiable convex optimization layers," in *Advances in Neural Information Processing Systems*, 2019, pp. 9558–9570.

[32] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, 2009, pp. 1–6.

[33] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," in *The Conference on Machine Learning and Systems*, 2020, pp. 429–450.

[34] M. Yurochkin, M. Agarwal, S. Ghosh, K. Greenewald, N. Hoang, and Y. Khazaeni, "Bayesian nonparametric federated learning of neural networks," in *ICML*, 2019, pp. 7252–7261.

[35] Y. Jiao, P. Wang, D. Niyato, B. Lin, and D. I. Kim, "Toward an automated auction framework for wireless federated learning services market," *IEEE Transactions on Mobile Computing*, vol. 20, no. 10, pp. 3034–3048, 2021.

[36] A. Richardson, A. Filos-Ratsikas, and B. Faltings, "Rewarding high-quality data via influence functions," *arXiv preprint arXiv:1908.11598*, 2019.

[37] T. Wang, J. Rausch, C. Zhang, R. Jia, and D. Song, "A principled approach to data valuation for federated learning," in *Federated Learning: Privacy and Incentive*, 2020, vol. 12500, pp. 153–167.

[38] Y. Sarikaya and O. Ercetin, "Motivating workers in federated learning: A Stackelberg game perspective," *IEEE Networking Letters*, vol. 2, no. 1, pp. 23–27, 2020.

[39] J. Kang, Z. Xiong, D. Niyato, H. Yu, Y. Liang, and D. I. Kim, "Incentive design for efficient federated learning in mobile networks: A contract theory approach," in *IEEE VTS Asia Pacific Wireless Communications Symposium*, 2019, pp. 1–5.

[40] J. Kang, Z. Xiong, D. Niyato, Y. Zou, Y. Zhang, and M. Guizani, "Reliable federated learning for mobile networks," *IEEE Wireless Communications*, vol. 27, no. 2, pp. 72–80, 2020.

[41] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6360–6368, 2020.

[42] J. Zhang, Y. Wu, and R. Pan, "Incentive mechanism for horizontal federated learning based on reputation and reverse auction," in *The Web Conference 2021*, 2021, pp. 947–956.

[43] J. Li, M. Khodak, S. Caldas, and A. Talwalkar, "Differentially private meta-learning," in *International Conference on Learning Representations*, 2020.

[44] Y. Zhao, J. Zhao, M. Yang, T. Wang, N. Wang, L. Lyu, D. Niyato, and K.-Y. Lam, "Local differential privacy based federated learning for internet of things," *IEEE Internet of Things Journal*, 2020.

[45] A. Girgis, D. Data, S. Diggavi, P. Kairouz, and A. Theertha Suresh, "Shuffled model of differential privacy in federated learning," in *International Conference on Artificial Intelligence and Statistics*, 2021, pp. 2521–2529.

[46] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by shuffling: From local to central differential privacy via anonymity," in *Annual ACM-SIAM Symposium on Discrete Algorithms*, 2019, pp. 2468–2479.

[47] R. B. Myerson, "Optimal auction design," *Mathematics of Operations Research*, vol. 6, no. 1, pp. 58–73, 1981.

APPENDIX A
TRAINING DM-REGRETNET

Consider a training sample $\mathcal{S} = (S^1, ..., S^T)$ consisting of T batches. Each batch $S^t = ((b^{(1)}), B^{(1)}), ..., (b^{(K)}, B^{(K)})), t \in [T]$ has K pairs of real bid profiles and financial budgets, and each profile $b^{(k)}, k \in [K]$ consists of a valuation function $v_i^{(k)}$, a privacy budget $\bar{\epsilon}_i^{(k)}$, and a data size $\bar{d}_i^{(k)}$. Then, at each training iteration t, we can estimate $rgt_i(\theta^t)$ by the *empirical regret*:

$$\hat{rgt}_i(\theta^t) = \frac{1}{K}\sum_{k=1}^{K} \frac{max(0, u_i^{\theta^t}(b_i^{*(k)}; \boldsymbol{b}_{-i}^{(k)}, B^{(k)}) - u_i^{\theta^t}(b_i^{(k)}; \boldsymbol{b}_{-i}^{(k)}, B^{(k)}))}{c_i^{\theta^t}(b_i^{(k)}; \boldsymbol{b}_{-i}^{(k)}, B^{(k)})}$$

where θ^t represents the network parameters at training iteration t and $b_i^{*(k)}$ is a bid that approximately maximizes i's utility and is searched through J updates of the following optimization process:

$$b_i'^{(k)} \leftarrow b_i'^{(k)} + \gamma \nabla_{b_i'} u_i^{\theta^t}(b_i'; \boldsymbol{b}_{-i}^{(k)}, B^{(k)})|_{b_i' = b_i'^{(k)}}$$

Similarly, we estimate $irv_i(\theta^t)$ by the *empirical IR violation*:

$$\hat{irv}_i(\theta^t) = \frac{1}{K}\sum_{k=1}^{K} \frac{max(0, -u_i^{\theta^t}(b_i^{(k)}; \boldsymbol{b}_{-i}^{(k)}, B^{(k)}))}{c_i^{\theta^t}(b_i^{(k)}; \boldsymbol{b}_{-i}^{(k)}, B^{(k)})}$$

Let $[z_{i0}^{(k)}, ..., z_{iM}^{(k)}]$ denote the allocation probabilities for data owner i given bid profile $b^{(k)}$ and financial budget $B^{(k)}$ under network parameters θ^t, and let $[z_{i0}'^{(k)}, ..., z_{iM}'^{(k)}] = softmax(\frac{[z_{i0}^{(k)}, ..., z_{iM}^{(k)}]}{\tau})$. Then, we have the *empirical deterministic allocation violation* $\hat{dav}_i(\theta^t)$ to estimate $dav_i(\theta^t)$:

$$\hat{dav}_i(\theta^t) = \frac{1}{K}\sum_{k=1}^{K}[\frac{M}{M+1} - \sum_{m\in[0,M]}(z_{im}'^{(k)} - \frac{1}{M+1})^2]$$

Finally, we should derive an empirical version of the expected error bound $\mathbb{E}_{(b,B)}[ERR(\tilde{g}_{\boldsymbol{\lambda},\theta}; \boldsymbol{\lambda} = \mathsf{Aggr}(\hat{\boldsymbol{\epsilon}}, \boldsymbol{d}))]$. Let $\hat{\epsilon}_1^{(k)}, ..., \hat{\epsilon}_n^{(k)}$ denote the estimated privacy losses determined by DM-RegretNet for bid profile $b^{(k)}$ and financial budget $B^{(k)}$. Given aggregation weights $\hat{\lambda}_1^{(k)}, ..., \hat{\lambda}_n^{(k)} = \mathsf{Aggr}([\hat{\epsilon}_1^{(k)}, ..., \hat{\epsilon}_n^{(k)}], [\bar{d}_1^{(k)}, ..., \bar{d}_n^{(k)}])$ and $W_i^{(k)} = \frac{\bar{d}_i^{(k)}}{\sum_{j\in[n]}\bar{d}_j^{(k)}}, \forall i$, we have the *empirical expected error bound*:

$$\hat{ERR}(\theta^t) = \frac{1}{K}\sum_{k=1}^{K}\sup_{g_1,...,g_n} ||\sum_{i=1}^{n}\hat{\lambda}_i^{(k)}\cdot\mathcal{L}^L_{\hat{\epsilon}_i^{(k)}}(g_i) - \sum_{i=1}^{n} W_i^{(k)} g_i||_2$$

We can solve Problem 4 by the augmented Lagrangian method and minimize the following Lagrangian function: [3]

$$\begin{aligned}&\mathcal{C}(\theta^t; \phi_{rgt}^t, \phi_{irv}^t, \phi_{dav}^t)\\ =&n\cdot\hat{ERR}(\theta^t) + \sum_{i=1}^{n}\phi_{rgt,i}^t\cdot\hat{rgt}_i(\theta^t) + \frac{\rho_{rgt}}{2}(\sum_{i=1}^{n}\hat{rgt}_i(\theta^t))^2\\ &+\sum_{i=1}^{n}\phi_{irv,i}^t\cdot\hat{irv}_i(\theta^t) + \frac{\rho_{irv}}{2}(\sum_{i=1}^{n}\hat{irv}_i(\theta^t))^2\\ &+\sum_{i=1}^{n}\phi_{dav,i}^t\cdot\hat{dav}_i(\theta^t) + \frac{\rho_{dav}}{2}(\sum_{i=1}^{n}\hat{dav}_i(\theta^t))^2\end{aligned}$$

where $\phi_{rgt}^t, \phi_{irv}^t, \phi_{dav}^t \in R^n$ are vectors of Lagrange multipliers and $\rho_{rgt}, \rho_{irv}, \rho_{dav} > 0$ are fixed hyperparameters that control the quadratic penalties. Finally, the network parameters of DM-RegretNet are updated at each iteration t as:

$$\theta^{t+1} \leftarrow \theta^t - \psi\nabla_\theta\mathcal{C}(\theta^t; \phi_{rgt}^t, \phi_{irv}^t, \phi_{dav}^t)$$

and the Lagrange multipliers are updated every Q iterations as:

$$\begin{aligned}\text{If } t \bmod Q = 0 : \forall i, \phi_{rgt,i}^{t+1} &\leftarrow \phi_{rgt,i}^t + \rho_{rgt}\cdot\hat{rgt}_i(\theta^t)\\ \phi_{irv,i}^{t+1} &\leftarrow \phi_{irv,i}^t + \rho_{irv}\cdot\hat{irv}_i(\theta^t)\\ \phi_{dav,i}^{t+1} &\leftarrow \phi_{dav,i}^t + \rho_{dav}\cdot\hat{dav}_i(\theta^t)\end{aligned}$$

In our experiments, we fine-tune and set the hyperparameters as follows: $T = 100$, $K = 1024$, $J = 100$, $Q = 10$, $\gamma = 0.1$, $\psi = 0.001$, and $\phi_{rgt,i}^1 = \phi_{irv,i}^1 = \phi_{dav,i}^1 = 1.0$; the allocation (payment) network consists of 2 hidden layers and 100 hidden nodes per layer. We train each model for 50 epochs. In addition, we set $\rho_{rgt} = \rho_{irv} = \rho_{dav} = 1.0$ at the first epoch of training and increase ρ_{rgt}, ρ_{irv} in steps of 1.0 at the end of every epoch. We note that since we only need the bid profiles and financial budgets to train DM-RegretNet, which are assumed to be nonprivate, fine-tuning the hyperparameters of DM-RegretNet does not cause any privacy leakage.

APPENDIX B
ALL-IN: SINGLE-MINDED AUCTION MECHANISM

We propose an auction mechanism All-in for *single-minded* data owners, each of whom has a step valuation function $v_i(\epsilon_i, d_i) = \begin{cases} V_i, & \epsilon_i \in (0, \bar{\epsilon}_i], d_i \in (0, \bar{d}_i] \\ 0, & \epsilon_i = 0 \text{ or } d_i = 0 \end{cases}$ where $V_i > 0$ is a constant set by i. Therefore, we can use V_i and V_i' to represent the real valuation v_i and the reported valuation v_i', respectively. Such cases are common in practice because some data owners are just willing to sell all their small datasets and privacy budgets at a single round of auction or only focus on whether their private information is leaked rather than how much is leaked. Obviously, each data owner i can only have two kinds of auction results: (1) win the auction with $\epsilon_i = \bar{\epsilon}_i'$ or (2) lose the auction with $\epsilon_i = 0$.

To meet the demands of single-minded bidders, we can design a truthful mechanism using Myerson's characterization [47], which indicates that the *monotonicity* and *critical payment* properties imply truthfulness. Concretely, monotonicity requires that a winner should still win if she re-reports a higher privacy budget, a larger data size, and/or a lower valuation with other bidders' bids fixed; the critical payment property ensures that winners are paid the maximum possible payments (i.e., critical payments) and hence that they have no incentive to

[3] When training RegretNet and M-RegretNet, we minimize the negated empirical privacy loss $\hat{NPL}(\theta^t) = -\frac{1}{K\cdot n}\sum_{k=1}^{K}\sum_{i=1}^{n} W_i^{(k)}\mathbb{E}[\epsilon_i^{(k)}]$ instead of $\hat{ERR}(\theta^t)$, where $\mathbb{E}[\epsilon_i^{(k)}]$ is the expected privacy loss of the i-th data owner of the k-th bid profile at the t-th batch.

misreport bids. However, the limited financial budget makes the problem more difficult because the winner selection should depend on the payments, which in turn depend on the selection results. Hence, we should carefully identify budget-feasible critical payments.

Algorithm 4 Auction Mech.: All-in

Input: (reported) bid profile $b' = (b'_1, ..., b'_n)$, financial budget B

Output: privacy losses $\epsilon_1, .., \epsilon_n$, payments $p_1, ..., p_n$

1: Calculate the unit valuations on privacy budgets: $\forall i, v_i^{unit} = \frac{V'_i}{d_i \cdot \bar{\epsilon}'_i}$
2: Sort data owners in ascending order of v_i^{unit}
3: Initialize the winner set $\mathcal{W} = \emptyset$ and critical unit payment $p^{unit} = 0$
4: **for** each data owner i in the sorted order **do**
5: If $v_i^{unit} \leq \frac{B}{\sum_{j \in \mathcal{W} \cup \{i\}} d_j \cdot \bar{\epsilon}'_j}$, add i into $\mathcal{W}$ and update critical unit payment $p^{unit} = \frac{B}{\sum_{j \in \mathcal{W} \cup \{i\}} d_j \cdot \bar{\epsilon}'_j}$
6: Calculate privacy losses: $\forall i, \epsilon_i = \bar{\epsilon}'_i$ if $i \in \mathcal{W}$; otherwise $\epsilon_i = 0$
7: Calculate payments: $\forall i, p_i = d_i \cdot \epsilon_i \cdot p^{unit}$
8: **return** $\epsilon_1, ..., \epsilon_n, p_1, ..., p_n$

To capture the interdependency between the winner selection and payment decision, All-in takes the payments into account when selecting winners. Concretely, to guarantee monotonicity, All-in selects data owners in ascending order of their *unit valuations* $v_i^{unit} = \frac{V'_i}{d_i \cdot \bar{\epsilon}'_i}$; intuitively, if a owner i decreases her valuation V'_i, increases her data size d_i, and/or increases her privacy budget $\bar{\epsilon}'_i$, she stays at the same position or moves to a former position in the order. Then, the winner selection procedure is to find the last owner whose unit valuation v_i^{unit} is covered by the critical unit price $\frac{B}{\sum_{j \in \mathcal{W} \cup \{i\}} d_j \cdot \bar{\epsilon}'_j}$. In this design, the winners' payments that exhaust the financial budget B are critical because if a winner i claims a higher unit valuation $v_i^{unit'} > p^{unit}$ to gain a higher payment, she definitely loses the auction due to the violation of BF. Therefore, truthfulness is ensured.

Proposition 2. *All-in satisfies truthfulness, IR, and BF.*

Proof. All-in satisfies IR because the critical unit payment p^{unit} is no lower than each winner i's unit valuation v_i^{unit}. Then, we prove that All-in satisfies truthfulness. Let V'_i be the reported V_i, $U_i = u_i(b_i; b'_{-i}, B)$ and $U'_i = u_i(b'_i, b'_{-i}, B)$. For each data owner i, we should discuss four cases as follows.

1) $\bar{\epsilon}'_i > \bar{\epsilon}_i$ and/or $d_i > \bar{d}_i$: Obviously, data owner i has no incentive because $U'_i = -\infty$.
2) $\bar{\epsilon}'_i < \bar{\epsilon}_i$ and/or $d_i < \bar{d}_i$: In the worst case, the critical unit payment is $p^{unit'} = \frac{B}{\sum_{j \in \mathcal{W}} d_j \cdot \bar{\epsilon}'_j}$. Then, we have $U'_i = d_i \cdot \bar{\epsilon}'_i \cdot p^{unit'} - V_i = \frac{B \cdot d_i \cdot \bar{\epsilon}'_i}{\sum_{j \in \mathcal{W}} d_j \cdot \bar{\epsilon}'_j} - V_i < \frac{B \cdot \bar{d}_i \cdot \bar{\epsilon}_i}{\sum_{j \in \mathcal{W}/i} d_j \cdot \bar{\epsilon}'_j + \bar{d}_i \cdot \bar{\epsilon}_i} - V_i = U_i$.
3) $\bar{\epsilon}'_i = \bar{\epsilon}_i$, $d_i = \bar{d}_i$ and $V'_i > V_i$: If $\frac{V'_i}{d_i \cdot \bar{\epsilon}'_i}$ is higher than the critical unit payment $p^{unit} = \frac{B}{\sum_{j \in \mathcal{W}/i} d_j \cdot \bar{\epsilon}'_j + \bar{d}_i \cdot \bar{\epsilon}_i}$, she loses the auction; otherwise, her utility does not change because the critical payment is unchanged.
4) $\bar{\epsilon}'_i = \bar{\epsilon}_i$, $d_i = \bar{d}_i$ and $V'_i < V_i$: Her utility does not change because of the unchanged critical payment.

□