Preventing Strategic Behaviors in Collaborative Inference for Vertical Federated Learning

Yidan Xing
Shanghai Jiao Tong University
Shanghai, China
katexing@sjtu.edu.cn

Zhenzhe Zheng*
Shanghai Jiao Tong University
Shanghai, China
zhengzhenzhe@sjtu.edu.cn

Fan Wu
Shanghai Jiao Tong University
Shanghai, China
fwu@cs.sjtu.edu.cn

ABSTRACT

Vertical federated learning (VFL) is an emerging collaborative machine learning paradigm to facilitate the utilization of private features distributed across multiple parties. During the inference process of VFL, the involved parties need to upload their local embeddings to be aggregated for the final prediction. Despite its remarkable performances, the inference process of the current VFL system is vulnerable to the strategic behavior of involved parties, as they could easily change the uploaded local embeddings to exert direct influences on the prediction result. In a representative case study of federated recommendation, we find the allocation of display opportunities to be severely disrupted due to the parties' preferences in display content. In order to elicit the true local embeddings for VFL system, we propose a distribution-based penalty mechanism to detect and penalize the strategic behaviors in collaborative inference. As the key motivation of our design, we theoretically prove the power of constraining the distribution of uploaded embeddings in preventing the dishonest parties from achieving higher utility. Our mechanism leverages statistical two-sample tests to distinguish whether the distribution of uploaded embeddings is reasonable, and penalize the dishonest party through deactivating her uploaded embeddings. The resulted mechanism could be shown to admit truth-telling to converge to a Bayesian Nash equilibrium asymptotically under mild conditions. The experimental results further demonstrate the effectiveness of the proposed mechanism to reduce the dishonest utility increase of strategic behaviors and promote the truthful uploading of local embeddings in inferences.

CCS CONCEPTS

• **Theory of computation → Algorithmic game theory and mechanism design**; • **Computing methodologies → Machine learning**.

KEYWORDS

Collaborative Inference; Strategic Behaviors; Mechanism Design; Vertical Federated Learning;

*Zhenzhe Zheng is the corresponding author.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3671663

ACM Reference Format:
Yidan Xing, Zhenzhe Zheng, and Fan Wu. 2024. Preventing Strategic Behaviors in Collaborative Inference for Vertical Federated Learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3671663

1 INTRODUCTION

With the development of machine learning techniques, the consensus that richer features and more available data could enhance prediction performances has been widely established. In recent years, federated learning (FL) is proposed as a cutting-edge collaborative machine learning paradigm to take advantage of the distributed data while protecting data privacy. The FL techniques are generally classified into horizontal FL (HFL) and vertical FL (VFL) [18] according to the distributed patterns of data. Targeting at the scenario with each party holding different features for an aligning set of samples, VFL requires each involved party to implement a local model which maps her local features to local embeddings, and requires the server to implement a top model that maps the aggregated local embeddings uploaded by the parties to the final prediction result (Figure 1). VFL techniques have been widely deployed in various scenarios, especially in recommendation system [16, 38], online advertising [20, 37], and finance [5, 6].

Despite the promising performances of VFL, we notice an unexplored deficiency of this collaborative paradigm: the inference process in VFL is vulnerable to the strategic manipulations on the uploaded local embeddings. Compared to HFL and centralized machine learning methods, the inference process in VFL requires each involved party to collaboratively upload the local embeddings for the current sample. As the learning models have been determined at the inference stage, the local embedding uploaded by one involved party could exert direct influences on the inference result, leaving chances for the party to manipulate the inference result in an predictable way. On the other hand, the involved parties may indeed have the motivations to strategically change the inference results towards their desired ones. For example, an organization may prefer to create better prediction for content belong or relevant to it when providing user behavioral feature embeddings to a recommendation system, and a bank may prefer to misguide other banks to provide lower loan limit for factually credible clients, with the aim to attract those clients and promote its own transactions.

In this work, we aim to formally investigate such kinds of strategic behaviors and the corresponding manipulation-resistant mechanism when collaborative inferences meet the strategic intentions of involved parties in VFL system. To characterize the behavioral pattern of involved parties, we resort to the celebrated concept of *utility function* and *Nash equilibrium* in game theory to describe

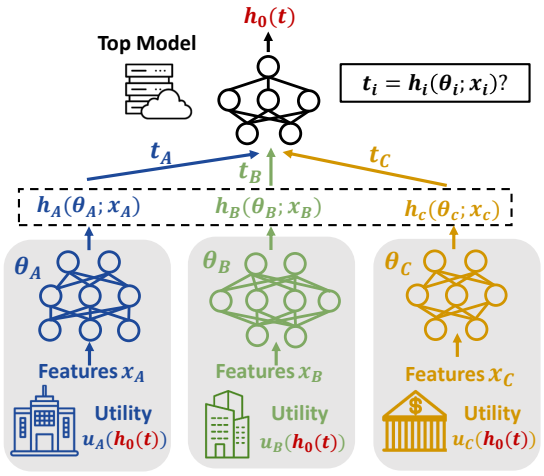


Figure 1: Overview of VFL System. The local embeddings uploaded by a party may differ from the true local embeddings.

the objective of the parties and a stable state of the strategic interactions, respectively.

In order to inspect the potential consequences of such strategic behaviors, we formulate a representative *federated recommendation game* between organizations who collaborate to provide item recommendation for users. The utilities of these organizations are defined as the exposure of item owned by them, and they could manipulate the uploaded user embeddings to change the predicted score and the allocation of exposure. Prominently, we find the resulted pure Nash equilibrium would indistinguishably allocate the exposure opportunities in a random way when there are two organizations in the game with comparable power, regardless of the properties of items they owned. In other words, when parties are obsessed with manipulating the uploaded embeddings for their own utility, the design of recommendation system would losses its original spirit, necessitating a manipulation-resistant mechanism against such strategic behaviors in VFL system.

While the strategic behaviors essentially arises from the inconsistency between the utilities of parties and the objective of VFL system, a natural idea is to introduce external monetary transfer to cover the misalignment between them following the mainstream of incentive mechanism design [40]. However, even if we do not consider the implementation practicability of a monetary transfer mechanism within the VFL system, its working principle would be unaffordable in our context. Since we need to preserve the correctness of the inference results, the final predictions could not be modified in any form to satisfy strategic intentions of parties. To elicit the true local embeddings, the money transfer mechanism should guarantee the sum of monetary reward and the utility of current prediction to be larger than any other strategy that may modify the true embeddings, thus requiring this sum to be at least the utility of the best possible prediction achievable through manipulation. As a result, the inference result worse for a party should simultaneously bring her larger monetary reward, leading the resulted expenditure to be incredibly large for the server and highly fluctuating on the distribution of inference results.

Given the infeasibility of adopting external monetary rewards, a manipulation-resistant mechanism have to be implemented fully based on the collaborative inference process. Due to the intrinsic uncertainty of data, it is generally impossible to confirm whether a specific local embedding has been manipulated, which compels us to consider utilizing the historical statistics of the embeddings to identify and constrain strategic behaviors. Nevertheless, it is unclear what kinds of statistics should be adopted among the mass of candidates, and whether these metrics could indeed help with our goal to prevent the considered strategic behaviors.

Inspired by the traditional economics literature [17], we consider the distribution information of local embeddings as a strong candidate to serve as the cornerstone of our manipulation-resistant mechanism. In particular, when the distribution of uploaded inference embeddings are enforced to align with its prior distribution, we demonstrate that the involved parties are unable to realize any dishonest utility increase in collaborative inference under mild assumptions describing the partial alignment between the server prediction function and the utilities of parties, and the training embeddings could also serve as the reference for prior distribution.

Despite these strong guarantees, due to the high dimensional nature of local embeddings in VFL applications, it is implausible for the server to realize precise restriction on the distribution of uploaded embeddings, thus initiates our final design of *distribution-based penalty mechanism*. Our mechanism works in alternative between two process: detection of potential strategic behaviors and penalization for the detected strategic behaviors. During the collaborative inference process, we periodically detect whether the uploaded embeddings follow the same distribution as the training embeddings with high probability, which is realized through applying statistical two-sample tests [12, 21], and temporarily deactivate the embeddings uploaded by a party as penalty when she is detected to be cheating. We theoretically prove the convergence of truth-telling strategies to a Bayesian Nash equilibrium in the large sample limit under appropriate mechanism parameters, which underscores the rationality and efficacy of our design. To further validate its empirical performances, we conduct extensive experiments to observe the influences of our penalty mechanism on different strategies. It turns out the utility of various manipulating strategies could be largely reduced to be similar or less than the utility of truth-telling strategy, thus effectively alleviating the incentives of parties to conduct strategic behaviors.

The main contributions of this work are summarized as follows:

- We consider the strategic behaviors to manipulate the local embeddings in collaborative inference for VFL system, which are unexplored in previous work. In a representative federated recommendation game, we demonstrate the destructive effects of strategic behaviors on prediction results when involved parties achieve a Nash equilibrium.
- We propose the distribution-based penalty mechanism as a flexible plug-in module of the vanilla VFL algorithm to prevent the considered strategic behaviors of manipulating local embeddings. With theoretically motivated design, the truth-telling strategy would converge to Beyesian Nash equilibrium under large sample limit, thus alleviates the strategic incentives of involved parties.

- We evaluate the proposed penalty mechanisms on public datasets for two typical manipulation strategies and their probabilistic variants. The empirical results validate the effectiveness of the proposed mechanisms in reducing the utility obtained by dishonest strategies and promoting the parties to upload true local embeddings during inferences.

2 RELATED WORK

Since the individual participants in FL usually have their own interests, the incentive and strategic problems in FL has been widely studied to facilitate the deployment of FL applications [18, 22, 40]. Most of the previous work study the methods to evaluate the contribution of participants as a reference for reward allocation or client selection [8, 25, 27, 32], incentivize the participants to keep active and dedicated in training [13, 31, 33, 41], as well as form coalitions to achieve better training performances for non-i.i.d. data [7, 9, 10]. As all the above work consider the incentive problem in the training stage of HFL or VFL, to the best of our knowledge, none of the existing work has considered the strategic behaviors of manipulating the intermediate local embeddings uploaded to the server during collaborative inferences in VFL system, which are orthogonal to the strategic behaviors in training stage.

In fact, the authors of [29] have proposed to utilize the same embedding manipulation approach from the perspective of a malicious attacker. Particularly, they investigate the set of adversarial dominating inputs (ADI) in inferences of VFL, such that the other party's influence on the inference result would be negligible. The attacks on FL system typically aim to replicate representative deep learning attacks for HFL, including [2, 4, 28, 34, 36, 39], while some other work attacking the VFL system exploit the characteristics of splitted model to infer the private features or labels of other participants [11, 19, 23, 24]. In opposed to these work that aim to protect VFL system against malicious attackers or honest-but-curious participants, our mechanism are designed for strategic participants with their own utility objectives, which provides a complementary perspective to protect the well-functionality of VFL system.

The main ideology of our distribution-based penalty mechanism is motivated by the linking mechanism [17] that restricts the total reported preferences of agents in a sequence of public decision problems to restrain the strategic behaviors. In detail, the agents are strictly limited in the frequency of reporting each preference across the problems. The research on linking mechanism is gradually progressed in terms of its additional properties, variants and applications [14, 26, 30, 35]. Compared with these work, due to the high-dimensional and complex nature of intermediate embeddings in VFL system, we could not learn the exact range and distribution of the high-dimensional embeddings, leading our mechanism and analysis to be distinct from the existing work.

3 PRELIMINARIES

Consider a typical VFL system with M parties[1] and a server, where the role of server could be assumed by one of the involved parties. The features are vertically distributed across the parties, with each party i privately owns c_i features of each sample. We use $x_i \in \mathbb{R}^{c_i}$ to denote the local features owned by party i for sample x. In order to realize the collaborative prediction for sample x, each party i holds a set of model parameters θ_i and a corresponding local embedding function $h_i(\cdot)$, which maps the model parameters and the input local sample features x_i to local embeddings. The server holds a set of model parameters θ_0 and a server prediction function $h_0(\cdot)$, which maps the server model parameters and all the uploaded local embeddings to a prediction in $\mathbb{R}$. As this work focuses on the strategic behaviors during the collaborative inference, we assume the embedding functions h_i and the server model h_0 to be some predetermined randomized functions, and would not go into details of the training and communication process.

With the above notations, the collaborative inference process for an unseen sample x could be formally described as follows: 1) each party i compute its local embedding $t_i := h_i(x_i)$ using the local features x_i; 2) each party i uploads t_i to the server; 3) the server computes $h_0(t_1, \ldots, t_M)$ using t_i; and 4) the prediction result $h_0(t_1, \ldots, t_M)$ is announced and takes effect. We use $\mathcal{T}_i$ to denote the space of potential local embeddings of party i, *i.e.*, $t_i \in \mathcal{T}_i$, and use $t := \prod_{i=1}^{M} t_i$ to denote the profile of local embeddings for all the parties. Similarly, we define the potential space of t as $\mathcal{T} := \prod_{i=1}^{M} \mathcal{T}_i$. We denote the distribution of local embeddings as $t \sim f$, and assume $t_i \sim f_i$ is independent with $t_j \sim f_j$ for $j \neq i$. Since the center has no control over the distributed local features, a party i might upload arbitrary local embeddings within $\mathcal{T}_i$ in collaborative inference. We use σ_i to denote the strategy of party i when uploading the embeddings, with $\sigma_i(t_i, t_i')$ characterizes the probability of uploading embedding t_i' when the true embedding is t_i under strategy σ_i, satisfying $\int_{t_i' \in \mathcal{T}_i} \sigma_i(t_i, t_i') dt_i' = 1, \forall t_i \in \mathcal{T}_i$. The strategy profile of all the parties is denoted as $\sigma := (\sigma_i)_{i=1}^{M}$, and the corresponding feasible space is denoted as $\Sigma := \prod_{i=1}^{M} \Sigma_i$. For convenience in notations, we would use subscript $-i$ to denote the embedding profile or its feasible space for all the parties except party i, *e.g.*, t_{-i} denotes the profile of uploaded local embeddings except party i, and $\mathcal{T}_{-i}$ denotes the corresponding feasible space of t_{-i}. We use I to denote the special *truth-telling* strategy, with $I(t_i, t_i') = 1$ when $t_i' = t_i$, and equals 0 otherwise.

Considering that the prediction result of the server would affect the utility of the involved parties, we use $u_i\left(h_0(t'); t_i\right)$[2] to denote the expected utility of party i when the uploaded embedding profile is t' and the true local embedding of party i is t_i. Therefore, the expected utility of party i when the strategy profile is σ and the server prediction function is h_0 could be calculated as

$$U_i^{h_0}(\sigma) = \int_{t \in \mathcal{T}} \int_{t' \in \mathcal{T}} u_i(h_0(t'); t_i) \sigma(t, t') dt' f(t) dt$$

with $\sigma(t, t') := \prod_{i=1}^{M} \sigma_i(t_i, t_i')$, and we may abbreviate h_0 when the context is clear.

In this work, we focus on the solution concept of Nash equilibrium to describe the stable state of strategic interactions between parties. In our context that each party only has incomplete information of the local embeddings, we consider a strategy profile

[1] We may use parties with participants interchangeably throughout the work, which also indicate the organizations in the federated recommendation game of Section 4.

[2] We assume the utility only depends on the prediction result and the party's own true local embeddings, since the party could not access the other party's true local embeddings, and have to rely on her local information to make decisions.

$\sigma = (\sigma_1, \ldots, \sigma_M)$ to be a Bayesian Nash equilibrium (BNE) if

$$U_i(\sigma) \geq U_i(\sigma_i', \sigma_{-i}), \forall \sigma_i' \in \Sigma_i, i \in [M].$$

In other words, when a strategy profile reaches BNE, none of the parties could achieve larger expected utility through changing her strategy, and we desire the truth-telling strategy $\sigma_I = (I)_{i=1}^M$, to be a BNE, such that the parties would be incentivized to upload the true local embeddings and the validity of prediction is preserved.

While studying the BNE requires us to make assumptions on distribution of embeddings, we may focus on one single round of inference to enable detailed analysis of the strategic interactions for specific embeddings (Section 4). Under this situation, when each party chooses a certain local embedding (instead of a distribution over potential embeddings) for uploading, and the uploaded embedding profile t' satisfies

$$u_i(h_0(t_i', t_{-i}'); t_i) \geq u_i(h_0(t_i'', t_{-i}'); t_i), \forall t_i'' \in \mathcal{T}_i, i \in [M],$$

then t' is called a pure-strategy Nash equilibrium (PNE). The PNE in each round of inference is a stronger equilibrium notion than the BNE over expectation of all the inference rounds, but is also more difficult and sometimes infeasible to achieve.

4 FEDERATED RECOMMENDATION GAME

Since our discussions until now stay on an abstract level, we would review a representative application of VFL system to illustrate the potential strategic behaviors of involved parties more concretely.

As briefly discussed in Section 1, an arising application of VFL is to aggregate user behavioral features from different organizations to provide better recommendation results for users [16, 38], which we term as federated recommendation. The strategic incentives of the organizations to manipulate the prediction results naturally arise here, as each organization prefers to display content beneficial for them. For example, some candidate items may originate from one of the organizations or contain content relevant to its business goal, which are more favorable for the organization to display.

To capture the key idea of this scenario, we assume each organization owns one unique item and aims to maximize the display probability for this item [3, 15] in federated recommendation, which is a moderate amplification of the competition faced by collaborating parties in practice. Following the literature studying games between content creators in recommendation system [15], we focus on the popular class of factorization-based recommendation algorithms. That is, each organization uploads the computed local user embedding t_i, and the server would use the product of the averaged user embedding and the item embedding b_i of the item owned by organization i as the matching score between the current user and item i. Suppose the server adopts a softmax policy of matching scores to display items, the expected utility (display probability) of each organization could then be calculated as

$$u_i\left(h_0(t')\right) = \frac{\exp(\tau^{-1} s_i)}{\sum_{j \in [M]} \exp(\tau^{-1} s_j)},$$

where $s_i := \langle b_i, \sum_i w_i t_i' \rangle$ is the predicted matching score between the item of party i and the current user, w_i denotes the aggregation weight for local embedding of party i, and $\tau > 0$ is the temperature parameter to control exploration in recommendation. Since this utility term does not depend on the true embeddings of parties, we drop t_i from the notation of u_i. We restrict $\|t_i\|_2 \leq 1$, or otherwise the party may report $\|t_i\|_2 \to \infty$ to increase its influence on the aggregated embedding. We use b_{ik} and t_{ik} to denote the k^{th} entry of b_i and t_i, respectively, and denote the number of dimensions of b_i, t_i as d.

In order to evaluate the consequences of strategic interactions for a specific profile of item and user embeddings, we would analyse the PNE resulted from the above utility function. If a PNE exists in the game and truth-telling does not constitute a PNE, then it indicates the parties would not conform to the truth-telling strategy, but would instead follow the behavior characterized by the PNE(s). Due to the limitation of space, the detailed proofs of our results in Section 4 and 5 are presented in Appendix A.

Theorem 4.1. *A PNE always exists in the federated recommendation game. Moreover, when $M = 2$, for any $b_1 \neq b_2$ and any positive weights, the unique PNE in the corresponding federated recommendation game is*

$$\forall k \in [d] : \; t_{1k} = -t_{2k} = \frac{b_{1k} - b_{2k}}{\left(\sum_{k'=1}^{d} (b_{1k'} - b_{2k'})^2\right)^{\frac{1}{2}}}, \tag{1}$$

Specifically, when $w_1 = w_2 = \frac{1}{2}$, the display probabilities would be $u_1 = u_2 = \frac{1}{2}$ for any item embeddings b.

In Theorem 4.1, the general PNE existence result is proved through showing the quasi-concavity of the utility function in the current setting and applying the Debreu-Glicksberg-Fan existence theorem. For the uniqueness result when $M = 2$, we first show any interior point with $\|t_i\| < 1$ could not be an equilibrium, then derive the detailed expressions of PNE through Karush–Kuhn–Tucker conditions. As we could observe from Theorem 4.1, despite the existence of PNE, its uniqueness when $M = 2$ suggests the failure of truth-telling strategy to be adopted by the organizations, and the resulted recommendation outcomes are significantly skewed by the utility-driven uploading of the involved parties. For the extreme case of $M = 2$ and $w_1 = w_2 = \frac{1}{2}$, the original properties of the user and item embeddings are completely disregarded, and the parties would get equal display chances for any user, leading the federated recommendation to lose its original design purpose.

While the federated recommendation system display poor performances against the parties' strategic behaviors, similar situations are not minority among the general VFL systems. Since the design philosophy of the VFL system is to aggregate valuable distributed features from each party to improve the prediction accuracy, the prediction results need to depend on the precise embeddings uploaded by the parties, and it is thus generally impossible for a standard VFL algorithm to prevent strategic manipulation in inferences. As the system designer, we should not assume that all the participants are disinterested with the prediction results and refrain from exploiting the vulnerabilities of the system, but instead need to establish effective manipulation-resistant mechanisms to mitigate the potential risks brought by strategic behaviors in collaborative inference.

5 METHODOLOGY

In this section, we would introduce our design of distribution-based penalty mechanism to prevent the strategic behaviors in collaborative inference, along with the corresponding design considerations.

5.1 Theoretical Basis

As indicated by the name of our mechanism, we adopt the distribution of uploaded local embeddings as the criteria to detect and penalize the strategic behaviors of involved parties. This choice is motivated by the traditional economics literature [17] on linking a sequence of public decision problems and restricting the number of reported preferences to overcome incentive issues.

Intuitively, monitoring the distribution of embeddings could effectively prevent the parties' strategic behaviors to always upload local embeddings from a specific set which are known to have higher probability of producing better inference results. To formalize the guarantees provided by constraining distribution of embeddings as suggested by this intuition, we require two additional conditions to facilitate rigorous theoretical proofs in our context: *independence* in distribution of local embeddings, and the standard *en-ante Pareto efficiency* [1] of the server prediction function.

Definition 5.1. *A server prediction function h_0 is ex-ante Pareto efficient for the utility functions $u = (u_i)_{i=1}^M$ and probability density functions f if there does not exist an alternative server prediction function h_0' such that*

$$\int_{t\in\mathcal{T}} u_i(h_0(t);t_i)f(t)dt \le \int_{t\in\mathcal{T}} u_i(h_0'(t);t_i)f(t)dt, \forall i,$$

or equivalently, $U_i^{h_0}(I^M) \le U_i^{h_0'}(I^M)$, and the inequality is strict for some i.

In plain words, a server prediction function satisfies ex-ante Pareto efficiency if there does not exist other server prediction function, such that every participant's expected utility under true local embeddings keeps non-decreasing, and at least one participant's expected utility strictly increases. Typical examples for ex-ante Pareto efficiency are the server prediction function always maximizes the sum of participants' utilities, or the server prediction function uniquely optimizes the utility function for one of the participants. As a more concrete example, in the federated recommendation scenario, as long as the recommendation system always allocate the full portion of display opportunities to the participants, then any server prediction function utilized during this allocating process would be ex-ante Pareto efficiency. This is because any server prediction function always trivially maximize the sum of utility of participants to be equal to one. As the distribution of uploaded local embeddings is a key measure for us, we define the marginal embedding distribution of a strategy σ_i as $f_i^{\sigma_i}$, with

$$f_i^{\sigma_i}(t_i') = \int_{t_i\in\mathcal{T}_i} \sigma_i(t_i,t_i')f_i(t_i)dt_i.$$

Theorem 5.2. *When each participant i's strategy is restricted to $\{\sigma_i : f_i^{\sigma_i} = f_i\}$ and the server prediction function h_0 is ex-ante Pareto efficient on u and f, then the truth-telling strategy $\{I^M\}$ is a BNE.*

By Theorem 5.2, under the independence and ex-ante Pareto efficiency conditions, if each participant's uploaded embeddings are strictly constrained to align with their prior distributions, then no participant could achieve higher utility through manipulating the uploaded embeddings when all the other participants adopt the truth-telling strategy. To prove Theorem 5.2, we note the independence in distributions of t_i would further indicate a relative independence in utility when the marginal distributions of all the parties are constrained to align with the prior distributions, *i.e.*, for any strategy profile σ satisfying $f_i = f_i^{\sigma_i}$ for each participant,

$$U_i(\sigma_i,\sigma_{-i}) = U_i(\sigma_i, I^{M-1}),\ \forall i.$$

As a result, when all the other participants adopt the truth-telling strategy, their expected utilities are guaranteed to keep stable regardless of the detailed reporting of a specific participant. If some participant i could realize a strict utility increment through changing her strategy, the ex-ante Pareto efficiency of the server prediction function would be broken, thus creates a contradiction.

Although strong guarantees of preventing strategic behaviors could be provided by the ex-ante Pareto efficiency, this condition might not always hold in reality. For example, when server prediction function is designed to optimize the accuracy of prediction and does not prioritize maximizing the utility function of involved parties, the condition of ex-ante Pareto efficiency would not hold. Therefore, we would like to investigate the guarantees that constraining $\{\sigma_i : f_i^{\sigma_i} = f_i\}$ could provide for more general server prediction functions. Since we are now under much weaker assumptions on h_0, we focus on the specific form of *linear utility* functions

$$u_i(h_0(t');t_i) = x_i^{h_0}(t')\cdot v_i(t_i),$$

where x_i is a function dependent on h_0. That is, we assume each prediction result t' brings $x_i^{h_0}(t')$ unit of valuable item (utility increase) to participant i, and the detailed amount of per-unit item value depends on participant i's true local embedding in the form $v_i(t_i)$. The linear utility function is widely-adopted in economics.

Theorem 5.3. *Assume that the distributions of local embeddings are discrete. For a strategic participant i with linear utility function, her utility could not be increased by using any $\sigma_i \ne I$ when each participant j's strategy is constrained within the range $\{\sigma_j : f_j^{\sigma_j} = f_j\}$, if $\forall t_i^1, t_i^2 \in \mathcal{T}_i$ with $v_i(t_i^1) \ge v_i(t_i^2)$,*

$$\mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i^1,t_{-i})] \ge \mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i^2,t_{-i})]. \tag{2}$$

To ensure the constraint on marginal distribution is sufficient to prevent dishonest utility increase, conditions (2) require a monotone property between the expected allocation $\mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i,t_{-i})]$ and the per-unit value $v_i(t_i)$ brought by a user with local embedding t_i. That is, a user (or other subject of prediction task) who would bring higher per-unit utility $v_i(t_i)$ for participant i, should simultaneously receive more expected allocation of items $\mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i,t_{-i})]$ after the overall evaluation $x_i^{h_0}$. Compared to the Pareto-efficiency condition, the monotone conditions (2) characterize another kind of coincidence between the server prediction function and the utility function of the participant. When conditions (2) hold for each participant i, Theorem 5.3 would provide the same BNE guarantee as in Theorem 5.2. We present Theorem 5.3 in the current form to emphasize its provided guarantees could be flexibly applied to each individual participant once the conditions hold, which keeps relevant independent with other participants in comparison to the ex-ante Pareto-efficiency condition in Theorem 5.2.

Algorithm 1: Distribution-Based Penalty Mechanism

Parameters: the test length m_i, n_i, and a non-decreasing penalty function $k_i : [0, 1] \to \mathbb{R}^+$;
Oracles: A valid two-sample test T_i for m_i uploaded embeddings and n_i training embeddings, and a data generator G_i approximates the distribution of (training) embeddings;

```
Initialize historical rejection rate of two-sample test q_i = 0;
Initialize a cache C_i;
while the collaborative inference is ongoing do
    Add each uploaded embedding to C_i, and use the current uploaded embedding for collaborative inference;
    if C_i is with length m_i then
        Apply T_i to embeddings in C_i and n_i random training embeddings;
        Clear C_i and update q_i;
        if T_i rejects the null hypothesis then
            for j = 1, ..., k_i(q_i) do
                Deactivate the embedding uploaded by participant i and use embeddings generated by G_i as substitute for each inference;
```

5.2 Distribution-Based Penalty Mechanism

Despite the promising guarantees provided by constraining the distributions of uploaded embeddings to align with its prior distribution, it is infeasible for the server to exactly implement this constraint for high-dimensional local embeddings uploaded by the parties. Although we could regard the distribution of training embeddings as an effective approximate to the prior distribution, enforcing the involved parties to upload embeddings exactly match with the training embeddings would lead the uploaded embeddings to be substantially different from the true local embeddings, and spoil the generalization capability of the VFL model. To adequately harness the efficacy of local embedding distributions in preventing strategic behaviors, we allow arbitrary local embeddings (within its domain) to be uploaded by the parties, and adopt additional design to reduce the incentives of conducting strategic behaviors through penalizing the problematic distributions.

The formal process of the distribution-based penalty mechanism is presented in Algorithm 1, which works individually for each party i. During the collaborative inference process, Algorithm 1 repeatedly collect embeddings uploaded by party i to distinguish whether a sequence of m_i uploaded embeddings comes from the same distribution of n_i (randomly sampled) training embeddings with high probability. We leverage corresponding methods in statistical literature to realize this detection task, technically termed as *two-sample tests*. If the null hypothesis that the two groups of samples come from the same distribution is rejected in the two-sample test for party i, this indicates party i has likely manipulated the uploaded local embeddings, and we would thus apply a penalty period to party i. During the penalty period, each uploaded embedding of participant i is deactivated and substituted with the random embeddings (output by a generator G_i) to eliminate her

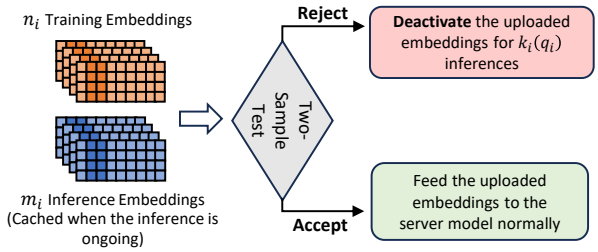


Figure 2: Illustration of Distribution-Based Penalty Mechanism for Party i

influences on the prediction results, and meanwhile leads her expected utility to decrease. The length of the current penalty period is calculated by the historical rejection rate q_i and the pre-designed penalty function k_i non-decreasing in q_i. In principle, we desire the generator G_i to approximate the prior distribution of party i's local embeddings, which could be realized by randomly drawing samples from the training embeddings.

As two-sample test is the key component to detect the consistency of distribution in our penalty mechanism, the analysis of our mechanism needs to depend on the properties of adopted two-sample tests. Formally speaking, given two groups of samples $X \sim p$ with size m and $Y \sim q$ with size n, a two-sample test is a statistical test $T(X, Y) : X^m \times X^n \mapsto \{0, 1\}$ to distinguish between the null hypothesis $\mathcal{H}_0 : p = q$ and the alternative hypothesis $\mathcal{H}_A : p \neq q$ [12]. Since the test is based on finite samples, it is possible that errors would be made for some situations. By convention, a type I error of a two-sample test occurs when the null hypothesis $p = q$ is wrongly rejected based on the observed samples, even though the data was generated with the same distribution. We define the type I error rate for a two-sample test T as α^T. Conversely, a type II error occurs when the null hypothesis $p = q$ is accepted on the observed samples, despite the fact $p \neq q$. We define the type II error rate of a two-sample test T against a specific $q \neq p$ as $\beta^T(q)$.

In our distribution-based penalty mechanism, we require the adopted two-sample test to satisfy $\beta^T(q) < \alpha^T$ for any $q \neq p$, *i.e.*, the acceptance rate of the null hypothesis is the highest when $q = p$ and be strictly smaller for $q \neq p$. Since this is a fundamental requirement for a well-functioning two-sample test, we call a two-sample test satisfying the above condition to be *valid*. Moreover, we also require each participant's expected utility to strictly decrease when her true local embeddings are substituted with random embeddings (drawn from her prior distribution), which is necessary to ensure the penalty period could effectively reduce the expected utility of a participant. We term the problem case (consist of u, f and h_0) satisfying this utility decrement condition for each party to be *feasible*, which could be verified through simulations in practice. When the above typical conditions hold, the proposed distribution-based penalty mechanism (Algorithm 1) is able to inherit the guarantees provided by strictly constraining the distribution (Section 5.1) in an asymptotic sense.

Theorem 5.4. *For any feasible problem case with valid two-sample tests T_i, suppose h_0 is ex-ante Pareto efficient on u and f, and $G_i \sim f_i$ for each participant i, then we could find some penalty functions $k_i(\cdot)$*

such that the expected per-round utility of truth-telling strategy $\{I^M\}$ converges to a BNE with the increase of inference rounds under the penalty-enabled server prediction function h_0^*.

Under the stated conditions, Theorem 5.4 guarantees no participant could obtain higher expected per-round utility than truth-telling as the inference proceeds, supposing all the other parties adopt the truth-telling strategy. The performance guarantee of our distribution-based penalty mechanism is established on the basis of results in Section 5.1. In principle, because no party could achieve larger utility through deviating to a strategy with the same marginal distribution (Theorem 5.2), the remaining chances to improve utility fall on the strategies with different marginal distributions. However, by the validity of two-sample tests, such kinds of strategies would result in larger historical rejection rate q_i and longer penalty period, thus also brings lower utility in the long term. Whilst Theorem 5.4 is formulated based on Theorem 5.2, an alternative result with the ex-ante Pareto efficiency condition replaced by conditions (2) could be formed based on Theorem 5.3. Though our theoretical results rely on conditions such as Pareto-efficiency and independent distribution, the principal idea of our design, *i.e.*, monitoring the distribution of uploaded embeddings, is broadly helpful in restraining the range of strategic behaviors in collaborative inference, even if the theoretical conditions are not strictly satisfied. This is also demonstrated by our experimental results in Section 6.

In both Algorithm 1 and Theorem 5.4, we do not characterize the detailed form of the penalty function k_i and the choice of sample length m_i, n_i in two-sample tests, but instead leave it flexible to accommodate the need of various scenarios. Choosing a penalty function $k_i(q_i)$ grows faster with q_i could provide stronger guarantee against strategic behaviors, but would simultaneously bring higher risk for honest parties when the number of conducted two-sample test is small and q_i has large variance. A similar tradeoff exists for the choice of test length m_i and n_i. While a larger m_i means lower error rate for two-sample tests, a smaller m_i allows to conduct more two-sample tests and get a stable q_i, which might be preferred when the number of total inference rounds is small. To choose appropriate mechanism parameters in practice, the server could conduct simulations on training embeddings to estimate the performances of the considered mechanism.

6 EXPERIMENTS

In the experiments, we aim to validate and investigate the following questions from an empirical view: (1) whether the considered strategic behaviors in collaborative inference are implementable and could bring the party substantially higher utility; (2) whether the proposed distribution-based penalty mechanism could effectively reduce the involved parties' incentives to conduct such strategic behaviors for practical datasets that not strictly satisfy the theoretical assumptions; and (3) how to set the parameters in the penalty mechanism to achieve good performances in practice.

6.1 Experimental Setup

Datasets and VFL Model We conduct experiments on two public datasets, *Criteo* and *Avazu*, with the task of click-through-rate (CTR) prediction. We assume there are two parties involved in VFL, with each party owning half of the features partitioned by their sequence in dataset. To validate the performances of our design for VFL models trained with different amount of data, we draw 1,000,000 samples to train and test the VFL model for Avazu, while the full dataset is available for Criteo. The training and testing sets are divided with proportion 9:1 for both the datasets. After the VFL model has been determined, we apply our mechanism on $N = 100,000$ samples (inference rounds) drawn from the testing set. We adopt the fully-connected neural network[3] (FCNN) for both the parties and the server, with 4 layers for the parties locally and 3 layers for the server, and the sparse features are first processed with an embedding layer before feeding into the local FCNN. The intermediate embeddings uploaded by each party are with dimension 40, which are concatenated to feed into the server network.

Strategic Settings We assume that there exists one strategic party in the system, which is without loss of generality as our mechanism works individually for each party. We consider the utility function of the strategic party to be the form $u_{exp} = \sum_{j\in[N]} ctr_j/N$ or $u_{click} = \sum_{j\in[N]}(ctr_j \cdot label_j)/N$, where ctr_j denotes the predicted CTR of the j^{th} test sample, and $label_j$ denotes its true label. Assuming that the display opportunity gained by the strategic party would be equal to the predicted CTR, these two utility functions represent the typical goal of obtaining more exposure opportunities and more expected clicks in recommendation.

Manipulation Strategies

- *Label-based* strategy: Considering that the local features of the training samples with positive label are likely to increase the prediction of CTR, the label-based strategy samples a local embedding from the training embeddings with positive labels to upload in each inference round. This label-based strategy is straightforward to implement in practice, which only requires the party to know a set of samples with positive labels.
- *Omniscient* strategy: In the omniscient strategy, we assume the strategy of the party is derived by optimizing the total predicted CTR under ℓ_2-regularization (applied to the difference between the original and manipulated embeddings), using the omniscient knowledge of local embeddings from both parties. To avoid the less meaningful case that the party extremely increases the scale of embeddings to dominantly create false-positive cases, we choose a regularization constant to ensure the resulting strategy achieves a sufficiently higher utility at an appropriate level. The optimization is performed using the stochastic gradient descent method.
- *Probabilistic Mixtures*: To validate our mechanisms against various potential strategies, we consider the *probabilistic mixtures* of the above two strategies with the true local embeddings, *e.g.*, a strategy with mixture probability 0.1 would report the true embeddings with 90% probability, and report according to the label-based (omniscient) strategy in the remaining 10% probability.

Mechanism Implementation When implementing Algorithm 1, we adopt the kernel two-sample test based on deep learning [21] with the test length for the training and inference embeddings set to be equal, *i.e.*, $n_i = m_i$. To reduce the variance of historical rejection rate q_i in implementation, we postpone all the penalties to the end of the inference stage, such that once the remaining

[3]Since the design of our mechanism only concerns the local embeddings uploaded by the participants, the detailed structure of VFL model would not induce great impacts on the trend of performances of the proposed mechanism.

Table 1: Original Utilities for Different Strategies

Criteo	True	Omniscient	Label-Based	Random
u_{exp}	0.2424	0.3591	0.3031	0.2345
u_{click}	0.1024	0.1372	0.1048	0.0836
Avazu	True	Omniscient	Label-Based	Random
u_{exp}	0.1389	0.2033	0.2761	0.1333
u_{click}	0.0392	0.0595	0.0478	0.0229

inference rounds are less than the total penalty length calculated by the latest rejection rate, the party would be penalized in all the remaining rounds. We choose penalty function in the linear form $k_i(q_i) = c \cdot n_i \cdot q_i$, *i.e.*, the penalty length for each rejection is the party's rejection rate times the current test length and a pre-determined penalty constant c. The detailed settings of the penalty constant c and the test length n_i in the mechanism would be characterized for each set of experiments.

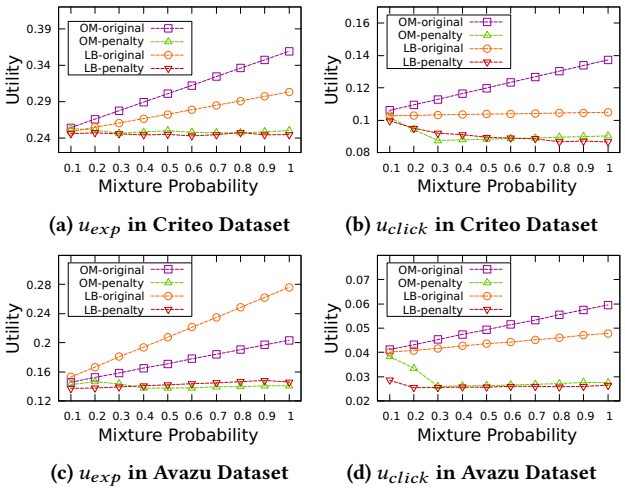
6.2 Experimental Results

Original Utilities of Different Strategies The original utilities of different strategies without the penalty mechanism are presented in Table 1. As we can observe, the omniscient strategy achieves evident higher utility on both the utility functions for two datasets, though it only applies small perturbations on true embeddings. The label-based strategy also achieves evident utility increase except for u_{click} in Criteo dataset. The reason might come from the limited influences of the party's local features on the prediction result and the relatively high proportion of positive samples in Criteo dataset.

Since we would substitute the party's original embeddings with randomly sampled training embeddings as penalty, we also validate the utility of such random "strategy" on the datasets. We find that the random strategy would lead to a lower utility for u_{ctr}, but achieve a similar (though still lower) utility for u_{click} compared with the true embeddings. In other words, using random strategy would not lead the total exposure of the party to decrease, despite the resulted mismatch between true label and predicted CTR. As a result, for parties with utility function u_{exp}, we could hardly reduce the utility obtained by strategic manipulations to be smaller than the utility obtained by true embeddings, and what we could achieve is to guide the two utilities to be close enough.

Performances of Distribution-Based Penalty Mechanism To observe the detailed performances of the proposed penalty mechanism, we conduct a set of experiments (Figure 3) that demonstrate the change in the utility of strategic party when the penalty mechanism is adopted. Due to the different characteristics of two datasets, we choose $n_i = 1800$ for Criteo dataset, and $n_i = 600$ for Avazu dataset, with both $c = 8$. As could be observed in Figure 3, the utilities obtained by different probabilistic mixtures of the omniscient (OM) and label-based (LB) strategies are largely reduced to be similar or less than the utility obtained by truthfully uploading the embeddings under the penalty mechanism, regardless of the original utilities obtained by those strategies, which demonstrates the effectiveness of our mechanism in diminishing the incentives of parties to adopt strategies other than truthful uploading.

In the trend of penalized utilities for OM and LB strategies on Avazu dataset, we could observe a slow utility growth after the



(a) u_{exp} in Criteo Dataset (b) u_{click} in Criteo Dataset

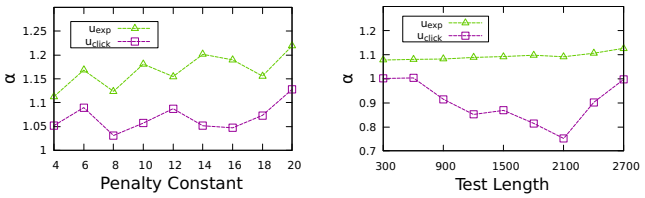
(c) u_{exp} in Avazu Dataset (d) u_{click} in Avazu Dataset

Figure 3: Comparison of Utilities obtained by Probabilistic Mixtures for Label-Based and Omniscient Strategies before and after the penalty mechanism is enabled, with $n_i = 1800$, $c = 8$ for Criteo Dataset, and $n_i = 600$, $c = 8$ for Avazu Dataset

mixture probability exceeds 0.4. This is because the mixture strategy at this point has been penalized in most of the inference rounds, and the utility increase comes entirely from the beginning inferences round for conducting the essential two-sample tests. For Criteo dataset, the utilities of different mixture strategies display slight fluctuations with the increase of mixture probability, which might due to the relatively high variance of two-sample tests under a smaller number of tests for Criteo.

Another important metric we need to observe is the total penalty length received by true embeddings, as immoderate penalty on a truthful party can significantly degrade the overall prediction performances of VFL system when it is not controlled at a relatively low level. For parameters adopted in Figure 3, the averaged penalty length received by true embeddings are 5,680 for Criteo dataset and 5,560 for the Avazu dataset, which is a small and acceptable penalty length compared to the 100,000 samples in total.

Influences of Mechanism Parameters The set of parameters we adopted in Figure 3 are actually not the deliberately fine-tuned ones to achieve the best performances. When testing the different mechanism parameters, we find a broad set of parameters could achieve satisfying effects around the parameters we present in Figure 3. Therefore, instead of presenting the similar performances achieved by successful mechanism parameters, we would like to demonstrate the importance of tailoring the mechanism parameters based on the characteristics of datasets and adopted two-sample tests. As a striking instance, simply applying a small test length to Criteo dataset and a large test length to Avazu dataset as opposed to Figure 3, *e.g.*, exchanging the n_i parameters for two datasets, would largely degrade the mechanism performances. To help with the evaluation of the mechanism's overall performances against the strategic behaviors, we define the metric of *utility approximate ratio* α to be the largest ratio between the utility obtained by a dishonest strategy and the utility of true embedding among all the considered strategies in Figure 3. We regard an utility approximate ratio less



(a) $n_i = 600$ **with Different Penalty Constant** $c \in [4, 20]$ **in Criteo Dataset**

(b) $c = 8$ **with Different Test Length** $n_i \in [300, 2700]$ **in Avazu Dataset**

Figure 4: Utility Approximate Ratio α for Failure Cases in Criteo and Avazu Dataset

than 1.1 to be acceptable for u_{exp} and 1.05 to be acceptable for u_{click}, which are both satisfied by the experiments in Figure 3.

In Figure 4a, we report the utility approximate ratios of test length $n_i = 600$ with different penalty constant $c \in [4, 20]$ on Criteo dataset. We can observe that the utility approximate ratio fails to satisfy the required standard in most cases and gradually increases with the penalty constant, though with fluctuations. This is caused by the poor performances of two-sample test with $n_i = 600$ when distinguishing the variants of LB strategies on Criteo dataset, such that both the true embeddings and the LB variants receives small penalties. In Figure 4b, we report the utility approximate ratios of $c = 8$ with different test lengths $n_i \in [300, 2700]$ on Avazu dataset. Despite the seemingly satisfying results of α, the total penalty received by the true embeddings has generally exceeded 15,000 rounds after $n_i \geq 1500$, and the relatively low α comes from applying large penalties for both the truth-telling strategy and alternative strategies. For its potential causes, the adopted two-sample test might have relatively high variances on the Avazu dataset and requires more tests to make q_i stable when computing the penalty length, which could not be provided by $n_i \geq 1500$ under the current number of inference samples.

Simulations of Multi-Party Setup Since our mechanism works independently for each party irrespective of other parties' behaviors, we can simulate the performances of our mechanism for the multi-party setup under the two-party setup. In detail, for any party in multi-party setup, we could regard all the other parties as a "giant" party and run our mechanism only on the local embeddings uploaded by the considered party. Based on this equivalence property, we simulate the 3-party and 4-party setup on Avazu dataset with two parties, by assuming the strategic party owning (approximately) 1/3 and 1/4 of the features. We adopt the same mechanism parameters as in Figure 3c and 3d, and the results are presented in Table 2. We could observe a significant decrease in the utilities of the OM and LB strategies under our penalty mechanism (OM-P and LB-P) compared to their utilities in the vanilla VFL system. In contrast, the utilities of uploading true local embeddings remained largely unaffected under the penalty mechanism (True and True-P), demonstrating the efficacy of our mechanism in various VFL settings. The discrepancies between utilities obtained by OM and LB strategies in Table 1 and 2 are likely due to the differences in the specific features owned by the strategic party and their varying significance in affecting the final prediction result, and it is not necessarily the case that owning more features would enable the party to achieve larger dishonest utility increase when similar manipulation strategies are adopted (Table 2).

Table 2: Changes in utilities of different strategies when the penalty mechanism is enabled and the strategic party owns 1/3 and 1/4 of the features for Avazu dataset

1/3 features	True	OM	LB	True-P	OM-P	LB-P
u_{exp}	0.1345	0.1449	0.1483	0.1343	0.1278	0.1283
u_{click}	0.0377	0.0420	0.0371	0.0375	0.0328	0.0323
1/4 features	True	OM	LB	True-P	OM-P	LB-P
u_{exp}	0.1390	0.1529	0.1542	0.1389	0.1349	0.1351
u_{click}	0.0376	0.0431	0.0369	0.0376	0.0333	0.0326

7 CONCLUSION

In this work, we consider the strategic behaviors in collaborative inference for vertical federated learning, where the parties could manipulate the uploaded local embeddings to change the inference results and maximize their own utilities. We model the strategic interactions between parties for a representative federated recommendation application, and our analysis reveals the adverse effects of the considered strategic behaviors. Specifically, we propose a class of distribution-based penalty mechanism to prevent such strategic behaviors. The proposed mechanism works through applying statistical two-sample tests to distinguish the deviation in embedding distributions and penalizing the parties based on the test results, whose performance is theoretically demonstrated. The experimental results validate the effectiveness of the proposed mechanism in terms of preventing the considered strategic behaviors.

ACKNOWLEDGEMENT

This work was supported in part by National Key R&D Program of China (No. 2022ZD0119100), in part by China NSF grant No. 62322206, 62132018, U2268204, 62025204, 62272307, 62372296. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

REFERENCES

[1] Kenneth J Arrow. 2012. *Social choice and individual values*. Vol. 12. Yale University Press.
[2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*. PMLR, 2938–2948.
[3] Omer Ben-Porat and Moshe Tennenholtz. 2018. A game-theoretic approach to recommendation systems with strategic content providers. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 1118–1128.
[4] Arjun Nitin Bhagoji, Supriyo Chakraborty, Prateek Mittal, and Seraphin Calo. 2019. Analyzing federated learning through an adversarial lens. In *Proceedings of the 36th International Conference on Machine Learning*. PMLR, 634–643.
[5] Chaochao Chen, Jun Zhou, Li Wang, Xibin Wu, Wenjing Fang, Jin Tan, Lei Wang, Alex X Liu, Hao Wang, and Cheng Hong. 2021. When homomorphic encryption marries secret sharing: Secure large-scale sparse logistic regression and applications in risk control. In *Proceedings of the 27th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2652–2662.
[6] Yong Cheng, Yang Liu, Tianjian Chen, and Qiang Yang. 2020. Federated learning for privacy-preserving AI. *Commun. ACM* 63, 12 (2020), 33–36.
[7] Sen Cui, Jian Liang, Weishen Pan, Kun Chen, Changshui Zhang, and Fei Wang. 2022. Collaboration Equilibrium in Federated Learning. In *Proceedings of the 28th*

ACM SIGKDD Conference on Knowledge Discovery & Data Mining. ACM, New York, NY, USA, 241–251.
[8] Yongheng Deng, Feng Lyu, Ju Ren, Yi-Chao Chen, Peng Yang, Yuezhi Zhou, and Yaoxue Zhang. 2021. FAIR: Quality-aware federated learning with precise user incentive and model aggregation. In *Proceedings of the 40th IEEE Conference on Computer Communications*. IEEE, 1–10.
[9] Kate Donahue and Jon Kleinberg. 2021. Model-sharing games: Analyzing federated learning under voluntary participation. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI Press, 5303–5311.
[10] Kate Donahue and Jon Kleinberg. 2021. Optimality and stability in federated learning: A game-theoretic approach. In *Proceedings of the 35th International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 1287–1298.
[11] Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X Liu, and Ting Wang. 2022. Label inference attacks against vertical federated learning. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, 1397–1414.
[12] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research* 13 (2012), 723–773.
[13] Jingoo Han, Ahmad Faraz Khan, Syed Zawad, Ali Anwar, Nathalie Baracaldo, Yi Zhou, Feng Yan, and Ali Raza Butt. 2022. TIFF: Tokenized Incentive for Federated Learning. In *Proceedings of the IEEE 15th International Conference on Cloud Computing*. IEEE, 407–416.
[14] Rafael Hortala-Vallve. 2010. Inefficiencies on linking decisions. *Social Choice and Welfare* 34, 3 (2010), 471–486.
[15] Jiri Hron, Karl Krauth, Michael Jordan, Niki Kilbertus, and Sarah Dean. 2022. Modeling content creator incentives on algorithm-curated platforms. In *Proceedings of the 11th International Conference on Learning Representations*.
[16] Yaochen Hu, Di Niu, Jianming Yang, and Shengping Zhou. 2019. FDML: A collaborative machine learning framework for distributed features. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2232–2240.
[17] Matthew O Jackson and Hugo F Sonnenschein. 2007. Overcoming incentive constraints by linking decisions. *Econometrica* 75, 1 (2007), 241–257.
[18] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning* 14, 1–2 (2021), 1–210.
[19] Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. 2022. Label Leakage and Protection in Two-party Split Learning. In *Proceedings of the 10th International Conference on Learning Representations*.
[20] Wenjie Li, Qiaolin Xia, Hao Cheng, Kouyin Xue, and Shu-Tao Xia. 2022. Vertical semi-federated learning for efficient online advertising. *arXiv preprint* (2022). https://arxiv.org/abs/2209.15635
[21] Feng Liu, Wenkai Xu, Jie Lu, Guangquan Zhang, Arthur Gretton, and Danica J. Sutherland. 2020. Learning Deep Kernels for Non-Parametric Two-Sample Tests. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 6316–6326.
[22] Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. 2024. Vertical Federated Learning: Concepts, Advances, and Challenges. *IEEE Transactions on Knowledge and Data Engineering* 36, 7 (2024), 3615–3634.
[23] Yang Liu, Zhihao Yi, and Tianjian Chen. 2020. Backdoor attacks and defenses in feature-partitioned collaborative learning. *arXiv preprint* (2020). https://arxiv.org/abs/2007.03608
[24] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. 2021. Feature inference attack on model predictions in vertical federated learning. In *Proceedings of the 37th IEEE International Conference on Data Engineering*. IEEE, 181–192.
[25] Hongtao Lv, Zhenzhe Zheng, Tie Luo, Fan Wu, Shaojie Tang, Lifeng Hua, Rongfei Jia, and Chengfei Lv. 2021. Data-free evaluation of user contributions in federated learning. In *19th International Symposium on Modeling and Optimization in Mobile, Ad hoc, and Wireless Networks*. IFIP, 81–88.
[26] Hitoshi Matsushima, Koichi Miyazaki, and Nobuyuki Yagi. 2010. Role of linking mechanisms in multitask agency with hidden information. *Journal of Economic Theory* 145, 6 (2010), 2241–2259.
[27] Lokesh Nagalapatti and Ramasuri Narayanam. 2021. Game of gradients: Mitigating irrelevant clients in federated learning. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*. AAAI Press, 9046–9054.
[28] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *2019 IEEE Symposium on Security and Privacy*. IEEE, 739–753.
[29] Qi Pang, Yuanyuan Yuan, Shuai Wang, and Wenting Zheng. 2023. ADI: Adversarial Dominating Inputs in Vertical Federated Learning Systems. In *2023 IEEE Symposium on Security and Privacy*. IEEE Computer Society, Los Alamitos, CA, USA, 1875–1892.
[30] Agustín Santos, Antonio Fernández Anta, José A Cuesta, and Luis López Fernández. 2016. Fair linking mechanisms for resource allocation with correlated player types. *Computing* 98 (2016), 777–801.
[31] Rachael Hwee Ling Sim, Yehong Zhang, Mun Choon Chan, and Bryan Kian Hsiang Low. 2020. Collaborative machine learning with incentive-aware model rewards. In *Proceedings of the 37th International Conference on Machine Learning*. PMLR, 8927–8936.
[32] Behnaz Soltani, Yipeng Zhou, Venus Haghighi, and John C. S. Lui. 2023. A Survey of Federated Evaluation in Federated Learning. In *Proceedings of the 32th International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence Organization, 6769–6777.
[33] Ming Tang and Vincent WS Wong. 2021. An incentive mechanism for cross-silo federated learning: A public goods perspective. In *Proceedings of the 40th IEEE Conference on Computer Communications*. IEEE, 1–10.
[34] Vale Tolpegin, Stacey Truex, Mehmet Emre Gursoy, and Ling Liu. 2020. Data poisoning attacks against federated learning systems. In *25th European Symposium on Research in Computer Security*. Springer, 480–501.
[35] Róbert F Veszteg. 2015. Linking decisions with standardization. *Studies in Microeconomics* 3, 1 (2015), 35–48.
[36] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. 2020. Attack of the tails: Yes, you really can backdoor federated learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*. Curran Associates Inc., Red Hook, NY, USA, 16070–16084.
[37] Penghui Wei, Hongjian Dou, Shaoguo Liu, Rongjun Tang, Li Liu, Liang Wang, and Bo Zheng. 2023. FedAds: A Benchmark for Privacy-Preserving CVR Estimation with Vertical Federated Learning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, 3037–3046.
[38] Chuhan Wu, Fangzhao Wu, Lingjuan Lyu, Yongfeng Huang, and Xing Xie. 2022. FedCTR: Federated Native Ad CTR Prediction with Cross-platform User Behavior Data. *ACM Transactions on Intelligent Systems and Technology* 13, 4 (2022), 62:1–62:19.
[39] Chulin Xie, Keli Huang, Pin Yu Chen, and Bo Li. 2020. DBA: Distributed Backdoor Attacks against Federated Learning. In *Proceedings of the 8th International Conference on Learning Representations*.
[40] Yufeng Zhan, Jie Zhang, Zicong Hong, Leijie Wu, Peng Li, and Song Guo. 2022. A survey of incentive mechanism design for federated learning. *IEEE Transactions on Emerging Topics in Computing* 10, 2 (2022), 1035–1044.
[41] Meng Zhang, Ermin Wei, and Randall Berry. 2021. Faithful edge federated learning: Scalability and privacy. *IEEE Journal on Selected Areas in Communications* 39, 12 (2021), 3790–3804.

A PROOFS OF RESULTS

For convenience in notations, we may abbreviate the server prediction function h_0 in notations when the context is clear.

A.1 Proof for Theorem 4.1

Proof. (Existence of PNE) To show the general existence of PNE, we apply the Debreu-Glicksberg-Fan PNE existence theorem. That is, a PNE exists in a game if the following conditions are satisfied: (1) the strategy space of each player is compact and convex; and (2) the utility function of each player is continuous and quasi-concave in her strategy. By our definition of t_i, it is clear the strategy space is compact and convex, and $u_i(t_i, t_{-i})$ is continuous in t_i. It remains to demonstrate $u_i(t_i, t_{-i})$ is quasi-concave in t_i.

Since $u_i(t_i, t_{-i})$ could be fully defined by $s = (s_1, s_2, \ldots, s_M)$, and s could be obtained by a linear transformation from $t_i = (t_{i1}, t_{i2}, \ldots, t_{id})$ regarding t_{-i} as constants, we only need to show $u_i(t_i, t_{-i})$ being quasi-concave on s. Recall

$$u_i(t_i, t_{-i}) = \frac{\exp(\tau^{-1} s_i)}{\sum_{j\in[M]} \exp(\tau^{-1} s_j)},$$

suppose we have two points s and s', such that $u_i(s) \geq a$ and $u_i(s') \geq a$. By definition of u_i, we would have $\exp(\tau^{-1} s_i) \geq \frac{a}{1-a} \cdot \left(\sum_{j\neq i} \exp(\tau^{-1} s_j)\right)$. Taking logarithm on both sides, we have $\tau^{-1} s_i \geq \ln \frac{a}{1-a} + \ln \left(\sum_{j\neq i} \exp(\tau^{-1} s_j)\right)$. Similarly, $\tau^{-1} s_i' \geq \ln \frac{a}{1-a} +$

$\ln\left(\sum_{j\neq i}\exp(\tau^{-1}s'_j)\right)$. For any $\lambda \in (0,1)$, we could thus obtain $\tau^{-1}(\lambda s_i + (1-\lambda)s'_i) \geq \ln\frac{a}{1-a} + \lambda\ln\left(\sum_{j\neq i}\exp(\tau^{-1}s_j)\right) + (1-\lambda)\ln\left(\sum_{j\neq i}\exp(\tau^{-1}s'_j)\right)$. On the other hand, by Holder's inequality,

$$\ln\left(\left(\sum_{j\neq i}\exp(\tau^{-1}s_j)\right)^{\lambda}\cdot\left(\sum_{j\neq i}\exp(\tau^{-1}s'_j)\right)^{1-\lambda}\right)$$
$$\geq \ln\left(\sum_{j\neq i}\exp\left(\tau^{-1}(\lambda s_j + (1-\lambda)s'_j)\right)\right),$$

Therefore, for $\bar{s} = \lambda s + (1-\lambda)s'$, $\tau^{-1}\bar{s}_i \geq \ln\frac{a}{1-a} + \ln\left(\sum_{j\neq i}\exp\left(\tau^{-1}\bar{s}_i\right)\right)$, which indicates $u_i(\bar{s}) \geq a$ and the quasi-concavity of $u_i(s)$. □

Proof. (Unique PNE when $M=2$) We would finish the proof by showing the following statements: When $n=2$, (1) no interior point of t_1 or t_2 would be a PNE, *i.e.*, $\|t_1\| = \|t_2\| = 1$; and (2) fix any t_{3-i}, the best response t_i must be obtained by scaling vector $\left(b_{ik} - b_{(3-i)k}\right)_{k\in[d]}$ with a constant. Note that party $3-i$ denotes the other party besides i. Fixing t_{-i}, the optimization problem faced by party i could be formulated as

$$\begin{aligned}\max_{t_i}\quad & f_i(t_i) = \frac{\exp(\tau^{-1}s_i)}{\sum_{j\in[n]}\exp(\tau^{-1}s_j)},\\ \text{s.t.}\quad & s_j = \sum_{k=1}^{d}\sum_{i'=1}^{n} t_{i'k}w_{i'}b_{jk}, \forall j\in[n],\\ & \sum_{k=1}^{d} t_{ik}^2 \leq 1.\end{aligned} \tag{3}$$

Using multi-variable chain rule, we could obtain

$$\begin{aligned}\frac{\partial f_i(t_i)}{\partial t_{ik}} &= \frac{\partial f_i(t_i)}{\partial s_i}\cdot\frac{\partial s_i}{\partial t_{ik}} + \sum_{j\neq i}\frac{\partial f_i(t_i)}{\partial s_j}\cdot\frac{\partial s_j}{\partial t_{ik}}\\ &= \frac{w_i\exp(\tau^{-1}s_i)}{\tau\left(\sum_{j'\in[n]}\exp(\tau^{-1}s'_j)\right)^2}\cdot\sum_{j\neq i}\exp(\tau^{-1}s_j)(b_{ik}-b_{jk}).\end{aligned}$$

By definition of PNE, t_i must be the solution to problem (3) supposing t_{-i} are fixed. To characterize the conditions of those best-response t_i, consider the following Karush–Kuhn–Tucker (KKT) conditions induced by problem (3).

$$\begin{cases}\frac{\partial f_i(t_i)}{\partial t_{ik}} + 2\lambda t_{ik} = 0, \quad \forall k\in[d]\\ \lambda\left(\sum_{k=1}^{d} t_{ik}^2 - 1\right) = 0\\ \sum_{k=1}^{d} t_{ik}^2 \leq 1\\ \lambda \leq 0\end{cases} \tag{4}$$

We would now claim that $\lambda \neq 0$ for $n=2$ and $b_1 \neq b_2$. If $\lambda = 0$, we require $\frac{\partial f_i(t_i)}{\partial t_{ik}} = 0$, $\forall k\in[d]$. However, when $n=2$, the term $\sum_{j\neq i}\exp(\tau^{-1}s_j)(b_{ik}-b_{jk})$ would reduce to $\exp(\tau^{-1}s_{3-i})(b_{ik}-b_{(3-i)k})$, which could not be 0 for every k for $b_1 \neq b_2$. As the term $\frac{w_i\exp(\tau^{-1}s_i)}{\tau\left(\sum_{j'\in[n]}\exp(\tau^{-1}s'_j)\right)^2}$ is strictly positive, $\frac{\partial f_i(t_i)}{\partial t_{ik}} = 0$ could not hold for every k, thus λ could not be 0, and we must have $\sum_{k=1}^{d} t_{ik}^2 = 1$ to satisfy conditions (4). As $\lambda \neq 0$, we could represent each entry of t_i as $t_{ik} = C_i\cdot\sum_{j\neq i}\exp(\tau^{-1}s_j)(b_{ik}-b_{jk})$ where $C_i = -\frac{1}{2\lambda}\cdot\frac{w_i\exp(\tau^{-1}s_i)}{\tau\left(\sum_{j'\in[n]}\exp(\tau^{-1}s'_j)\right)^2} > 0$ is the same constant for all t_{ik}. When $n=2$, we further denote $C'_i = C_i\cdot\exp(\tau^{-1}s_{3-i})$, then we must have $t_{ik} = C'_i\cdot(b_{ik}-b_{(3-i)k})$. Since we have derived $\sum_{k=1}^{d} t_{ik}^2 = 1$ and $C'_i > 0$, we could thus deduce

$$t_{ik} = \frac{b_{ik}-b_{(3-i)k}}{\left(\sum_{k'=1}^{d}(b_{ik'}-b_{(3-i)k'})^2\right)^{\frac{1}{2}}}$$

is the unique solution (best-response strategy) to problem (3). Combining the unique best-response strategies for both party 1 and party 2 finishes the proof for our statement. □

A.2 Proof for Theorem 5.2

Lemma A.1. *For any strategy profile σ satisfying $f_i = f_i^{\sigma_i}$, $\forall i$,*

$$U_i(\sigma_i,\sigma_{-i}) = U_i(\sigma_i, I^{M-1})$$

Proof. (Lemma A.1) Note that

$$U_i(\sigma_i, I^{M-1}) = \int_{t\in\mathcal{T}}\int_{t'_i\in\mathcal{T}_i} u_i\left((t'_i,t_{-i});t_i\right)\sigma_i(t_i,t'_i)dt'_i f(t)dt,$$

we could have

$$\begin{aligned}U_i(\sigma_i,\sigma_{-i}) &= \int_{t\in\mathcal{T}}\int_{t'\in\mathcal{T}} u_i(t';t_i)\prod_{i=1}^{M}\sigma_i(t_i,t'_i)dt' f(t)dt\\ &= \int_{t'_1\in\mathcal{T}_1}\cdots\int_{t'_M\in\mathcal{T}_M}\int_{t_i\in\mathcal{T}_i} u_i(t';t_i)\sigma_i(t_i,t'_i)f_i(t_i)\\ &\qquad\prod_{j\neq i}\left(\int_{t_j\in\mathcal{T}_j}\sigma_j(t_j,t'_j)f_j(t_j)dt_j\right)dt_i dt'_1\ldots dt'_M\\ &= \int_{t'_1\in\mathcal{T}_1}\cdots\int_{t'_M\in\mathcal{T}_M}\int_{t_i\in\mathcal{T}_i} u_i(t';t_i)\sigma_i(t_i,t'_i)f_i(t_i)\\ &\qquad\prod_{j\neq i} f_j(t'_j)dt_i dt'_1\ldots dt'_M\\ &= U_i(\sigma_i, I^{M-1}),\end{aligned}$$

where the last equality comes from regarding t'_j as t_j for $j\neq i$. □

Proof. (Theorem 5.2) For the strategy profile $\sigma = \{I^M\}$, suppose participant i deviates to an alternative strategy σ'_i with $f_i^{\sigma'_i} = f_i$, and $U_i^{h_0}(\sigma'_i, I^{M-1}) > U_i^{h_0}(I^M)$. Then consider another (randomized) server aggregation function h'_0 defined as $h'_0(t) = h_0(\sigma'_i(t_i), t_{-i})$. Since h'_0 just maps the uploaded embedding of participant i according to σ'_i on the basis of h_0, $U_j^{h'_0}(I^M) = U_j^{h_0}(I,(\sigma'_i, I^{M-2}))$ for $j\neq i$, and $U_i^{h'_0}(I^M) = U_i^{h_0}(\sigma'_i, I^{M-1})$.

From Lemma A.1, as $f_j^I = f_j$ and $f_i^{\sigma'_i} = f_i$, we have

$$U_j^{h_0}(I,(\sigma'_i,I^{M-2})) = U_j^{h_0}(I^M) = \int_{t\in\mathcal{T}} u_j(h_0(t);t_j)f(t)dt,\ \forall j\neq i.$$

Therefore, we would have $U_i^{h'_0}(I^M) > U_i^{h_0}(I^M)$ and $U_j^{h'_0}(I^M) = U_j^{h_0}(I^M), \forall j\neq i$, contradicting with the Pareto efficiency of h_0. □

A.3 Proof for Theorem 5.3

Proof. We conduct the proofs for discrete embeddings $t_i \in T_i$. For any potential strategy σ_i, construct a directed graph $\mathcal{G}$, whose nodes correspond to each possible local embedding $t_i \in T_i$. We construct the edges in this graph to denote the strategic report $\sigma_i(t_i^1, t_i^2) > 0$ for $t_i^1 \neq t_i^2$, such that each directed edge pointing from t_i^1 to t_i^2 has weight $\mathbb{P}(t_i = t_i^1) \cdot \sigma_i(t_i^1, t_i^2)$, *i.e.*, the probability that participant i has true embedding t_i^1 and misreport embedding t_i^2 under strategy σ_i. By $\{\sigma_i : f_i^{\sigma_i} = f_i\}$, we require $\forall t_i^2 \in \mathcal{T}_i$,

$$\sum_{t_i^1 \in \mathcal{T}_i} \sigma_i(t_i^1, t_i^2) \cdot \mathbb{P}(t_i = t_i^1) = \mathbb{P}(t_i = t_i^2). \tag{5}$$

Based on (5), we would further have

$$\begin{aligned}\sum_{t_i^1 \neq t_i^2} \sigma_i(t_i^1, t_i^2) \cdot \mathbb{P}(t_i = t_i^1) &= (1 - \sigma_i(t_i^2, t_i^2)) \cdot \mathbb{P}(t_i = t_i^2) \\ &= \sum_{t_i^1 \neq t_i^2} \sigma_i(t_i^2, t_i^1) \cdot \mathbb{P}(t_i = t_i^2),\end{aligned}$$

where the last inequality is due to $\sum_{t_i^1 \in \mathcal{T}_i} \sigma_i(t_i^2, t_i^1) = 1,\ \forall t_i^2 \in \mathcal{T}_i$. Therefore, in the constructed graph, each node would have the sum of weight of in-edges to equal the sum of weight of out-edges.

To prove the statement, we would start from the graph of any strategy $\sigma_i \neq I$, and gradually remove all the edges in this graph to approach the edgeless graph (correspond to the truth-telling strategy I). We would demonstrate that (1) the adjustment must finally lead to an edgeless graph, and (2) each step in the adjustment would result in a feasible strategy with non-decreasing utility, thus finishes the proof. Our adjustment is as follows: in each step, we find a cycle in the graph. We remove the edge with the smallest weight in this cycle, whose weight is denoted as w, and also update the weight of other edges in this cycle to minus w, then add w to $\sigma_i(t_i, t_i)$ for each node t_i in this cycle.

To prove statement (1), since we have formulated a directed graph, suppose the graph is not edgeless and contains no cycle during the adjustment, there must exist some sink node and source node in the graph. However, by our construction, each node must have the equal sum of weights of in-edges and out-edges, thus leads to a contradiction. For statement (2), since each step of adjustment preserves Equation (5), the adjusted strategy is still feasible and satisfies $\{\sigma_i : f_i^{\sigma_i} = f_i\}$. We only remains to show each step of adjustment would lead to non-decreasing utility given conditions (2). For a strategy σ_i, its resulted utility could be calculated as

$$\begin{aligned}&\sum_{t_i^1 \in \mathcal{T}_i} \sum_{t_i^2 \in \mathcal{T}_i} \mathbb{P}(t_i = t_i^1) \cdot \sigma_i(t_i^1, t_i^2) \cdot v_i(t_i^1) \cdot \mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i^2, t_{-i})] \\ &\quad = \sum_{t_i^1 \in \mathcal{T}_i} \sum_{t_i^2 \in \mathcal{T}_i} w(t_i^1, t_i^2) \cdot v_i(t_i^1) \cdot \mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i^2, t_{-i})],\end{aligned}$$

where we use $w(t_i^1, t_i^2)$ to denote the weight of edge from t_i^1 to t_i^2 for $t_i^1 \neq t_i^2$, and define $w(t_i^1, t_i^1) = \mathbb{P}(t_i = t_i^1) \cdot \sigma_i(t_i^1, t_i^1)$. W.l.o.g., define the n nodes in the current cycle as $t_i^1, \dots t_i^n$ (with $t_i^{n+1} = t_i^1$ for convenience in notations). Then by our construction of the adjustment, except for the unchanged parts of the graph, the utility before this step of adjustment is $w \cdot \left[\sum_{j=1}^n v_i(t_i^j) \cdot \mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i^{j+1}, t_{-i})]\right]$, and the utility after this step would be $w \cdot \left[\sum_{j=1}^n v_i(t_i^j) \cdot \mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i^j, t_{-i})]\right]$. By conditions (2), since larger $v_i(t_i^j)$ implies larger $\mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i^j, t_{-i})]$, applying the rearrangement inequality, we would have

$$\sum_{j=1}^n v_i(t_i^j) \cdot \mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i^j, t_{-i})] \geq \sum_{j=1}^n v_i(t_i^j) \cdot \mathbb{E}_{t_{-i}}[x_i^{h_0}(t_i^{j+1}, t_{-i})],$$

thus proves the non-decreasing of utility in each step of adjustment. □

A.4 Proof for Theorem 5.4

Proof. Define n to be the number of total inference rounds. It is sufficient for us to find a penalty function $k_i(\cdot)$ for each participant i, such that when $n \to \infty$, there does not exist a strategy σ_i with $U_i^{h_0^*}(\sigma_i, I^{M-1}) > U_i^{h_0^*}(I^M)$. For convenience in notations, we abbreviate $\beta^T(f_i^{\sigma_i})$ to be $\beta(\sigma_i)$, and define $k_i'(\beta(\sigma_i)) = k_i(\beta(\sigma_i))/m_i$.

To evaluate the change of utility for an arbitrary strategy σ_i during the penalty period, we further define the expected utility increment ratio of σ_i over the penalty reporting strategy σ_i^p : $\sigma_i^p(t_i, \cdot) \sim G_i,\ \forall t_i \in \mathcal{T}_i$ when other participants report truthfully as $\delta_i(\sigma_i) := \frac{U_i(\sigma_i, I^{M-1})}{U_i(\sigma_i^p, I^{M-1})} - 1$. Recall that we are under the feasible problem case, which indicates $\delta_i(I) > 0$. By the strong law of large number, we would have $q_i(\sigma_i)$ converges to $\beta(\sigma_i)$ when $n \to \infty$. Therefore, the proportion of penalty period among the entire inference period would converge to $\frac{\beta(\sigma_i) \cdot k_i'(\beta(\sigma_i))}{1 + \beta(\sigma_i) \cdot k_i'(\beta(\sigma_i))}$, since when each two-sample test of length m_i ends, there is an additional $\beta(\sigma_i)$ probability to have a penalty period with length $k_i'(\beta(\sigma_i)) \cdot m_i$. Thus, we could calculate the expected per-round utility of strategy σ_i when $n \to \infty$ as

$$\begin{aligned}U_i^{h_0^*}(\sigma_i, I^{M-1}) &= \frac{1}{1 + \beta(\sigma_i) \cdot k_i'(\beta(\sigma_i))} \cdot U_i^{h_0}(\sigma_i, I^{M-1}) \\ &\quad + \frac{\beta(\sigma_i) \cdot k_i'(\beta(\sigma_i))}{1 + \beta(\sigma_i) \cdot k_i'(\beta(\sigma_i))} \cdot U_i^{h_0}(\sigma_i^p, I^{M-1}) \\ &= \left(\frac{\delta_i(\sigma_i)}{1 + \beta(\sigma_i) \cdot k_i'(\beta(\sigma_i))} + 1\right) \cdot U_i^{h_0}(\sigma_i^p, I^{M-1}).\end{aligned}$$

That is, to prove the BNE when $n \to \infty$, we need

$$\frac{\delta_i(I)}{1 + \alpha \cdot k_i'(\alpha)} \geq \frac{\delta_i(\sigma_i)}{1 + \beta(\sigma_i) \cdot k_i'(\beta(\sigma_i))},\ \forall \sigma_i \in \Sigma_i. \tag{6}$$

For any σ_i with $f_i^{\sigma_i} = f_i$, $\beta(\sigma_i) = \alpha$, and by Theorem 5.2, $\delta_i(I) \geq \delta_i(\sigma_i)$, which holds regardless of the form of functions k_i. For any σ_i with $f_i^{\sigma_i} \neq f_i$, consider $k_i'(\alpha)$ to be in the form of $\alpha^c \cdot B$ with constants B and c, and substitute it into condition (6), we would equivalently require

$$\delta_i(\sigma_i) - \delta_i(I) \leq \left(\delta_i(I) \cdot \beta^{c+1}(\sigma_i) - \delta_i(\sigma_i) \cdot \alpha^{c+1}\right) \cdot B,\ \forall \sigma_i. \tag{7}$$

By feasibility of the problem case and the validity of the two-sample test, we have $\delta_i(I) > 0$ and $\beta(\sigma_i) > \alpha$. Therefore, for each σ_i with $f_i^{\sigma_i} \neq f_i$, we are able to find a large enough positive constant c_{σ_i}, such that $\delta_i(I) \cdot \beta^{c_{\sigma_i}+1}(\sigma_i) - \delta_i(\sigma_i) \cdot \alpha^{c_{\sigma_i}+1} > 0$. Taking c to be the supreme over those c_{σ_i}, we would have this term to be positive for all the σ_i with $f_i^{\sigma_i} \neq f_i$. Then taking B to be a large positive constant such that condition (6) is satisfied for all the σ_i finishes the proof. □