Complementary Knowledge Distillation for Robust and Privacy-Preserving Model Serving in Vertical Federated Learning

Dashan Gao[1,2], Sheng Wan[1,2], Lixin Fan[3], Xin Yao[1], Qiang Yang[2]

[1] Southern University of Science and Technology, Shenzhen, China
[2] Hong Kong University of Science and Technology, Hong Kong SAR, China
[3] Webank AI Lab, Shenzhen, China
dgaoaa@cse.ust.hk, swanae@cse.ust.hk, lixinfan@webank.com, xiny@sustech.edu.cn, qyang@cse.ust.hk

Abstract

Vertical Federated Learning (VFL) enables an active party with labeled data to enhance model performance (utility) by collaborating with multiple passive parties that possess auxiliary features corresponding to the same sample identifiers (IDs). Model serving in VFL is vital for real-world, delay-sensitive applications, and it faces two major challenges: 1) *robustness* against arbitrarily-aligned data and stragglers; and 2) *privacy* protection, ensuring minimal label leakage to passive parties. Existing methods fail to transfer knowledge among parties to improve robustness in a privacy-preserving way. In this paper, we introduce a privacy-preserving knowledge transfer framework, Complementary Knowledge Distillation (CKD), designed to enhance the robustness and privacy of multi-party VFL systems. Specifically, we formulate a Complementary Label Coding (CLC) objective to encode only complementary label information of the active party's local model for passive parties to learn. Then, CKD selectively transfers the CLC-encoded complementary knowledge 1) from the passive parties to the active party, and 2) among the passive parties themselves. Experimental results on four real-world datasets demonstrate that CKD outperforms existing approaches in terms of robustness against arbitrarily-aligned data, while also minimizing label privacy leakage.

Introduction

Vertical Federated Learning (VFL) (Yang et al. 2019) enables global model building among organizations with datasets sharing overlapping samples but differing in features. In VFL, an active party with labeled data aligns samples with passive parties holding auxiliary features. *Model serving* (Wang et al. 2023), the process of inferring a trained machine learning model in a production environment to receive input data and respond with predictions in real time, is particularly challenging in the context of VFL.

Figure 1 illustrates the concept of model serving in a VFL system, highlighting two major challenges: *robustness* against arbitrarily-aligned data, and *label privacy* protection. Robustness primarily involves: 1) maintaining high utility amidst arbitrary feature alignments across multiple parties; and 2) ensuring timely, accurate predictions even with delays from straggling passive parties. For label privacy, the crux lies in ensuring that passive parties cannot infer labels from their own bottom model outputs (Fu et al. 2022).

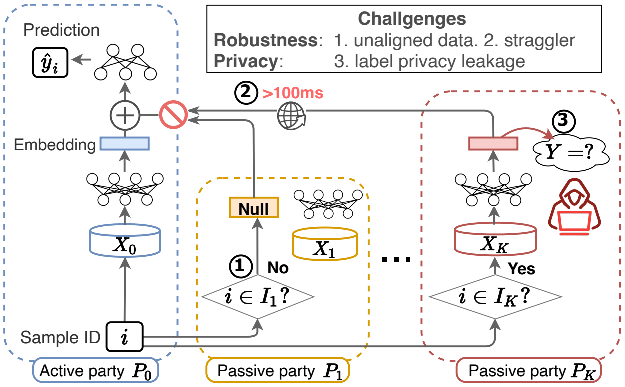
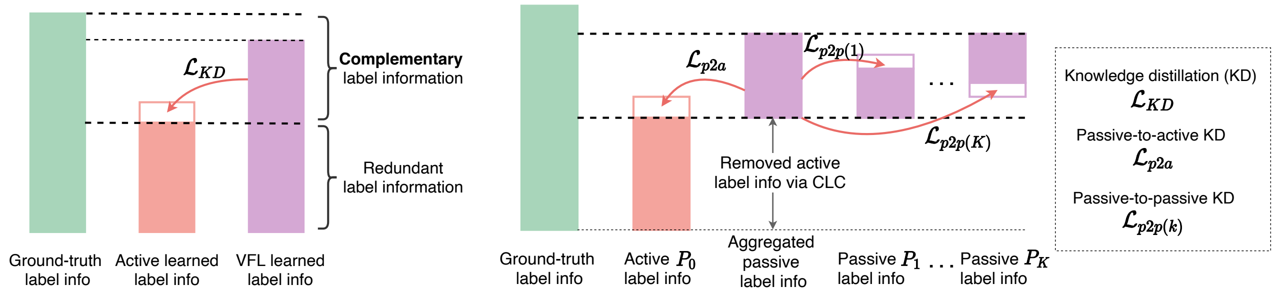


Figure 1: Illustration of model serving in VFL system and the two major challenges: 1) *robustness* against unaligned data and stragglers, and 2) *label privacy* protection.

The robustness and privacy of VFL model serving have recently emerged as a critical focal point of research. **Robustness:** In two-party VFL, knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015) has been employed to enhance robustness against unaligned data (Li et al. 2023; Ren, Yang, and Chen 2022), as shown in Figure 2(a). Based on alignment results, samples are routed to appropriate models (local or VFL model) for inference. However, such ”alignment → routing → inference” paradigm lacks scalability in multi-party settings, given the exponential complexity of candidate models. Meanwhile, the recent party-wise dropout technique (Sun et al. 2023) does not effectively transfer knowledge, leading to inferior utility on unaligned data. **Privacy:** Existing KD-based methods expose redundant label information to passive parties to transfer knowledge, compromising label privacy. While cryptographic protections (Ren, Yang, and Chen 2022) have been proposed, they introduce significant overheads and often fail to meet stringent efficiency requirements. Recent inference-phase protection techniques (Sun et al. 2023; Zou, Liu, and Zhang 2023) tend to reduce the label knowledge learned by passive parties, neglecting to preserve active party's unlearned information for passive parties to learn and transfer. In sum-



(a) Existing methods

(b) Our CKD

Figure 2: Schematic comparison of existing methods with our CKD approach. Horizontally overlapped bars represent shared label information. Our CKD trains passive parties to transfer only *complementary* label information (purple), removing redundant information (coral) learned by the active party from ground-truth labels.

mary, existing methods do not adequately address the dual challenges of robustness and privacy in model serving.

In addressing robustness and privacy, we focus on the knowledge transfer process. While knowledge transfer bolsters VFL robustness, it risks label leakage (Fu et al. 2022). Notably, passive parties should provide what we term as ”*complementary*” label knowledge, which is unlearned by the active party's model, rather than mirroring a superior teacher model. By transferring only this ”complementary” information, we can mitigate privacy risks. This insight, however, introduces two main challenges:

Privacy challenge: How to extract the complementary label information to train passive parties? To address this challenge, we propose a Complementary Label Coding (CLC) approach. The CLC simultaneously 1) minimizes the KL-divergence between the original label and the federated prediction, which integrates the knowledge of active party's local prediction and the passive parties' learning objective, and 2) minimizes the mutual information between the active party's local predictions and the passive parties' learning objective. We find that optimizing the CLC objective can be reduced to *LogitBoost* (Freund and Schapire 1997), a simple and efficient boosting algorithm. LogitBoost converts the original labels into *re-weighted pseudo-residuals* based on the active party's local predictions, thereby eliminating redundant label privacy.

Robustness challenge: How to transfer the complementary knowledge to improve robustness? To enhance robustness against arbitrarily-aligned data, we further propose Complementary Knowledge Distillation (CKD) approach with two strategies: passive-to-active (p2a) distillation and passive-to-passive (p2p) distillation, as shown in Figure 2(b). In p2a distillation, we distill knowledge from passive parties to the active party. Specifically, the teacher model is constructed from the sum of the local model prediction and the federated predicted pseudo-residuals. In p2p distillation, we further distill knowledge from the ensemble of passive parties' bottom models to each bottom model through ensemble distillation (Lin et al. 2020).

We evaluate our CKD approach on four public datasets. In model serving, CKD excels over six baselines, ensuring high utility even with arbitrarily-aligned data, and simultaneously maintaining low label privacy leakage. The key contributions of this work are summarized as follows:

- Introduction of the Complementary Label Coding (CLC) objective, a novel method for dynamically extracting complementary label information for passive parties.
- Development of Complementary Knowledge Distillation (CKD), a technique to transfer knowledge among parties while preserving privacy.
- Comprehensive evaluation of CKD and CLC on four real-world datasets, demonstrating their superior robustness and privacy protection.

The remainder of the paper is structured as follows: We begin by discussing related works. Then, we provide the problem formulation and the robustness and privacy concepts. Next, we propose the CLC method. Subsequently, we detail our CKD approach, and finally, we present experimental evaluations of CKD.

Related Work

Robustness against Unaligned Data in VFL. Recently, the robustness of VFL model serving has attracted growing attentions. SplitKD (Li et al. 2023) and (Ren, Yang, and Chen 2022; Wan et al. 2023) are the pioneer work that use knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015) to transfer knowledge from the VFL model to the active party's local model. Moreover, these methods require $2^{K-1} - 1$ candidate VFL models for all arbitrarily-aligned data when scale to K-parties. There are also many studies using ensemble distillation methods (Lin et al. 2020) to improve robustness in horizontal FL. However, these methods are not applicable to VFL due to serious privacy leakage issues (Fu et al. 2022). Party-wise dropout (PtyDrop) (Sun et al. 2023) randomly dropout some passive parties during training, without knowledge transfer. However, PtyDrop leads to even inferior utility than standalone local model on unaligned data. Therefore, it is still an open challenge to design a privacy-preserving knowledge transfer method to improve robustness in multi-party VFL.

Knowledge transfer in VFL is also related to learning using privileged features (Vapnik and Izmailov 2015; Vapnik and Vashist 2009), which are available only in training stage but unavailable in test stage. TVFL (Wang et al. 2023) actively filters out passive parties to improve efficiency on *fully* aligned data. In contrast, robust VFL aims to achieve the highest possible utility on *arbitrarily-aligned* test data.

Privacy-preserving VFL. Existing privacy protection approaches in VFL (Gao, Yao, and Yang 2022) include cryptographic methods (Ren, Yang, and Chen 2022; Fu et al. 2021; Gao et al. 2019) and perturbation methods (Sun et al. 2022). Cryptographic methods incur high communication and computation overheads, thus are unbearable in multi-party VFL settings. (Ren, Yang, and Chen 2022) integrates cryptographic methods with KD to protect privacy in training and inference. Perturbation methods mainly focus on protecting label leakage from gradients during training (Li et al. 2022), rather than model serving. (Sun et al. 2022) consider model serving in split learning. But they fail to remove the redundant label information learned by the active party. Some recent works (Sun et al. 2023; Zou, Liu, and Zhang 2023) incorporate a loss term to minimize the mutual information between the outputs of the passive party's model and the labels, manually tuning the loss weight to balance the privacy-utility trade-off. In contrast, we explicitly define a *distinct* learning target to train passive parties, ensuring *zero* mutual information between the active party's model output and the learning targets of passive parties.

Problem Description

Problem Formulation

Vertical Federated Learning Setting. In a typical VFL setting, the training data $\mathcal{D}$ has n samples with sample identifiers (IDs) I and labels Y. The feature space $\mathcal{X} = \mathcal{X}_0 \times \mathcal{X}_1 \cdots \mathcal{X}_K$ is partitioned among $K+1$ parties by feature. An active party P_0 has labeled local features $\{\boldsymbol{x}_0, \boldsymbol{y}\}$. Meanwhile, K passive parties $\{P_k\}_{k=1}^{K}$ only have auxiliary features $\{\boldsymbol{x}_k\}_{k=1}^{K}$.

As shown in Figure 3, our CKD framework predicting a sample with ID $i \in I$ aligned among a set of passive parties $\mathcal{K} \subseteq \{1, \ldots, K\}$ can be expressed as:

$$f_{CKD}(i) = Merge\Big(f_\theta(i), g_\lambda\big(\{h_{\psi_k}(i)\}_{k\in\mathcal{K}}\big)\Big), \quad (1)$$

where the active party P_0 trains a local model $f_\theta : \mathcal{X}_0 \mapsto \mathcal{Y}$ to predict labels. Each passive party P_k trains a bottom model $h_{\psi_k} : \mathcal{X}_k \mapsto \mathcal{Y}$ to learn the complementary label knowledge (i.e., pseudo-residual) of f_θ. A top model $g_\lambda : \mathcal{Y}^K \mapsto \mathcal{Y}$ is trained to robustly aggregate the outputs of available bottom models. Finally, $Merge : \mathcal{Y}^2 \mapsto \mathcal{Y}$ merges the local prediction and federated pseudo-residual to make prediction. For simplicity, we use sample ID i to index each sample. That is, $f_\theta(i)$ and $h_{\psi_k}(i)$ denote $f_\theta(\boldsymbol{x}_{0,i})$ and $h_{\psi_k}(\boldsymbol{x}_{k,i})$, respectively.

Threat model. We focus on the risk of label privacy leakage during the model serving stage, where this risk originates from the output of the passive parties' bottom models (Fu et al. 2022). We assume that these passive parties are *semi-honest* and do not collude, signifying that they adhere to the protocol but may attempt to extract private information from the data available to them. Specifically, an adversarial passive party P_k seeks to infer the raw label y_i from its model's output $h_{\psi_k}(i)$, given its features $\boldsymbol{x}_{k,i} \in \mathcal{X}_k$.

Notably, under the semi-honest threat model, *we don't account for Byzantine attackers* (Zhang et al. 2015), which involve malicious parties intentionally trying to disrupt or deceive the system, thereby excluding considerations for robustness against such Byzantine attacks.

Robustness and Privacy

The dataset $\mathcal{D}$ has a **g**round-**t**ruth ID-label joint distribution $p_{gt}(i, y)$, with uniform sample weight $p_{gt}(i) \sim U$ and private label $p_{gt}(y|i) = 1(y = y_i)$. We use the standard error to measure the utility and label privacy leakage of a model:

Definition 1 (Standard error). *Given a dataset with* ***g****round-****t****ruth distribution $p_{gt}(i, y)$, let $\mathbf{KL}(\cdot||\cdot)$ denote the KL-divergence, the standard error of a model f is defined as:*

$$\mathbb{R}_{p_{gt}(i,y)}(f) = \mathbb{E}_{i\sim p_{gt}(i)}[\mathbf{KL}(p_{gt}(y|i)||f(i))].$$

Robustness Metric. In the model serving stage, we introduce a robustness metric to quantify the performance consistency of the federated model f_{CKD} defined in Eq. 1 across different subsets of passive parties $\mathcal{K}$. The robustness metric for the model is conceptualized as:

$$\sum_{\mathcal{K}\subseteq\{1,\ldots,K\}} \mathbb{R}_{p_{gt}(i,y)}(f_{CKD}) \quad \textbf{(Robustness Metric)}$$

This metric captures the aggregated performance of the model over all possible combinations of passive parties, aiming to provide a comprehensive measure of its robustness in diverse settings. However, it's worth noting that *directly optimizing this metric can be challenging due to its combinatorial nature*. Thus, while it serves as a conceptual guide to understand robustness, our subsequent methods and discussions do not directly optimize this exact objective.

Privacy Metric. The essence of privacy in our context revolves around the label information of a dataset. According to Definition 1, we define the private label information as:

Definition 2 (Private label information). *The private label information of a dataset is defined as its ID-label joint-distribution $p_{gt}(i, y)$.*

To protect label privacy, the active party trains the passive parties' models using a *distinct* distribution $p_{pas}(i, y)$, different from $p_{gt}(i, y)$. The goal is to ensure that the passive parties' models do not inadvertently leak sensitive label information. A natural metric to capture this privacy leakage is the mutual information (MI) between $p_{gt}(i, y)$ and $p_{pas}(i, y)$:

$$I(p_{gt}(i, y); p_{pas}(i, y)) \quad \textbf{(Privacy Metric)}$$

However, if this MI is too low (e.g., 0), it implies that the passive parties gain minimal label knowledge from $p_{pas}(i, y)$, which could adversely affect the VFL utility. Instead of directly minimizing this MI, **it's crucial to retain label knowledge that the active party's local model f_θ hasn't yet learned**. This insight motivates our proposal of the complementary label coding, which we detail in the subsequent section.

Complementary Label Coding

Motivated by the identified limitations, we introduce the concept of *Complementary Label Coding (CLC)*, to decouple the label privacy $p_{gt}(i,y)$ into two distinct components:

1. The *redundant* label information $p_{act}(i,y) = p_{gt}(i) \cdot p_{act}(y|i)$, which is already captured by the local model $f_\theta = p_{act}(y|i)$.
2. The *complementary* label information $p_{clc}(i,y)$ that the local model has yet to learn.

Therefore, the CLC objective is defined to optimize the complementary label information $p_{clc}(i,y)$ as follows:

$$\min_{p_{clc}(i,y)} \mathbb{E}_{i\sim p_{gt}(i)}[\mathbf{KL}(p_{gt}(y|i)||p_{fed}(y|i))] \;\textbf{(Utility)}, \quad (2)$$

$$s.t. \;\; p_{fed}(y|i) = Merge(f_\theta(i), h^*_{pas}(i)), \quad (3)$$

$$h^*_{pas}(i) = \arg\min_{h_{pas}} \mathbb{E}_{i\sim p_{clc}(i)}[\mathbf{KL}(p_{clc}(y|i)||h_{pas}(i))],$$

$$I(p_{act}(i,y); p_{clc}(i,y)) = 0 \qquad \textbf{(Privacy)}. \quad (4)$$

Remark: Eq. 2 and Eq. 3 aim to minimize the KL-divergence between the original label $p_{gt}(y|i)$ and the federated prediction $p_{fed}(y|i)$, which integrates both the local prediction and the passive parties' learning objective via $Merge()$. Meanwhile, Eq. 4 ensures that $f_\theta(i)$ and $h^*_{pas}(i)$ share no mutual information, making them independent.

Interestingly, we find that the CLC objective can be reduced to the LogitBoost (Freund and Schapire 1997) objective, as shown in Proposition 1.

Proposition 1. *Given $p_{gt}(i) \sim U$ is a uniform distribution, ground-truth label $y_i = p_{gt}(y|i)$, local model output logit $f_\theta(i)$, and the expected passive model output $h^*_{pas}(i)$. The original CLC objective in Eq. 2 is equivalent to LogitBoost (Freund and Schapire 1997) objective:*

$$\min_{p_{clc}(i,y)} \sum_{i=1}^{n} \frac{1}{n} \mathcal{L}_{CE}(y_i, \; f_\theta(i) + h^*_{pas}(i)), \quad (5)$$

where $\mathcal{L}_{CE}(y,y') = \ln(1+\exp(-y\cdot\sigma(y')))$ is the cross-entropy loss taking logit y' as input. $\sigma()$ is softmax function.

Proof sketch: According to (Zhang 2004), for independent input features f_θ and h^*_{pas} constrained in Eq. 4, the optimal $Merge(\cdot,\cdot)$ is a linear, naive Bayes classifier. That is,

$$Merge^*(f_\theta(i), h^*_{pas}(i)) = \sigma(a\cdot f_\theta(i) + b\cdot h^*_{pas}(i) + c).$$

As the local model $f_\theta(i)$ is trained to fit $p_{gt}(y|i)$ in prior, we have $a^* = 1$. Meanwhile, we fix $[b,c] = [1,0]$ to optimize the corresponding $h^*_{pas}(i)$. Therefore, the CLC objective Eq. 2 can be reformulated as:

$$\min_{p_{clc}(i,y)} \mathbb{E}_{i\sim p_{gt}(i)}[\mathbf{KL}\left(p_{gt}(y|i)||\sigma(f_\theta(i) + h^*_{pas}(i))\right)],$$

which can further be reduced to Eq. 5. □

Therefore, we use the Newton method to optimize Eq. 5 and get the optimized $p_{clc}(i,y)$ as follows:

Proposition 2. *(Freund and Schapire 1997) Given $\hat{y}_{0,i} = \sigma(f_\theta(i))$ is the locally predicted probability. Using the Newton method, the optimization result $p_{clc}(i,y)$ of Eq. 5 is:*

$$p_{clc}(i) = \frac{\hat{y}_{0,i}(1-\hat{y}_{0,i})}{\sum_{j=1}^{n} \hat{y}_{0,j}(1-\hat{y}_{0,j})}, \; p_{clc}(y|i) = \frac{y_i - \hat{y}_{0,i}}{\hat{y}_{0,i}(1-\hat{y}_{0,i})}. \quad (6)$$

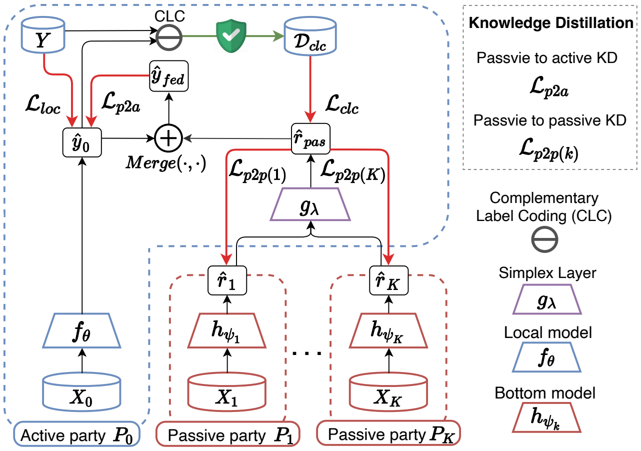


Figure 3: Complementary Knowledge Distillation (CKD) training overview. For simplicity, we use the output logit and probability interchangeably.

We denote $\mathcal{D}_{clc} = (\boldsymbol{w}, \boldsymbol{r})$ as the CLC-encoded private label information, where $\boldsymbol{w} = p_{clc}(i)$ is *sample weights* and $\boldsymbol{r} = p_{clc}(y|i)$ is *pseudo-residuals*. The label privacy leakage risk of CLC is guaranteed as follows:

Theorem 1 (Privacy guarantee)**.** *When the standard error of the local model f_θ trained on $p_{gt}(i,y)$ satisfies $\mathbb{R}_{p_{gt}(i,y)}(f_\theta) \to 0$, the privacy leakage from CLC-encoded results $p_{clc}(i,y)$ satisfies $I(p_{gt}(i,y); p_{clc}(i,y)) \to 0$.*

As the standard error of the active party's local model approaches zero, the label leakage from the CLC results also nears zero.

Our Proposed Approach

Based on the concept of *Complementary Label Coding (CLC)* introduced in the previous section, we present our Complementary Knowledge Distillation (CKD) framework designed to enhance the robustness of multi-party VFL model serving, as illustrated in Figure 3.

1) The architecture of CKD is structured as follows: the active party trains a standalone local model f_θ tailored to fit labels, while passive parties train an ensemble model h_{pas} to fit the CLC-encoded *re-weighted pseudo-residuals* $D_{clc} = (\boldsymbol{w}, \boldsymbol{r})$. A specially designed simplex layer g_λ is incorporated to aggregate the outputs of available bottom models h_{ψ_k} in a robust manner. 2) Subsequently, we detail the online complementary knowledge distillation process, which dynamically transfers the CLC-encoded label knowledge both to the active party and among passive parties.

Framework Architecture

Active Party's Local Model. The active party P_0 trains a standalone local model f_θ on its local dataset $\{\boldsymbol{x}_0, \boldsymbol{y}\}$ as follows:

$$\mathcal{L}_{loc} = \sum_{i=1}^{n} \frac{1}{n} \mathcal{L}_{CE}\big(y_i, f_\theta(i)\big),$$

where $f_\theta(i)$ is the output logit of the i-th sample $\boldsymbol{x}_i$, $\mathcal{L}_{CE}$ is the cross-entropy loss taking logit $f_\theta(i)$ as input.

Ensemble of Passive Parties' Models. As shown in Figure 3 (middle part), all passive parties collaboratively train a federated ensemble model $h_{pas} = g_\lambda \circ \{h_{\psi_k}\}_{k=1}^K$ to fit the CLC-encoded *re-weighted pseudo-residuals* $\mathcal{D}_{clc} = (\boldsymbol{w}, \boldsymbol{r})$. Each passive party P_k trains its bottom model h_{ψ_k}. Then, the active party trains a *simplex layer* g_λ to robustly aggregate the outputs of the available bottom models h_{ψ_k}. In all, the ensemble of passive parties' models h_{pas} is as follows:

$$h_{pas}(i) = g_\lambda \circ \{h_{\psi_k}\}_{k\in\mathcal{K}}(i) = \frac{\sum_{k\in\mathcal{K}} \lambda_k \cdot h_{\psi_k}(i)}{\sum_{k\in\mathcal{K}} \lambda_k} \qquad (7)$$
$$s.t.\ \ \lambda_k \geq 0,\ \ \forall k \in [1, K],$$

where $\mathcal{K}$ denotes the set of available passive parties of sample i, and $\sum_{k\in\mathcal{K}} \lambda_k$ serves as a normalization factor. We constrain $\{\lambda_k\}_{k\in\mathcal{K}} = \Delta^{|\mathcal{K}|}$ to be a simplex (i.e., one-sum, non-negative) to ensure the simplex layer g_λ is robust to different number of available parties.

To protect label privacy, the passive parties only learn the CLC-encoded complementary label information. Therefore, the objective of the ensemble models is as follows:

$$\mathcal{L}_{clc} = \sum_{i=1}^{n} w_i \cdot ||r_i - h_{pas}(i)||_2^2,$$

where $(w_i, r_i) \in \mathcal{D}_{clc}$ is the CLC-encoded weight and pseudo-residual of sample i.

Overall CKD Model. In summary, the CKD model f_{CKD} can be expressed as the sum of active party's local predictions and the passive parties' predicted pseudo-residuals:

$$f_{CKD}(i) = f_\theta(i) + \alpha \cdot h_{pas}(i), \qquad (8)$$

where $\alpha > 0$ is the weight of the predicted pseudo-residual. The predicted probability $\hat{y}_{fed,i} = \sigma(f_{CKD}(i))$, where $\sigma(\cdot)$ is the softmax function.

Online Knowledge Distillation

The federated model f_{CKD} outperforms the local model f_θ alone by utilizing the predicted pseudo-residuals from passive parties to rectify the imprecise predictions of the local model. Therefore, we adopt the federated model f_{CKD} as the teacher model and employ knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015) to transfer knowledge from f_{CKD} to the local model f_θ. The *passive-to-active (p2a)* distillation loss is defined based on KL-divergence as follows:

$$\mathcal{L}_{p2a} = T^2 \sum_{i=1}^{n} w_i \cdot \mathbf{KL}(\sigma(f_{CKD}(i)/T)||\sigma(f_\theta(i)/T)),$$

where $T > 1$ is the temperature to generate softened labels, $\sigma(\cdot)$ is the softmax function, $f_{CKD}(i)$ and $f_\theta(i)$ are the logit of the federated model and the local model, respectively.

To further improve utility on partially-aligned data, we conduct *passive-to-passive (p2p)* distillation by adopting ensemble distillation (Lin et al. 2020) to transfer knowledge from the ensemble of bottom models to each bottom model:

$$\mathcal{L}_{p2p(k)} = T^2 \sum_{i=1}^{n} w_i \cdot \mathbf{KL}(\sigma(h_{pas}(i)/T)||\sigma(h_{\psi_k}(i)/T)),$$

Algorithm 1 CKD: Training

```
Require: Aligned data $\mathcal{D} = \{\{\boldsymbol{x}_k\}_{k=1}^K, \boldsymbol{y}\}$ indexed by $I$.
    ▷ Cold start
  Active party $P_0$ trains $f_\theta$ on $\{\boldsymbol{x}_0, \boldsymbol{y}\}$ via $\mathcal{L}_{loc}$.
    ▷ Federated complementary knowledge distillation
  Passive parties $\{P_k\}_{k=1}^K$ initialize $\{\psi_k\}_{k=1}^K$.
  for each batch of sample ID $\boldsymbol{b} \subset I$ do
      $P_0$ updates $\mathcal{D}_{clc} = (\boldsymbol{w}, \boldsymbol{r})$ via Eq. 6.
      ▷ Loss Computation
      $\{P_k\}_{k=1}^K$ compute $\{h_{\psi_k}(\boldsymbol{b})\}_{k=1}^K$, send to $P_0$.
      $P_0$ computes $f_{CKD}(i)$ via Eq. 7 and Eq. 8.
      $P_0$ computes $\mathcal{L}_{loc}, \mathcal{L}_{clc}, \mathcal{L}_{p2a}$, and $\mathcal{L}_{p2p}$.
      ▷ Model Update
      $P_0$ updates simplex layer $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \frac{\partial \mathcal{L}_{clc}}{\partial \boldsymbol{\lambda}}$.
      $P_0$ updates local model $\theta \leftarrow \theta - \frac{\partial \mathcal{L}_{act}}{\partial \theta}$.
      $\{P_k\}_{k=1}^K$ update models $\psi_k \leftarrow \psi_k - \frac{\partial \mathcal{L}_{pas(k)}}{\partial \psi_k}$.
  end for
Ensure: $\theta$, $\{\psi_k\}_{k=1}^K$, and $\boldsymbol{\lambda}$.
```

Algorithm 2 CKD: Robust Model Serving

```
Require: Sample ID $i$, timeout delay $t$, each party $P_k$ has
      test set with IDs $I_k$, the trained CKD model $f_{CKD}$.
  Active party $P_0$ broadcasts sample ID $i$ to $\{P_k\}_{k=1}^K$.
  Meanwhile, $P_0$ locally predicts $f_\theta(i)$.
  $\{P_k\}_{k=1}^K$ send back $h_{\psi_k}(i)$ if $i \in I_k$ else $Null$.
  $P_0$ waits until timeout $t$ or receives all $\{h_{\psi_k}(i)\}_{k=1}^K$.
  $P_0$ computes $f_{CKD}(i)$ following Eq. 7 and Eq. 8.
Ensure: $\hat{y}_{fed,i} = \sigma(f_{CKD}(i))$
```

Model Training. Algorithm 1 demonstrates the training process of the proposed CKD. To update the local model, the total loss of the local model f_θ is formulated as:

$$\mathcal{L}_{act} = \mathcal{L}_{loc} + \beta \cdot \mathcal{L}_{p2a},$$

where $\beta > 0$ is a coefficient that determines the weight of knowledge distillation loss. The total loss of each passive party P_k's bottom model h_{ψ_k} is defined as follows:

$$\mathcal{L}_{pas(k)} = \mathcal{L}_{clc} + \beta \cdot \mathcal{L}_{p2p(k)}.$$

Robust Model Serving

Algorithm 2 illustrates the model serving process of CKD when dealing with arbitrarily-aligned data. Initially, the active party P_0 broadcasts the sample ID i to all passive parties $\{P_k\}_{k=1}^K$. In response, each passive party P_k checks if the sample i exists within its ID set I_k. If found, it computes and returns the prediction $h_{\psi_k}(i)$; otherwise, it sends back a $Null$ response. Concurrently, P_0 performs a local inference to obtain $f_\theta(i)$ using its local model. Once all responses are gathered or a timeout is reached, P_0 calculates the federated prediction $\hat{y}_{fed,i} = \sigma(f_{CKD}(i))$ based on Eq. 7 and Eq. 8.

Experimental Studies

In our experimental evaluation, we seek to address two primary research questions: **RQ1**: How do complementary knowledge distillation and the simplex layer in CKD enhance the robustness of VFL compared to existing methods? **RQ2**: Is CKD capable of providing stronger label privacy protection compared to previous methods?

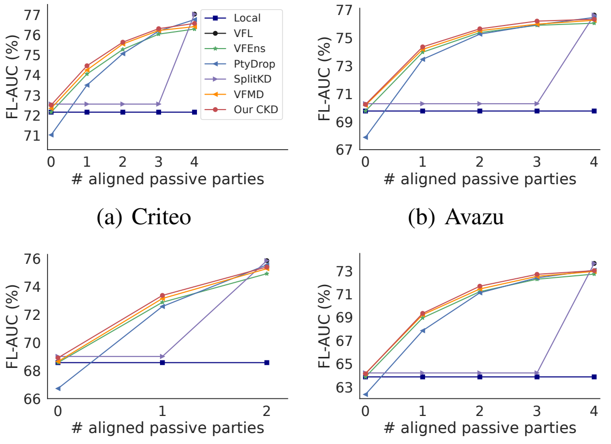
Experimental Setting

Datasets. We evaluate CKD using four widely used public real-world datasets spanning various domains: two for click-through rate (CTR) prediction, one for movie ratings, and one for healthcare. 1) **Criteo**[1]: A dataset containing a month's worth of ad click records with 13 numerical and 26 categorical features. Features are randomly distributed among one active party and four passive parties. 2) **Avazu**[2]: Comprising 21 categorical fields, this dataset's fields are randomly distributed among five parties. We use a subset of 10 million records for both datasets. 3) **HetRec** (Cantador, Brusilovsky, and Kuflik 2011): A movie rating dataset linking the MovieLens10M dataset (Harper and Konstan 2015) to RottenTomatoes reviews. Ratings are binarized using a threshold of 2.5. Features are distributed among one active party and two passive parties. 4) **MIMIC-III** (Johnson et al. 2016): A dataset for predicting in-hospital mortality based on the initial 48 hours of ICU data, containing 714 features. Features are distributed among five parties.

Implementations. For all datasets, we randomly sample 80% data for training and the rest for testing. We adopt the widely used DeepFM (Guo et al. 2017) for both local and bottom models on Criteo, Avazu and HetRec. We use a 3-layer MLP for both local and bottom models on MIMIC-III. The models are optimized by Adam (Kingma and Ba 2015). We set the learning rate to $1e-4$, the weight decay to $1e-4$, the passive ensemble model weight α to 1.5, and the batch size to 2048. We use 5-fold validation to determine early stopping. All training data are fully-aligned. For KD, we set the temperature T to 20 and KD loss weight β to 3.

Compared methods. We compare our CKD with six other methods in experiments. Cryptographic approaches (Ren, Yang, and Chen 2022) are not included due to their expensive communication and computational costs.

- **Local** model is only trained on the local data $\{\boldsymbol{x}_0, \boldsymbol{y}\}$.
- **Vanilla VFL** (Yang et al. 2019) trains a federated model and tests on fully-aligned data.
- **VFEns** trains ensemble of models on all parties via split learning (Vepakomma et al. 2018). VFEns trains each $h_{\psi(i)}$ to fit labels via split learning and averages the predictions of available models.
- **PtyDrop** (Sun et al. 2023) randomly dropout passive parties in training for robustness against unaligned data. However, it does not transfer knowledge between parties.
- **SplitKD** (Li et al. 2023) distills knowledge from a federated model trained on fully-aligned data to a local model trained on the active party's local data.



(a) Criteo

(b) Avazu

(c) HetRec

(d) MIMIC-III

Figure 4: The robustness against various unaligned parties.

- **VFMD** is an ensemble distillation method adopted from FedDF (Lin et al. 2020) in horizontal FL. All parties train their local models for label prediction and distill knowledge from the ensemble to each model.

Metrics. We use the AUC (Area Under ROC curve) metric in our experiments. 1) **Utility**: We evaluate the AUC of the federated model on arbitrarily-aligned test data with various number of aligned passive parties. Higher AUC values indicate superior model utility. 2) **Privacy**: For privacy evaluation, we calculate the average AUC of the label predictions made by the passive parties via Passive Model Completion (PMC) attack (Fu et al. 2022). An ideal privacy leakage AUC value is close to 0.5.

Robustness

To assess the robustness of CKD during model serving, we simulate scenarios with arbitrarily-aligned data involving varying numbers of aligned passive parties. This setup mirrors real-world situations where the alignment of passive parties can significantly influence the system's utility. Specifically, we emulate the absence of different passive parties and compare the federated model's AUC (FL-AUC) across all methods using four datasets.

Figure 4 shows that CKD achieves the best utility on partially-aligned data, and matches the top-performing methods on unaligned data. Notably, the distillation-based methods, namely CKD, SplitKD, and VFMD, demonstrate superiority over other baselines like VFEns and PtyDrop when data is partially aligned, attributed to their ability to transfer knowledge from passive to active parties. For SplitKD, the reliance on a standalone local model for partially-aligned data compromises its utility. Conversely, PtyDrop's performance diminishes when data is either unaligned or aligned with only a few passive parties. Although VFMD exhibits robustness close to CKD, a significant concern arises with VFMD, which we will address later: it poses substantial label privacy risks since it directly trains passive parties on labels.

[1] https://labs.criteo.com/category/dataset/

[2] https://www.kaggle.com/c/avazu-ctr-prediction

Method	Criteo			Avazu			HetRec			MIMIC-III		
	L-Util↑	P-Util↑	Priv↓	L-Util↑	P-Util↑	Priv↓	L-Util↑	P-Util↑	Priv↓	L-Util↑	P-Util↑	Priv↓
Local	72.2	-	-	69.8	-	-	68.6	-	-	63.9	-	-
VFEns	72.2	74.1	72.7	69.8	74.0	71.2	68.6	72.9	69.0	63.9	69.0	64.9
PtyDrop	70.8	73.5	62.4	67.9	73.5	60.4	66.7	72.6	61.6	62.4	67.8	57.3
SplitKD	**72.6**	72.6	70.5	70.2	70.2	69.8	**69.0**	69.0	69.2	**64.2**	64.2	65.1
VFMD	72.4	74.3	73.2	70.2	74.2	72.5	68.8	73.1	69.4	64.2	69.2	65.3
(Our) CKD	72.5	**74.5**	**59.7**	70.2	**74.3**	**57.9**	68.9	**73.4**	**60.6**	64.1	**69.3**	**56.7**

Table 1: The comparative results of utility and privacy on four datasets. *L-Util* and *P-Util* indicate the AUC (%) on active party's **l**ocal data $\boldsymbol{x}_0$ and **p**artially-aligned data $\{\boldsymbol{x}_0, \boldsymbol{x}_k\}_{k\in K}$, respectively. *Priv* is the privacy leakage AUC (%) of bottom models.

Loss		Criteo			Avazu		
$\mathcal{L}_{p2a}$	$\mathcal{L}_{p2p}$	L-Util↑	P-Util↑	Priv↓	L-Util↑	P-Util↑	Priv↓
×	×	72.1	74.2	59.6	69.8	74.0	57.8
✓	×	72.5	74.4	**59.4**	70.1	74.2	**57.6**
✓	✓	72.5	**74.5**	59.7	**70.2**	**74.3**	57.9

Table 2: Impact of $\mathcal{L}_{p2a}$ and $\mathcal{L}_{p2p}$ on CKD. *L-Util* and *P-Util* denote the AUC (%) on **l**ocal data $\boldsymbol{x}_0$ and **p**artially-aligned data $\{\boldsymbol{x}_0, \boldsymbol{x}_k\}_{k\in K}$, respectively. *Priv* is the privacy leakage AUC (%) of bottom models against the PMC attack.

	# Aligned passive parties			
Top model	1	2	3	4
Averaging Layer	74.3	75.1	75.8	76.2
Simplex Layer	**74.5**	**75.6**	**76.2**	**76.6**

Table 3: Comparative AUC (%) results of different top models for robust aggregation in CKD on the Criteo dataset.

Label Privacy Protection

We assess the label privacy protection capabilities of CKD. Table 1 presents the label privacy leakage AUC of CKD and other baselines when subjected to the PMC attack (Fu et al. 2022), alongside their utility on unaligned and partially-aligned data. We have *excluded* Vanilla VFL from this table because it's inapplicable to partially aligned data, and its privacy leakage AUC mirrors that of the SplitKD model. Notably, CKD stands out by offering effective knowledge transfer from passive parties to the other parties, achieving comparable local utility (L-Util) with SplitKD and the highest utility on partially-aligned data (P-Util), and significantly enhancing label privacy protection against passive parties. While PtyDrop offers similar label privacy protection, it falls short in knowledge transfer, resulting in a local utility even lower than the Local baseline. Therefore, CKD not only excels in knowledge transfer but also demonstrates superior label privacy protection compared to other baselines, reinforcing its position as a leading solution for privacy-preserving knowledge transfer in multi-party VFL systems.

Additional Experiments

Effect of knowledge transfer losses. We evaluate the impact of passive-to-active knowledge transfer loss $\mathcal{L}_{p2a}$ and passive-to-passive loss $\mathcal{L}_{p2p}$ on CKD's performance. Table 2 illustrates that both losses enhance CKD's efficacy. Specifically, 1) introducing $\mathcal{L}_{p2a}$ (as seen by comparing the first two rows) boosts local utility while enhancing label privacy. This enhancement arises as knowledge is transferred from passive to active parties, refining local utility and concurrently reducing encoded complementary label information in passive models, thereby protecting label privacy. 2) Introducing $\mathcal{L}_{p2p}$ (evident from the last two rows) further elevates local utility, albeit with a slight compromise in label privacy. This is attributed to the enhanced knowledge sharing among passive parties, which, while improving utility, slightly amplifies label privacy leakage.

Impact of the Simplex Layer We evaluate the role of the simplex layer g_λ in robustly aggregating against unaligned features. Table 3 compares the utility across different numbers of aligned passive parties, contrasting the simplex layer with an averaging layer. The results reveal that the simplex layer consistently achieves higher utility across varying numbers of aligned passive parties. This superior performance is attributed to the simplex layer's ability to effectively discern the contributions of passive parties and integrate the complementary label information into the simplex space. Consequently, the model's robustness against missing parties is improved.

Conclusions

In this work, we introduced Complementary Knowledge Distillation (CKD), a novel approach designed to enhance both the robustness and privacy of multi-party Vertical Federated Learning (VFL) against arbitrarily-aligned data in model serving. Our method begins with the formulation of a Complementary Label Coding (CLC) technique, which encodes the complementary label information that is unlearned by the active party's local model. Subsequently, CKD is proposed to distill this complementary knowledge both to the active party's local model and among the passive parties. Experimental validation on four public datasets confirms the effectiveness of CKD in bolstering the robustness of multi-party VFL, while also maintaining label privacy. Looking ahead, our future work will delve into the convergence analysis of CKD and explore the integration of CKD with other protection mechanisms, such as cryptographic methods, to further protect label privacy during the training phase.

Acknowledgements

We thank the anonymous reviewers/SPC/AC for the constructive comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant No.62250710682), Guangdong Provincial Key Laboratory (Grant No.2020B121201001), the Program for Guangdong Introducing Innovative and Entrepreneurial Teams (Grant No. 2017ZT07X386), and Research Institute of Trustworthy Autonomous Systems (RITAS).

References

Cantador, I.; Brusilovsky, P.; and Kuflik, T. 2011. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *Proceedings of the 5th ACM conference on Recommender systems*, RecSys 2011. New York, NY, USA: ACM.

Freund, Y.; and Schapire, R. E. 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1): 119–139.

Fu, C.; Zhang, X.; Ji, S.; Chen, J.; Wu, J.; Guo, S.; Zhou, J.; Liu, A. X.; and Wang, T. 2022. Label Inference Attacks Against Vertical Federated Learning. In *31st USENIX Security Symposium (USENIX Security 22)*, 1397–1414. Boston, MA: USENIX Association. ISBN 978-1-939133-31-1.

Fu, F.; Shao, Y.; Yu, L.; Jiang, J.; Xue, H.; Tao, Y.; and Cui, B. 2021. VF2Boost: Very Fast Vertical Federated Gradient Boosting for Cross-Enterprise Learning. In *Proceedings of the 2021 International Conference on Management of Data*, 563–576.

Gao, D.; Liu, Y.; Huang, A.; Ju, C.; Yu, H.; and Yang, Q. 2019. Privacy-preserving Heterogeneous Federated Transfer Learning. In *2019 IEEE International Conference on Big Data (Big Data)*, 2552–2559.

Gao, D.; Yao, X.; and Yang, Q. 2022. A Survey on Heterogeneous Federated Learning. arXiv:2210.04505.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. DeepFM: A Factorization-Machine Based Neural Network for CTR Prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, 1725–1731. ISBN 9780999241103.

Harper, F. M.; and Konstan, J. A. 2015. The Movielens Datasets: History and context. *ACM transactions on interactive intelligent systems (TIIS)*, 5(4): 1–19.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Lehman, L.-w. H.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Anthony Celi, L.; and Mark, R. G. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data*, 3(1): 1–9.

Kingma, D.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*. San Diega, CA, USA.

Li, O.; Sun, J.; Yang, X.; Gao, W.; Zhang, H.; Xie, J.; Smith, V.; and Wang, C. 2022. Label Leakage and Protection in Two-party Split Learning. *International Conference on Learning Representations (ICLR)*.

Li, W.; Xia, Q.; Deng, J.; Cheng, H.; Liu, J.; Xue, K.; Cheng, Y.; and Xia, S.-T. 2023. VFed-SSD: Towards Practical Vertical Federated Advertising. arXiv:2205.15987.

Lin, T.; Kong, L.; Stich, S. U.; and Jaggi, M. 2020. Ensemble Distillation for Robust Model Fusion in Federated Learning. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2351–2363. Curran Associates, Inc.

Ren, Z.; Yang, L.; and Chen, K. 2022. Improving Availability of Vertical Federated Learning: Relaxing Inference on Non-overlapping Data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4): 1–20.

Sun, J.; Du, Z.; Dai, A.; Baghersalimi, S.; Amirshahi, A.; Atienza, D.; and Chen, Y. 2023. Robust and IP-Protecting Vertical Federated Learning against Unexpected Quitting of Parties. *arXiv preprint arXiv:2303.18178*.

Sun, J.; Yang, X.; Yao, Y.; and Wang, C. 2022. Label Leakage and Protection From Forward Embedding in Vertical Federated Learning. *arXiv preprint arXiv:2203.01451*.

Vapnik, V.; and Izmailov, R. 2015. Learning Using Privileged Information: Similarity Control and Knowledge Transfer. *J. Mach. Learn. Res.*, 16(1): 2023–2049.

Vapnik, V.; and Vashist, A. 2009. A New Learning Paradigm: Learning Using Privileged Information. *Neural networks*, 22(5-6): 544–557.

Vepakomma, P.; Gupta, O.; Swedish, T.; and Raskar, R. 2018. Split Learning for Health: Distributed Deep Learning Without Sharing Raw Patient Data. *arXiv preprint arXiv:1812.00564*.

Wan, S.; Gao, D.; Gu, H.; and Hu, D. 2023. FedPDD: A Privacy-preserving Double Distillation Framework for Cross-silo Federated Recommendation. In *2023 International Joint Conference on Neural Networks (IJCNN)*, 1–8.

Wang, J.; Zhang, L.; Cheng, Y.; Li, S.; Zhang, H.; Huang, D.; and Lan, X. 2023. Tunable Vertical Federated Learning towards Communication-Efficient Model Serving. In *IEEE INFOCOM 2023 - IEEE Conference on Computer Communications*, 860–869.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated Machine Learning: Concept and Applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2): 1–19.

Zhang, H. 2004. The Optimality of Naive Bayes. *Proceedings of the Seventeenth International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*, 1(2): 3.

Zhang, L.; Ding, G.; Wu, Q.; Zou, Y.; Han, Z.; and Wang, J. 2015. Byzantine attack and defense in cognitive radio networks: A survey. *IEEE Communications Surveys & Tutorials*, 17(3): 1342–1363.

Zou, T.; Liu, Y.; and Zhang, Y.-Q. 2023. Mutual Information Regularization for Vertical Federated Learning. *arXiv preprint arXiv:2301.01142*.