

---

# Federated Transformer: Multi-Party Vertical Federated Learning on Practical Fuzzily Linked Data

---

Zhaomin Wu, Junyi Hou, Yiqun Diao, Bingsheng He

National University of Singapore, Singapore

[zhaomin@nus.edu.sg](mailto:zhaomin@nus.edu.sg), [{juniyi.h,yiqun,hebs}@comp.nus.edu.sg](mailto:{juniyi.h,yiqun,hebs}@comp.nus.edu.sg)

## Abstract

Federated Learning (FL) is an evolving paradigm that enables multiple parties to collaboratively train models without sharing raw data. Among its variants, Vertical Federated Learning (VFL) is particularly relevant in real-world, cross-organizational collaborations, where distinct features of a shared instance group are contributed by different parties. In these scenarios, parties are often linked using fuzzy identifiers, leading to a common practice termed as *multi-party fuzzy VFL*. Existing models generally address either multi-party VFL or fuzzy VFL between two parties. Extending these models to practical multi-party fuzzy VFL typically results in significant performance degradation and increased costs for maintaining privacy. To overcome these limitations, we introduce the *Federated Transformer (FeT)*, a novel framework that supports multi-party VFL with fuzzy identifiers. FeT innovatively encodes these identifiers into data representations and employs a transformer architecture distributed across different parties, incorporating three new techniques to enhance performance. Furthermore, we have developed a multi-party privacy framework for VFL that integrates differential privacy with secure multi-party computation, effectively protecting local representations while minimizing associated utility costs. Our experiments demonstrate that the FeT surpasses the baseline models by up to 46% in terms of accuracy when scaled to 50 parties. Additionally, in two-party fuzzy VFL settings, FeT also shows improved performance and privacy over cutting-edge VFL modelstings.

## 1 Introduction

Federated Learning (FL) is a learning paradigm that enables multiple parties to collaboratively train a model while preserving the privacy of their local data [27]. Among its various forms, Vertical Federated Learning (VFL) [53] is particularly prevalent form in real-world applications as highlighted in a recent technical report [48]. In VFL, participants possess different features of the same set of instances, where common features, such as names or addresses, serve as *identifiers* (a.k.a. *keys*) to link datasets across these parties.

Real-world applications often necessitate *multi-party fuzzy VFL*, characterized by two key attributes. First, it supports collaboration among *multiple parties*, commonly observed in collaborations across hospitals [33], sensors [52], and financial institutions [38]. Second, it accommodates scenarios where these parties are linked using fuzzy identifiers, such as addresses. Such scenarios are prevalent in applications, as illustrated in an analysis [50] of the German Record Linkage Center [11]. For instance, multiple vehicle rental companies that are fuzzily linked by source and destination addresses in the same city can collaborate to predict travel times.

To illustrate the significance of multi-party fuzzy VFL, consider the application of travel cost prediction in a city through collaboration among taxi, car, bike, and bus companies, as shown in Figure 1. Since personal travel information is private and cannot be shared, VFL is essential.

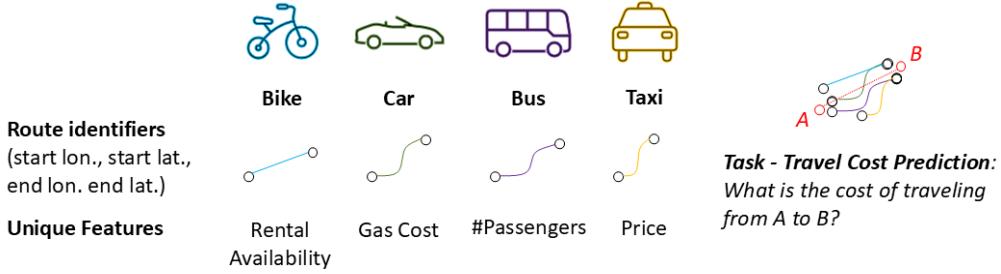


Figure 1: Real application of multi-party fuzzy VFL: travel cost prediction in a city

Additionally, route identifiers—starting and ending GPS locations—can only be linked using fuzzy methods. However, linking closely related source and destination points with multi-party fuzzy VFL can significantly enhance prediction accuracy.

Existing VFL approaches generally address either the multi-party aspect or the fuzzy identifier issue. Several methods [33, 52, 38, 21, 26] facilitate multi-party VFL using Private Set Intersection (PSI) [10] to link datasets. These methods often presume the existence of precise, universal keys, which are not feasible in common VFL scenarios involving fuzzy identifiers. Conversely, other studies [17, 34, 50] propose two-party fuzzy VFL models that utilize cross-party key similarities for training. However, when extended to multi-party fuzzy VFL, these similarity-based approaches encounter significant challenges in performance and privacy. While some methods achieve reasonable performance, they often compromise privacy or incur prohibitive costs.

Despite the potential of multi-party fuzzy VFL, several significant challenges must be addressed for effective implementation. First, as the number of parties with *fuzzy identifiers* increases, maintaining performance becomes increasingly challenging. The addition of parties leads to a quadratic growth in the number of key pairs, an increase in incorrect linkages between fuzzy identifiers, and larger model sizes. These factors collectively heighten the risk of overfitting and adversely impact model performance. Second, the rising costs of preserving privacy intensify as more parties with *correlated data* participate, leading to either significant computational costs [33, 52, 38] or accuracy loss [47]. Third, in a collaboration of multiple parties, a communication bottleneck arises for the party with labels, termed the *primary party*. This party must communicate with all other parties without labels, termed *secondary parties*, in each training round, placing substantial demands on the primary party’s bandwidth. These challenges significantly hinder the practical deployment of VFL.

To address these issues, we introduce the *Federated Transformer (FeT)* to enhance performance and reduce privacy costs in multi-party fuzzy VFL. First, to tackle performance issues, we encode key similarities into data representations aligned by *positional encoding averaging*, which eliminates the need for quadratic calculations of key pairs. Additionally, we have designed a trainable *dynamic masking* module that automatically filters out incorrectly linked pairs, enhancing model accuracy by up to 13% in 50-party fuzzy VFL on the MNIST dataset. Second, to mitigate the escalating costs of privacy protection, we introduce SplitAvg, a hybrid approach that merges encryption-based and noise-based methods, maintaining a consistent noise level regardless of the number of participating parties. Third, to alleviate communication overhead on the primary party, we implement a *party dropout* strategy, which randomly excludes certain secondary parties during each training round. This effectively reduces communication costs by approximately 80% and improves model generalization. Our codes are available on GitHub<sup>1</sup>. In summary, our contributions are as follows:

- We design *Federated Transformer (FeT)*, a novel model achieving promising performance under multi-party fuzzy VFL.
- We introduce *SplitAvg* to enhance the privacy of FeT by protecting local representations in multi-party fuzzy VFL, with a theoretical proof of its differential privacy.
- Experimental results demonstrate that FeT significantly outperforms baseline models by up to 46% in terms of accuracy in 50-parties VFL. Moreover, while providing enhanced privacy, FeT consistently surpasses state-of-the-art models even in traditional two-party fuzzy VFL scenarios.

<sup>1</sup><https://github.com/Xtra-Computing/FeT>

## 2 Preliminaries

In this section, we provide the foundational concepts necessary for understanding our approach to differential privacy. Differential Privacy (DP) offers a rigorous mathematical framework for preserving individual privacy. It quantifies privacy in terms of the probability of producing the same output from two similar datasets that differ by exactly one record.

**Definition 1.** Consider a randomized function  $\mathcal{M} : \mathbb{R}^d \rightarrow \mathcal{O}$  and two neighboring databases  $D_0, D_1 \sim \mathbb{R}^d$  that differ by exactly one record. For every possible output set  $O \subseteq \mathcal{O}$ ,  $\mathcal{M}$  satisfies  $(\varepsilon, \delta)$ -differential privacy if

$$\Pr[\mathcal{M}(D_0) \in O] \leq e^\varepsilon \Pr[\mathcal{M}(D_1) \in O] + \delta,$$

where  $\varepsilon \geq 0$  and  $\delta \geq 0$ .

A single query that adheres to differential privacy is termed a *mechanism*. For example, the Gaussian mechanism [4] is commonly used to achieve DP by adding Gaussian noise to the output of the function.

**Theorem 1** (Gaussian Mechanism [4]). *For a function  $f : \mathbb{X} \rightarrow \mathbb{R}^d$  characterized by a global  $L_2$  sensitivity  $\Delta_2$ , which signifies that the maximum difference in the  $L_2$ -norm of the outputs of  $f$  on any two neighboring databases is  $\Delta_2$ , and for any  $\varepsilon \geq 0$  and  $\delta \in [0, 1]$ , the Analytic Gaussian Mechanism is defined as  $\mathcal{M}(x) = f(x) + Z$ , where  $Z \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$ . This mechanism satisfies  $(\varepsilon, \delta)$ -differential privacy if  $\Phi\left(\frac{\Delta_2}{2\sigma} - \frac{\varepsilon\sigma}{\Delta_2}\right) - e^\varepsilon \Phi\left(-\frac{\Delta_2}{2\sigma} - \frac{\varepsilon\sigma}{\Delta_2}\right) \leq \delta$ , where  $\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^t e^{-x^2/2} dx$  is the cumulative distribution function (CDF) of the standard univariate Gaussian distribution.*

When multiple queries are made on the same database, independent Gaussian noise is added to each query to maintain differential privacy. The privacy loss of the composition of Gaussian mechanisms is formulated in Theorem 2.

**Theorem 2** (Moments Accountant [1]). *There exist constants  $c_1$  and  $c_2$  so that given the sampling probability  $q = \frac{L}{N}$  and the number of steps  $T$ , for any  $\varepsilon < c_1 q^2 T$ , DPSGD [1] is  $(\varepsilon, \delta)$ -differentially private for any  $\delta > 0$  if we choose  $\sigma > c_2 \frac{q\sqrt{T \log(1/\delta)}}{\varepsilon}$ .*

## 3 Related Work

**Performance.** Traditional VFL approaches [29, 7] are typically limited to two-party scenarios. In contrast, existing multi-party VFL methods [12, 51, 33, 52, 38, 21, 26] often rely on the assumption of precise identifiers that ensure perfect alignment across all parties. These methods generally employ the SplitNN framework [45], where each party maintains a portion of the model, and the models are collaboratively trained on well-aligned data samples through the transfer of representations and gradients, commonly known as split learning. However, the requirement for perfect data alignment is impractical in many real-world scenarios [50, 3], where identifiers are often imprecise.

To address this limitation, semi-supervised VFL [22, 30, 19, 20] has emerged, attempting to improve model performance by leveraging unlinked records through semi-supervised or self-supervised learning. However, these methods still assume that datasets from each party can be precisely linked using exact identifiers, a premise that is often untenable in real-world settings [50, 3]. Given that the quality of linkage significantly impacts VFL accuracy [34], exploring effective linkage methods remains a pivotal issue in VFL.

On the other hand, FedSim [50], based on real linkage projects at the German Record Linkage Center (GRLC) [3], acknowledges that the keys of each party are usually not precisely linkable and that records may have one-to-many relationships, leading to fuzzy linkage scenarios, as seen with keys like GPS addresses. FedSim enhances training performance by performing soft linkage and conducting training based on transmitted key similarities. Nonetheless, it faces limitations in scalability beyond two parties and introduces new privacy concerns by directly transferring similarities.

In summary, while existing studies face significant performance challenges when handling fuzzy keys in multi-party settings, our proposed FeT demonstrates a scalable design that addresses these challenges and shows promising performance improvements in both multi-party fuzzy VFL and two-party settings compared to FedSim.

**Privacy.** The privacy concerns associated with VFL are multifaceted. First, the primary party may infer data representations from secondary parties [31]. Second, the secondary party may derive gradients from the primary party [41, 54]. Third, external attackers could conduct membership inference attacks [39] on the deployed model [51]. This paper primarily addresses the second concern: safeguarding representations, while acknowledging other concerns as open challenges.

To address the privacy of representations in VFL, various methods have been proposed, falling into two primary categories: encryption-based methods and noise-based methods. Encryption-based methods [26, 14, 33, 52, 38, 21, 36] utilize computationally intensive cryptographic techniques to encrypt intermediate results. However, these methods often incur significant communication overhead when scaled to multiple parties. Conversely, noise-based methods [47, 46] protect data by perturbing [47] or manipulating [46] local representations. These methods typically do not provide theoretical privacy guarantees or require substantial amounts of noise when scaling to multiple parties in VFL, which can degrade performance. Unlike existing studies that rely solely on either approach, this paper explores a combined strategy incorporating both encryption-based and noise-based methods, ensuring the model scales effectively to multiple parties without the need for excessive noise.

## 4 Problem Statement

In this section, we formalize the concept of multi-party fuzzy VFL. We consider a supervised learning task where one party holding labels, termed the *primary party*  $P$ , collaborates with  $k$  parties that do not hold labels, referred to as the *secondary parties*. The primary party  $P$  possesses  $n$  data records denoted as  $\mathbf{x}^P := \{x_i\}_{i=1}^n$  along with corresponding labels  $\mathbf{y} := \{y_i\}_{i=1}^n$ . Each secondary party  $S_k$  maintains its own dataset  $\mathbf{x}^{S_k}$ . All parties share common features, referred to as identifiers, expressed as  $\mathbf{x}^i = [\mathbf{k}^i, \mathbf{d}^i]$ , where  $[.]$  signifies concatenation. These identifiers  $\mathbf{k}^i$  may exhibit inaccuracies and fuzziness, despite residing within the same range.

$$\min_{\theta} \frac{1}{n} \sum_{i=1}^n \mathcal{L}(f(\theta; x_i^P, \mathbf{x}^{S_1}, \dots, \mathbf{x}^{S_k}); y_i) + \Omega(\theta)$$

In this formulation,  $\mathcal{L}(\cdot)$  denotes the loss function,  $\theta$  represents the model parameters, and  $\Omega(\theta)$  refers to the regularization term. The symbol  $n$  indicates the number of samples in the primary party  $P$ .

**Threat Model.** This study focuses on defending feature reconstruction attacks [25, 31], which target local representations shared with the primary party. FeT operates under the assumption that all participating parties are *honest-but-curious*, meaning they adhere to the protocol but may attempt to infer additional information about other parties. Furthermore, we assume that the parties do not collude with one another. While other forms of attacks, such as label inference attacks [13] and backdoor attacks aimed at compromising labels and gradients, exist, they are considered orthogonal to the objectives of this study. These additional threats will be explored in our future research.

## 5 Approach

In this section, we address the performance and communication challenges inherent in multi-party fuzzy VFL. To tackle these issues, we introduce a transformer-based architecture named the Federated Transformer (FeT). This model encodes keys into data representations, thereby reducing reliance on key similarities. To accurately exclude incorrectly linked data records, we propose a trainable dynamic masking module that generates masks based on keys. Furthermore, to combat overfitting caused by the large model and to alleviate communication bottlenecks faced by the primary party, we introduce a party dropout strategy that randomly invalidates some parties during training. Additionally, we identify a positional encoding misalignment issue across parties in the FeT and propose positional encoding averaging to ensure consistent alignment, thereby enhancing model performance.

### 5.1 Model Structure

The architecture of the FeT is illustrated in Figure 2. In FeT, each secondary party has an encoder, while the primary party has both an encoder and a subsequent decoder. The internal structure of both the encoder and decoder closely adheres to the conventional transformer model [43]. We utilize

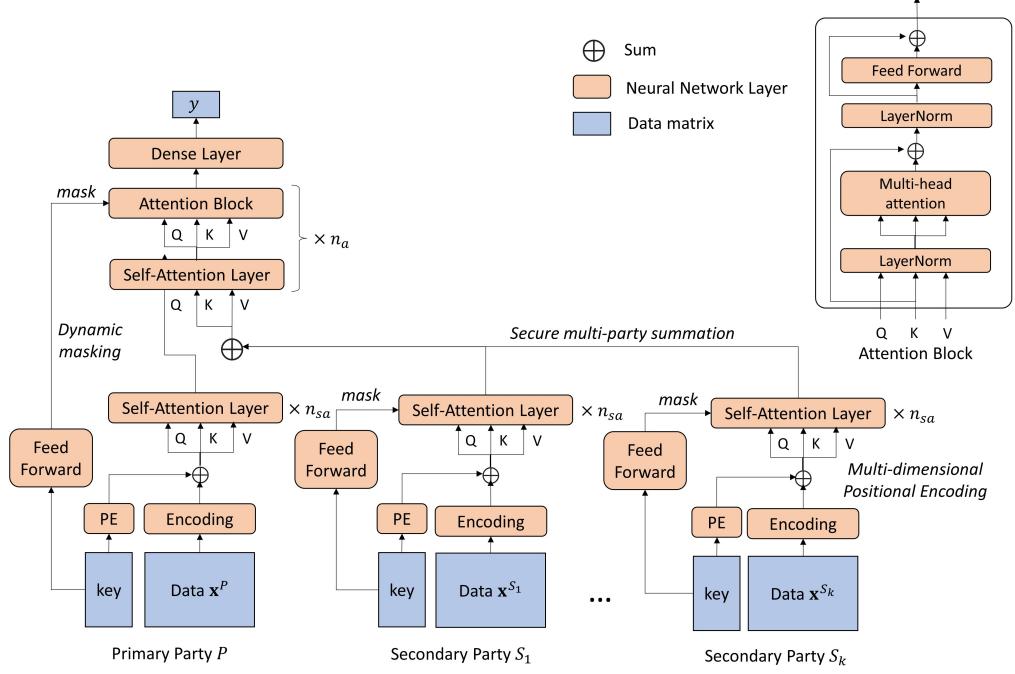


Figure 2: Structure of federated transformer (PE: multi-dimensional positional encoding)

multi-dimensional positional encoding [28] to integrate key information into feature vectors. Outputs from the encoders at the secondary parties are aggregated and then fed into the decoder. Details regarding the privacy mechanisms employed during this aggregation phase are discussed in Section 6, while the details of the training process are elaborated in Section 5.2. We then elaborate on three techniques designed to improve performance and reduce communication costs.

**Dynamic Masking.** The size of the neighborhood varies significantly depending on the party and key values. Consequently, including a large number of neighbors  $K$  for all parties can hinder the model’s ability to extract meaningful information and result in overfitting. To address this, we introduce a dynamic “key padding mask” in the transformer, generated from the identifier values using a trainable MLP. This approach allows the model to effectively disregard distant data records, thereby eliminating the influence of irrelevant data when  $K$  is large. This concept resembles the weight gate in FedSim, but it diverges by using identifiers as inputs instead of similarities, enhancing privacy by preventing the transmission of similarity data across parties.

The learned dynamic masking is visualized in Figure 3. We derive two key insights from the visualization: (1) Dynamic masking effectively focuses on a localized area around the primary party’s identifiers. Data records with distant identifiers on secondary parties (in cooler colors) receive small negative mask values, reducing their significance in the attention layers - without accessing the primary party’s original identifiers. (2) The focus area varies in scale and direction across samples: for example, the left figure concentrates on a small bottom area, the middle figure on a small top area, and the right figure on a broad area in all directions.

**Party Dropout.** Extending the Federated Transformer (FeT) to support multiple parties can be challenging for several reasons. First, the communication bandwidth required by the primary party becomes a significant bottleneck within the SplitAvg framework, increasing linearly with the number of parties. Second, the inclusion of many parties can result in an excessive number of parameters, which may lead to overfitting. To mitigate these issues, we introduce the concept of *party dropout*. Inspired by traditional dropout [40], we randomly set a portion  $r_d$  of the parties’ representations to zero during training. This method serves as a form of regularization, thus helping to reduce overfitting, while also significantly cutting down on communication overhead. In our experiments, we demonstrate that increasing the party dropout rate to 0.8 leads to minimal accuracy loss and can

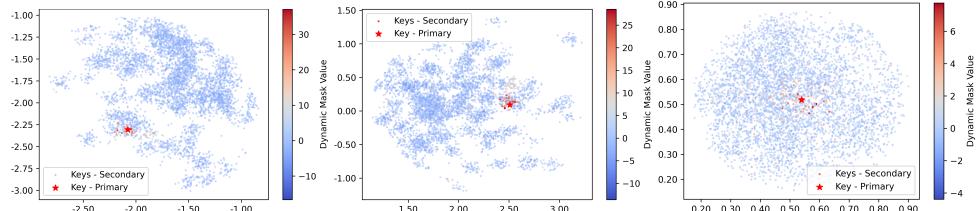
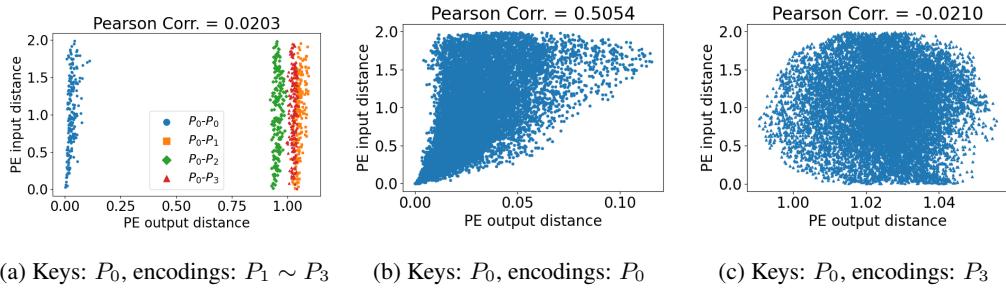


Figure 3: Learned dynamic masks of different samples: Each figure displays one sample (red star) from the primary party fuzzily linked with 4900 samples (circles) from 49 secondary parties. The position indicates the sample’s identifier, and colors reflect learned dynamic mask values. Larger mask values signify higher importance in attention layers.



(a) Keys:  $P_0$ , encodings:  $P_1 \sim P_3$       (b) Keys:  $P_0$ , encodings:  $P_0$       (c) Keys:  $P_0$ , encodings:  $P_3$

Figure 4: Misalignment of positional encoding ( $P_0$ : primary party;  $P_1 \sim P_3$ : secondary parties)

even improve accuracy. Consequently, the communication overhead on the primary party can be reduced by up to 80%, enhancing scalability when dealing with large numbers of parties.

Like traditional dropout, it is crucial to maintain consistent scaling of the representations during both training and testing phases. This consistency is naturally achieved within the SplitAvg framework. During the averaging process, if a ratio  $r_d k$  of parties is set to zero, we adjust by dividing only by the number of non-zero parties,  $(1 - r_d)k$ . This method ensures that the scale of the averaged representations remains consistent across training and testing, regardless of the value of  $r_d$ .

**Positional Encoding Averaging.** In positional encoding (PE), it is generally expected that the distances between encoded representations are positively correlated with the distances between identifiers. In FeT, each party employs its own encoder and PE layer, each tasked with encoding its local identifiers into representations. This configuration leads to significant PE misalignment issues, as illustrated in Figure 4. Although the identifiers and their corresponding encoded representations maintain a positive correlation within the PE layer of each party, there is almost no correlation between the identifiers and encoded representations across different parties. This lack of correlation causes integration issues and affects accuracy. However, directly sharing a PE layer among all parties is not viable due to privacy concerns. To address this, we propose positional encoding averaging.

Every  $T_{pe}$  epoch, the positional encoding layers are averaged and broadcast to all parties under a secure multi-party computation (MPC) scheme, akin to FedAvg [23] in horizontal federated learning [27]. While the privacy of the transmitted model can be a concern, this issue is an orthogonal open problem in horizontal federated learning.

## 5.2 Training

The FeT training process begins with employing Privacy-Preserving Record Linkage (PPRL) [44] to evaluate identifier similarities between the primary party  $P$  and each secondary party. Secondary parties each contribute a random subset for linkage (line 5). For each  $P$ ’s record,  $K$  nearest neighbors within these subsets from secondary parties are determined (line 6). The training leverages data embeddings of dimensions  $B \times L \times H$ , where  $B$  is batch size,  $L$  is the sequence length, and  $H$  is the hidden layer size, following the transformer architecture. In FeT’s context,  $L = 1$  for the primary and  $L = K$  for secondary parties, linking each primary record with  $K$  neighboring records from the secondaries. Identifiers are transformed into vectors using multi-dimensional positional

encoding [28] and combined with data representations for processing via self-attention blocks (lines 7, 10). Secondary parties' representations are averaged under the MPC protocol. The primary party then employs attention blocks for forward propagation to compute the final prediction (line 13). Backpropagation sends gradient updates from the primary to secondary parties to refine their local models (lines 14-16). The privacy mechanism including norm clipping (lines 8, 11) and distributed Gaussian noise (line 12) are further discussed in Section 6.

---

**Algorithm 1:** Training Process of Federated Transformer

---

**Input :** Primary party  $\mathbf{x}^P$ , secondary parties  $\mathbf{x}^{S_1}, \dots, \mathbf{x}^{S_k}$ , label  $\mathbf{y}$ , noise scale  $\sigma$ , sampling ratio  $q$ , clipping threshold  $C$   
**Output :** Local models  $\theta_l^P, \theta_l^{S_1}, \dots, \theta_l^{S_k}$  and aggregation model  $\theta_a^P$

```

1 Initialize model parameters  $\theta_a^P, \theta_l^P, \theta_l^{S_1}, \dots, \theta_l^{S_k}$ 
2 for epoch  $t \in [T]$  do
3   for instance  $x_i^P \in \mathbf{x}^P$  on primary party do
4     for party  $h \in \{S_1, \dots, S_k\}$  do
5        $\tilde{\mathbf{x}}^h \leftarrow$  randomly choose  $q$  ratio from  $\mathbf{x}^h$ 
6        $\tilde{\mathbf{x}}_i^h \leftarrow$  link  $K$  records with  $x_i^P$  from  $\tilde{\mathbf{x}}^h$ 
7        $\tilde{\mathbf{R}}_i^h \leftarrow f(\theta_l^h; \tilde{\mathbf{x}}_i^h)$ 
8        $\hat{\mathbf{R}}_i^h \leftarrow \tilde{\mathbf{R}}_i^h / \max\left(1, \frac{\|\tilde{\mathbf{R}}_i^h\|_2}{C}\right)$  // Norm clipping
9     end
10     $\mathbf{R}_i^P \leftarrow f(\theta_l^P; x_i^P)$ 
11     $\hat{\mathbf{R}}_i^P \leftarrow \mathbf{R}_i^P / \max\left(1, \frac{\|\mathbf{R}_i^P\|_2}{C}\right)$  // Norm clipping
12     $\mathbf{H}_i \leftarrow \text{MPCAvg}\left(\mathbf{R}_i^{S_1}, \dots, \mathbf{R}_i^{S_k}, \mathcal{N}(0, C^2\sigma^2)\right)$ 
13     $\hat{y}_i \leftarrow f(\theta_a^P; \mathbf{H}_i)$ 
14     $\nabla_{\theta_a^P} \leftarrow \frac{\partial \ell(y_i, \hat{y}_i)}{\partial \theta_a^P}$ ,  $\theta_a^P \leftarrow \theta_a^P - \eta_t \nabla_{\theta_a^P}$ 
15    for party  $h \in \{P, S_1, \dots, S_k\}$  do
16       $\nabla_{\theta_l^h} \leftarrow \frac{\partial \ell(y_i, \hat{y}_i)}{\partial \theta_l^h}$ ,  $\theta_l^h \leftarrow \eta_t \nabla_{\theta_l^h}$ 
17    end
18  end
19 end

```

---

## 6 Privacy

In this section, we address the challenges of privacy in multi-party fuzzy VFL. First, the risk of transferring raw similarities has been mitigated by the design of the FeT itself. Second, to address the increasing costs when more parties join, we introduce a multi-party privacy-preserving VFL framework, *SplitAvg*, which is compatible with FeT. The architecture of SplitAvg is illustrated in Figure 5. SplitAvg integrates differential privacy (DP), secure multi-party computation (MPC) [6], and norm clipping to enhance the privacy of representations. Additionally, to further improve the utility of FeT under DP, we employ privacy amplification techniques that reduce the noise scales by incorporating random sampling.

### 6.1 Differentially Private Split Neural Network - SplitAvg

This section outlines three techniques applied to the SplitAvg to improve privacy: representation norm clipping, privacy amplification, and MPC with distributed Gaussian noise. These strategies collectively protect the privacy of each secondary party's representations through differential privacy and ensure that privacy risks do not escalate with an increase in the number of parties due to MPC.

**Representation Norm Clipping.** The magnitude of the  $\ell_2$ -norm is pivotal in determining the sensitivity of differential privacy. To limit the maximum change of the  $\ell_2$ -norm, norm clipping is essential. Specifically, for a representation  $\mathbf{R}$ , we ensure that  $\|\mathbf{R}\|_2 \leq C$ , where  $C$  is a predefined positive

real number. This is achieved by scaling  $\mathbf{R}$  by a factor of  $C$ , formally,  $\hat{\mathbf{R}} = \mathbf{R} / \max(1, \|\mathbf{R}\|_2/C)$ . Through this process, any representation  $\mathbf{R}$  with  $\|\mathbf{R}\|_2$  exceeding  $C$  is scaled to  $C$ , while values of  $\|\mathbf{R}\|_2$  below  $C$  remain unaffected.

**Secure Multi-Party Computation with Distributed Gaussian Noise.** To address the challenges of applying differential privacy in multi-party VFL, we propose a method that integrates noise addition into the process of aggregating representations, facilitated through MPC. In this setup, each secondary party first independently conducts representation norm clipping to limit the scale of their data. Subsequently, these clipped representations, along with Gaussian noise  $\mathcal{N}(0, \sigma^2/k^2)$  independently generated by each of the  $k$  secondary parties, are aggregated through averaging under MPC. For the primary party, this aggregation is equivalent to adding independent Gaussian noise  $\mathcal{N}(0, \sigma^2)$  to the averaged result. The adoption of MPC in our framework ensures that the secondary parties do not need to individually add  $\mathcal{N}(0, \sigma^2)$  noise to their representations. Instead, as the primary party only has access to the averaged result under MPC, each secondary party can add a smaller amount of noise. This method effectively improves utility with a small efficiency cost due to MPC.

**Privacy Amplification by Secondary Subsampling.** This technique is specifically designed for FeT configurations. According to the principle of privacy amplification [5], applying a function to a randomly sampled subset of data enhances privacy compared to applying the same function to the entire dataset. By initiating linkage from a randomly sampled subset rather than the full dataset, the privacy parameters effectively shift from  $(\varepsilon, \delta)$  to  $(q\varepsilon, q\delta)$ , where  $q < 1$  represents the sampling rate. In FeT, the primary party typically selects subsets of candidate data records for training from each secondary party, targeting those with neighboring identifiers. By pre-sampling these subsets at a rate of  $q$  before conducting a k-nearest neighbors (kNN) search, we avoid processing the entire dataset, which in turn reduces the noise required to maintain the same privacy level.

## 6.2 Privacy Guarantee

Our analysis of differential privacy focuses on a hypothetical global dataset linked using all secondary parties, denoted as  $\mathbf{x}^{S_1}, \dots, \mathbf{x}^{S_k}$ . Since these datasets are correlated, removing one data record from this global dataset will result in changes to the representations in all secondary parties. Consequently, privacy loss accumulates across secondary parties without the use of MPC. However, with MPC, a single aggregated noise, formed by distributing smaller noise contributions among parties, can be added, effectively reducing the overall required noise. The privacy guarantee for these representations is formally articulated in Theorem 3, with the proof provided in Appendix A.

**Theorem 3.** *For certain constants  $c_1$  and  $c_2$ , given a sampling rate  $q$ , the total number of epochs  $T$ , and the number of batches  $B$  in each epoch, each representation  $\mathbf{R}^{S_k}$  achieves  $(\varepsilon, \delta)$ -differential privacy for all  $\varepsilon < c_1 q^2 T$ , with any  $\delta > 0$ , by selecting the standard deviation  $\sigma$  of the Gaussian noise mechanism as follows:*

$$\sigma > c_2 \frac{q\sqrt{BT \log(1/\delta)}}{\varepsilon}. \quad (1)$$

## 7 Experiments

This section presents the experimental setup and results. We begin by outlining the experimental settings in Section 7.1, followed by an assessment of performance across varying numbers of parties and neighbors in Section 7.2. We then analyze the privacy of FeT in Section 7.3. Additionally, an ablation study is conducted in Appendix C to evaluate the contribution of each component to performance, including dynamic masking, party dropout, positional encoding, key fuzziness, and

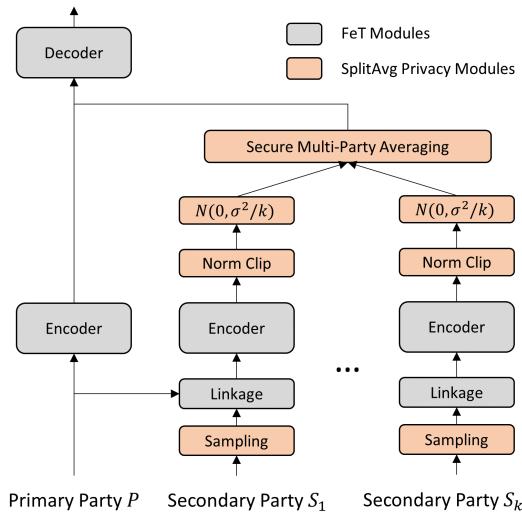


Figure 5: Differentially private split-sum neural network

SplitAvg. The performance of FeT with exact key matching is assessed in Appendix D, while the computational and memory efficiency of MPC and training is evaluated in Appendix E. Privacy evaluation on two-party real datasets is included in Appendix F. Furthermore, FeT’s performance under imbalanced feature splits across parties, based on VertiBench [49], is presented in Appendix G.

## 7.1 Experimental Settings

**Datasets.** Our experiments utilize five datasets, including three real-world datasets: `house` [35, 2], `bike` [8, 42], and `hdb` [18, 37], along with two high-dimensional datasets: `gisette` [16] and `MNIST` [24]. Detailed descriptions of these datasets can be found in Appendix 7.1. To simulate multi-party fuzzy VFL, we partition the features equally and randomly among the parties. The primary party’s feature dimensions are reduced to 4 using principal component analysis (PCA) to serve as universal keys. To create fuzzy linked scenarios, we add independent Gaussian noise with a scale of 0.05 to the keys of each party.

**Baselines.** We include three baselines in our experiments: (1) Solo: training only on the primary party; (2) Top1Sim: linking each data record in the primary party only with its most similar neighbor in the secondary parties; (3) FedSim [50]: training on the top  $K$  neighboring data records.

## 7.2 Performance

**Two-party fuzzy VFL.** In this experiment, we evaluate the performance of FeT in two-party settings without privacy mechanisms. The detailed results are presented in Table 1. Our experiments demonstrate that FeT consistently outperforms the leading two-party fuzzy VFL methods. Notably, this performance improvement is achieved while enhancing privacy protections, as FeT does not involve transferring similarity data.

Table 1: Root Mean Squared Error (RMSE) on real-world two-party fuzzy VFL datasets

Algorithm	house	bike	hdb
Solo	$73.27 \pm 0.16$	$244.33 \pm 0.75$	$33.97 \pm 0.61$
Top1Sim	$58.54 \pm 0.35$	$256.19 \pm 1.39$	$31.56 \pm 0.21$
FedSim	$42.12 \pm 0.23$	$235.67 \pm 0.27$	$27.13 \pm 0.06$
FeT	<b><math>39.75 \pm 0.29</math></b>	<b><math>232.98 \pm 0.62</math></b>	<b><math>26.94 \pm 0.15</math></b>

**Effect of Number of Neighbors  $K$ .** In this experiment, we assess the impact of the number of neighbors,  $K$ , on FeT’s performance by varying  $K$  from 1 to 100. The results are displayed in Figure 6. The FedSim baseline is trained using the optimal  $K$  value (i.e., 50 for `hdb` and 100 for `house` and `bike`). The figure reveals two key insights: First, FeT’s performance improves as  $K$  increases, demonstrating its ability to filter useful information even as the number of unrelated data records grows. Second, FeT consistently outperforms all baselines at larger values of  $K$ , highlighting its superiority in fuzzy VFL scenarios.

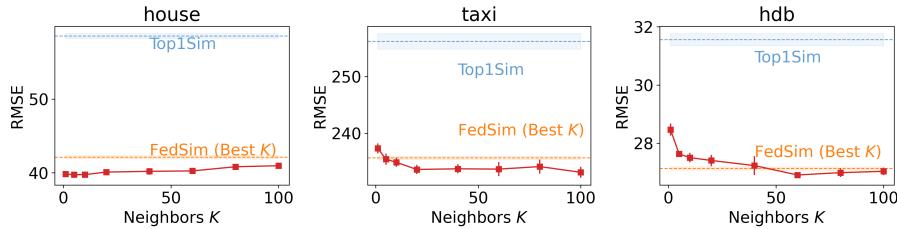


Figure 6: Effect of Different Number of Neighbors  $K$  on FeT Performance

**Effect of Number of Parties** In this experiment, we assess FeT’s performance in fuzzy VFL with various numbers of parties. Due to the absence of real multi-party VFL data, we employ synthetic data for our evaluations. We partition the features equally and randomly among the parties, reducing

the primary party’s feature dimensions to 4 using PCA as the universal keys. To simulate fuzzy linked scenarios, we add independent Gaussian noise with a scale of 0.05 to the keys of each party.

Figure 7 shows that FeT generally outperforms the baselines, particularly with a larger number of parties. This advantage is attributed to Solo’s lack of informative features and Top1Sim’s noise-affected linkage. FedSim performs poorly as the top-1 linked secondary parties are unaware of the primary parties’ keys, leading to misalignment in subsequent soft linkage and training steps. On the *gisette* dataset with  $k = 10$ , FeT and other models slightly underperform compared to Solo, likely due to overfitting given *gisette*’s small size.

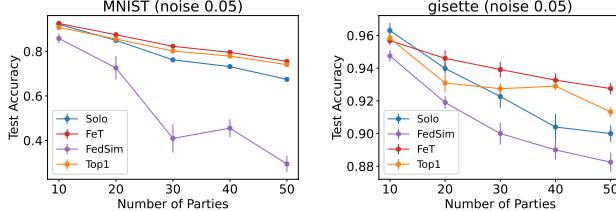


Figure 7: Impact of number of parties on FeT performance

### 7.3 Privacy

In this subsection, we analyze how the performance of FeT varies with different noise scales ( $\sigma$ ) and sampling rates on secondary parties, demonstrating the impact of privacy constraints on its accuracy. The results are depicted in Figure 8. We observe three key points: First, a moderate sampling rate has a negligible effect on model performance and may even slightly improve performance (e.g., on the MNIST dataset) by reducing overfitting. Second, despite increasing noise levels and enhanced privacy guarantees, FeT consistently outperforms baseline models. Third, the  $\varepsilon - \sigma$  privacy-noise curves illustrate that solely adding Gaussian noise without MPC, even with advanced analysis theorems such as Rényi Differential Privacy (RDP) [32], would require much larger noise scales compared to our approach that integrates MPC.

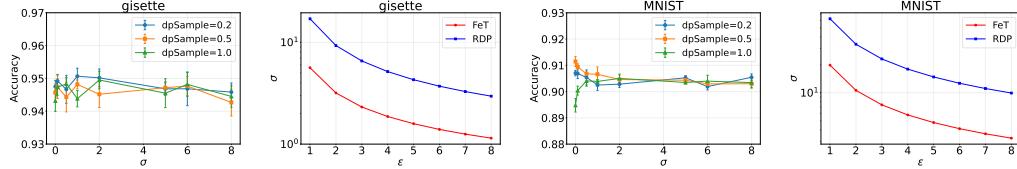


Figure 8: Impact of noise scale  $\sigma$  on FeT accuracy and relationship between  $\sigma$  and  $\varepsilon$  under 10-party fuzzy VFL (RDP: without MPC, privacy loss calculated by Rényi differential privacy)

## 8 Conclusion

In this study, we introduce the Federated Transformer (FeT), specifically designed to support multi-party VFL while effectively addressing critical challenges related to performance, privacy, and communication. Furthermore, we provide theoretical proof of FeT’s differential privacy, ensuring that data representations remain protected from secondary parties. Notably, our experiments demonstrate that FeT surpasses baseline models, even under stringent privacy guarantees and within the traditional two-party setting, establishing its efficacy and robustness in complex federated environments.

**Broader Impact.** The architecture of FeT, even without privacy mechanisms, has potential applications in multimodal learning. Multimodal tasks often require the alignment of data records across different modalities, which can be quite challenging. For instance, aligning 24Hz video streams with 48kHz audio tracks is complex, as each video frame may correspond to a range of audio samples. FeT has shown its capability to effectively learn from such fuzzily aligned data. Furthermore, the transformer model has proven effective across various data types, including images, text, and tabular data, highlighting FeT’s suitability for multimodal learning applications.

## Acknowledgments

This research/project is supported by the National Research Foundation Singapore and DSO National Laboratories under the AI Singapore Programme (AISG Award No: AISG2-RP-2020-018), the National Research Foundation Singapore and Infocomm Media Development Authority under its Trust Tech Funding Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation Singapore, DSO National Laboratories, and Infocomm Media Development Authority. This research is also sponsored by Webank Scholars Program. We thank the AMD Heterogeneous Accelerated Compute Clusters (HACC) program (formerly known as the XACC program - Xilinx Adaptive Compute Cluster program) for the generous hardware donation.

## References

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*. 308–318.
- [2] Airbnb. 2019. Airbnb prices dataset. <http://insideairbnb.com/get-the-data.html>.
- [3] Manfred Antoni and Rainer Schnell. 2019. The past, present and future of the German Record Linkage Center (GRLC). *Jahrbücher für Nationalökonomie und Statistik* 239, 2 (2019), 319–331.
- [4] Borja Balle and Yu-Xiang Wang. 2018. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In *International Conference on Machine Learning*. PMLR, 394–403.
- [5] Amos Beimel, Hai Brenner, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. 2014. Bounds on the sample complexity for private learning and private data release. *Machine learning* 94 (2014), 401–437.
- [6] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 1175–1191.
- [7] Timothy J Castiglia, Anirban Das, Shiqiang Wang, and Stacy Patterson. 2022. Compressed-VFL: Communication-Efficient Learning with Vertically Partitioned Data. In *Proceedings of the 39th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 162)*, Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (Eds.). PMLR, 2738–2766. <https://proceedings.mlr.press/v162/castiglia22a.html>
- [8] CitiBike. 2016. Citi bike system data. <https://www.citibikenyc.com/system-data>.
- [9] Yuren Cong, Wentong Liao, Hanno Ackermann, Bodo Rosenhahn, and Michael Ying Yang. 2021. Spatial-temporal transformer for dynamic scene graph generation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 16372–16382.
- [10] Changyu Dong, Liqun Chen, and Zikai Wen. 2013. When private set intersection meets big data: an efficient and scalable protocol. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. 789–800.
- [11] Johanna Eberle and Michael Weinhardt. 2016. Record linkage of the linked employer-employee survey of the socio-economic panel study (SOEP-LEE) and the establishment history panel (BHP). *German Record Linkage Center Working Paper Series, No. WP-GRLC-2016-01* (2016).
- [12] Siwei Feng, Han Yu, and Yuebing Zhu. 2024. MMVFL: A Simple Vertical Federated Learning Framework for Multi-Class Multi-Participant Scenarios. *Sensors* 24, 2 (2024). <https://doi.org/10.3390/s24020619>

- [13] 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, Boston, MA, 1397–1414. <https://www.usenix.org/conference/usenixsecurity22/presentation/fu-chong>
- [14] Fangcheng Fu, Huanran Xue, Yong Cheng, Yangyu Tao, and Bin Cui. 2022. BlindFL: Vertical Federated Machine Learning without Peeking into Your Data. In *Proceedings of the 2022 International Conference on Management of Data* (Philadelphia, PA, USA) (*SIGMOD ’22*). Association for Computing Machinery, New York, NY, USA, 1316–1330. <https://doi.org/10.1145/3514221.3526127>
- [15] O. Goldreich, S. Micali, and A. Wigderson. 1987. How to play ANY mental game. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing* (New York, New York, USA) (*STOC ’87*). Association for Computing Machinery, New York, NY, USA, 218–229. <https://doi.org/10.1145/28395.28420>
- [16] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror. 2004. Gisette Dataset. [https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/gisette\\_scale.bz2](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary/gisette_scale.bz2)
- [17] Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Richard Nock, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2017. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv preprint arXiv:1711.10677* (2017).
- [18] Singapore HDB. 2018. Resale flat prices in Singapore. <https://data.gov.sg/dataset/resale-flat-prices>.
- [19] Yuanqin He, Yan Kang, Xinyuan Zhao, Jiahuan Luo, Lixin Fan, Yuxing Han, and Qiang Yang. 2022. A hybrid self-supervised learning framework for vertical federated learning. *arXiv preprint arXiv:2208.08934* (2022).
- [20] Chung-ju Huang, Leye Wang, and Xiao Han. 2023. Vertical Federated Knowledge Transfer via Representation Distillation for Healthcare Collaboration Networks. In *Proceedings of the ACM Web Conference 2023*. ACM, 4188–4199.
- [21] Yimin Huang, Wanwan Wang, Xingying Zhao, Yukun Wang, Xinyu Feng, Hao He, and Ming Yao. 2023. EFMVFL: an efficient and flexible multi-party vertical federated learning without a third party. *ACM Transactions on Knowledge Discovery from Data* 18, 3 (2023), 1–20.
- [22] Yan Kang, Yang Liu, and Xinle Liang. 2022. FedCVT: Semi-supervised vertical federated learning with cross-view training. *ACM Transactions on Intelligent Systems and Technology (TIST)* 13, 4 (2022), 1–16.
- [23] Jakub Konečný, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. 2016. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492* (2016).
- [24] Yann LeCun and Corinna Cortes. 2005. The mnist database of handwritten digits. <https://api.semanticscholar.org/CorpusID:60282629>
- [25] Jingtao Li, Adnan Siraj Rakin, Xing Chen, Zhezhi He, Deliang Fan, and Chaitali Chakrabarti. 2022. Ressfl: A resistance transfer framework for defending model inversion attack in split federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10194–10202.
- [26] Jialin Li, Tongjiang Yan, and Pengcheng Ren. 2023. VFL-R: a novel framework for multi-party in vertical federated learning. *Applied Intelligence* 53, 10 (2023), 12399–12415.
- [27] Qinbin Li, Zeyi Wen, Zhaomin Wu, Sixu Hu, Naibo Wang, Yuan Li, Xu Liu, and Bingsheng He. 2021. A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering* (2021).

- [28] Yang Li, Si Si, Gang Li, Cho-Jui Hsieh, and Samy Bengio. 2021. Learnable fourier features for multi-dimensional spatial positional encoding. *Advances in Neural Information Processing Systems* 34 (2021), 15816–15829.
- [29] Yang Liu, Xinwei Zhang, Yan Kang, Liping Li, Tianjian Chen, Mingyi Hong, and Qiang Yang. 2022. FedBCD: A communication-efficient collaborative learning framework for distributed features. *IEEE Transactions on Signal Processing* 70 (2022), 4277–4290.
- [30] Yang Liu, Xiong Zhang, and Libin Wang. 2020. Asymmetrical vertical federated learning. *arXiv* (2020).
- [31] Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. 2021. Feature inference attack on model predictions in vertical federated learning. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 181–192.
- [32] Ilya Mironov. 2017. Rényi differential privacy. In *2017 IEEE 30th computer security foundations symposium (CSF)*. IEEE, 263–275.
- [33] Vaikkunth Mugunthan, Pawan Goyal, and Lalana Kagal. 2021. Multi-vfl: A vertical federated learning system for multiple data and label owners. *arXiv preprint arXiv:2106.05468* (2021).
- [34] Richard Nock, Stephen Hardy, Wilko Henecka, Hamish Ivey-Law, Jakub Nabaglo, Giorgio Patrini, Guillaume Smith, and Brian Thorne. 2021. The Impact of Record Linkage on Learning from Feature Partitioned Data. In *Proceedings of the 38th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 139)*, Marina Meila and Tong Zhang (Eds.). PMLR, 8216–8226. <https://proceedings.mlr.press/v139/nock21a.html>
- [35] Qichen Qiu. 2017. Kaggle dataset: housing price in Beijing. <https://www.kaggle.com/ruiqurm/lianjia>.
- [36] Xinch Qiu, Heng Pan, Wanru Zhao, Chenyang Ma, Pedro PB Gusmao, and Nicholas D Lane. 2023. vFedSec: Efficient Secure Aggregation for Vertical Federated Learning via Secure Layer. *arXiv preprint arXiv:2305.16794* (2023).
- [37] Salary.sg. 2020. Secondary school rankings in Singapore. <https://www.salary.sg/2020-secondary-schools-ranking-2020-psle-cut-off/>.
- [38] Haoran Shi, Yali Jiang, Han Yu, Yonghui Xu, and Lizhen Cui. 2022. MVFLS: multi-participant vertical federated learning based on secret sharing. *The Federate Learning* (2022), 1–9.
- [39] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*. IEEE, 3–18.
- [40] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.
- [41] Jiankai Sun, Xin Yang, Yuanshun Yao, and Chong Wang. 2022. Label leakage and protection from forward embedding in vertical federated learning. *arXiv preprint arXiv:2203.01451* (2022).
- [42] New York TLC. 2016. TLC trip record data. <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>.
- [43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).
- [44] Dinusha Vatsalan, Ziad Sehili, Peter Christen, and Erhard Rahm. 2017. Privacy-preserving record linkage for big data: Current approaches and research challenges. In *Handbook of Big Data Technologies*. Springer.

- [45] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. 2018. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564* (2018).
- [46] Praneeth Vepakomma, Abhishek Singh, Otkrist Gupta, and Ramesh Raskar. 2020. NoPeek: Information leakage reduction to share activations in distributed deep learning. *ICDM Workshop* (2020).
- [47] Chang Wang, Jian Liang, Mingkai Huang, Bing Bai, Kun Bai, and Hao Li. 2020. Hybrid differentially private federated learning on vertically partitioned data. *arXiv preprint arXiv:2009.02763* (2020).
- [48] Webank. 2023. The application and development report of open-source privacy computing framework (FATE) in financial industry. <https://www.163.com/dy/article/HR7M3K7P055219FH.html>
- [49] Zhaomin Wu, Junyi Hou, and Bingsheng He. 2024. VertiBench: Advancing Feature Distribution Diversity in Vertical Federated Learning Benchmarks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net. <https://openreview.net/forum?id=glwwbaeKm2>
- [50] Zhaomin Wu, Qinbin Li, and Bingsheng He. 2022. A Coupled Design of Exploiting Record Similarity for Practical Vertical Federated Learning. *Advances in Neural Information Processing Systems* 35 (2022), 21087–21100.
- [51] Zhaomin Wu, Qinbin Li, and Bingsheng He. 2022. Practical vertical federated learning with unsupervised representation learning. *IEEE Transactions on Big Data* (2022).
- [52] Yang Yan, Guozheng Yang, Yan Gao, Cheng Zang, Jiajun Chen, and Qiang Wang. 2022. Multi-participant vertical federated learning based time series prediction. In *Proceedings of the 8th International Conference on Computing and Artificial Intelligence*. 165–171.
- [53] Liu Yang, Di Chai, Junxue Zhang, Yilun Jin, Leye Wang, Hao Liu, Han Tian, Qian Xu, and Kai Chen. 2023. A Survey on Vertical Federated Learning: From a Layered Perspective. *arXiv preprint arXiv:2304.01829* (2023).
- [54] Tianyuan Zou, Yang Liu, Yan Kang, Wenhan Liu, Yuanqin He, Zhihao Yi, Qiang Yang, and Ya-Qin Zhang. 2022. Defending batch-level label inference and replacement attacks in vertical federated learning. *IEEE Transactions on Big Data* (2022).