

Improving Heterogeneous Model Reuse by Density Estimation

Anke Tang^{1,2}, Yong Luo^{1,2*}, Han Hu³, Fengxiang He⁴,
Kehua Su^{1*}, Bo Du^{1,2}, Yixin Chen⁵, Dacheng Tao⁶

¹ School of Computer Science, National Engineering Research Center for Multimedia Software, Institute of Artificial Intelligence and Hubei Key Laboratory of Multimedia and Network Communication Engineering, Wuhan University, China

² Hubei LuoJia Laboratory, Wuhan, China

³ School of Information and Electronics, Beijing Institute of Technology, China

⁴ JD Explore Academy, JD.com, Inc., China

⁵ Department of CSE, Washington University in St. Louis, USA

⁶ The University of Sydney, Australia

{anketang, luoyong}@whu.edu.cn, hhu@bit.edu.cn, fengxiang.f.he@gmail.com,
{skh, dubo}@whu.edu.cn, chen@cse.wustl.edu, dacheng.tao@gmail.com

Abstract

This paper studies multiparty learning, aiming to learn a model using the private data of different participants. Model reuse is a promising solution for multiparty learning, assuming that a local model has been trained for each party. Considering the potential sample selection bias among different parties, some heterogeneous model reuse approaches have been developed. However, although pre-trained local classifiers are utilized in these approaches, the characteristics of the local data are not well exploited. This motivates us to estimate the density of local data and design an auxiliary model together with the local classifiers for reuse. To address the scenarios where some local models are not well pre-trained, we further design a multiparty cross-entropy loss for calibration. Upon existing works, we address a challenging problem of heterogeneous model reuse from a decision theory perspective and take advantage of recent advances in density estimation. Experimental results on both synthetic and benchmark data demonstrate the superiority of the proposed method.

1 Introduction

In recent years, leveraging centralized large-scale data by deep learning has achieved remarkable success in various application domains. However, there are many scenarios where different participants separately collect data, and data sharing is prohibited due to the privacy legislation and high transmission cost. For example, in some specific applications, such as medicine and autonomous driving, learnable data is inherently privacy-related and decentralized, and each local dataset is often insufficient to train a reliable prediction model [Savage, 2017; Rajkomar *et al.*, 2019]. Therefore, multiparty

learning is proposed to learn a reliable model using separated private datasets without sharing trainable samples [Pathak *et al.*, 2010].

Most of the existing multiparty learning systems focus on training a shared global model to simultaneously achieve satisfactory accuracy and protect data privacy. These systems usually assume that each party trains a homogeneous local model, e.g., training neural networks with the same architecture [Shokri and Shmatikov, 2015]. This makes it possible to directly average model parameters or aggregate gradient information [Warnat-Herresthal *et al.*, 2021; McMahan *et al.*, 2016; Li *et al.*, 2019]. Some other works assume that each party has already trained a local model on its local dataset, and then apply model reuse to learn a global model [Pathak *et al.*, 2010; Yang *et al.*, 2017]. A typical example is the heterogeneous model reuse (HMR) method presented in [Wu *et al.*, 2019]. Since only the output predictions of local models are utilized to derive a global model, the data can be non-i.i.d distributed and the architectures of different local models can vary among different parties. In addition, training of the global model can be quite efficient and data transmission cost can be significantly reduced.

There also exist some other model reuse approaches that may be utilized for multiparty learning. For example, pre-trained nonlinear auxiliary classifiers are adapted to new object functions in [Li *et al.*, 2012]. Alternatively, the simple voting strategy can be adapted and improved to ensemble local models [Zhou, 2012; Wu *et al.*, 2019]. In addition to the local models, a few works consider the design of specification to assist model selection and weight assignment [Ding *et al.*, 2020; Wu *et al.*, 2023]. However, some important characteristics of the local data, such as the data density information are simply ignored in these approaches.

This motivates us to propose a novel heterogeneous model reuse method from a decision theory perspective that exploits the density information of local data. In particular, in addition to the local model provided by each party, we estimate

*Corresponding authors: Yong Luo, Kehua Su.

the probability density function of local data and design an auxiliary generative probabilistic model for reuse.

The proposed model ensemble strategy is based on the rules of Bayesian inference. By feeding the target samples into the density estimators, we can obtain confidence scores of the accompanying local classifier when performing prediction for these samples. Focusing on the semantic outputs, the heterogeneous local models are treated as black boxes and are allowed to abstain from making a final decision if the confidence is low for a certain sample in the prediction. Therefore, aided by the density estimation, we can assign sample-level weight to the prediction of the local classifier. Besides, when some local models are insufficiently trained on local datasets, we design a multiparty cross-entropy loss for calibration. The designed loss automatically assigns a larger gradient to the local model that provides a more significant density estimation, and thus, enables it to obtain faster parameter updates.

To summarize, the main contributions of this paper are:

- we propose a novel model reuse approach for multiparty learning, where the data density is explored to help the reuse of biased models trained on local datasets to construct a reliable global model;
- we design a multiparty cross-entropy loss, which can further optimize deep global model in an end-to-end manner.

We conduct experiments on both synthetic and benchmark data for image classification tasks. The experimental results demonstrate that our method is superior to some competitive and recently proposed counterparts [Wu *et al.*, 2019; Wu *et al.*, 2023]. Specifically, we achieve a significant 17.4% improvement compared with [Wu *et al.*, 2023] in the case of three strict disjoint parties on the benchmark data. Besides, the proposed calibration operation is proven to be effective even when local models are random initialized without training.

2 Related Work

In this section, we briefly summarize related works on multiparty learning and model reuse.

2.1 Multiparty Learning

Secure multiparty computation (SMC) [Yao, 1986; Lindell, 2005] naturally involves multiple parties. The goal of SMC is to design a protocol, which is typically complicated, to exchange messages without revealing private data and compute a function on multiparty data. SMC requires communication between parties, leading to a huge amount of communication overhead. The complicated computation protocols are another practical challenge and may not be achieved efficiently. Despite these shortcomings, the capabilities of SMC still have a great potential for machine learning applications, enabling training and evaluation on the underlying full dataset. There are several studies on machine learning via SMC [Juvekar *et al.*, 2018; Mohassel and Rindal, 2018; Kumar *et al.*, 2020]. In some cases, partial knowledge disclosure may be considered acceptable and traded for efficiency.

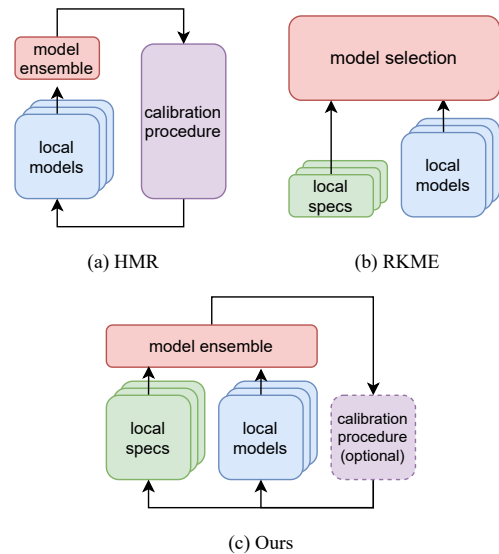


Figure 1: A comparison of our heterogeneous model reuse method with HMR [Wu *et al.*, 2019] and RKME [Wu *et al.*, 2023]. In HMR, multiple local models are simply combined and carefully calibrated to construct a global model. RKME does not require calibration, but some specifications that summarize local datasets are utilized for complicated model selection in the deployment phase. We utilize different types of specifications of local datasets in a different way, and design cheap aggregation strategy for model ensemble, where the calibration is optional due to satisfactory zero-shot test accuracy.

For example, a SMC framework [Knott *et al.*, 2021] is proposed to perform an efficient private evaluation of modern machine-learning models under a semi-honest threat model.

Differential privacy [Dwork, 2008] and k-Anonymity [Sweeney, 2002] are used in another line of work for multiparty learning. These methods try to add noise to the data or obscure certain sensitive attributes until the third party cannot distinguish the individual. The disadvantage is that there is still heavy data transmission, which does not apply to large-scale training. In addition to transmitting encrypted data, there are studies on the encrypted transmission of parameters and training gradients, such as federated learning [Yang *et al.*, 2019] and swarm learning [Warnat-Herresthal *et al.*, 2021]. Federated learning was first proposed by Google and has been developed rapidly since then, wherein dedicated parameter servers are responsible for aggregating and distributing local training gradients. Besides, swarm learning is a data privacy-preserving framework that utilizes blockchain technology to decentralize machine learning-based systems. However, these methods usually can only deal with homogeneous local models [Pathak *et al.*, 2010; Rajkumar and Agarwal, 2012].

2.2 Model Reuse

Model reuse aims to learn a reliable model for target task by reusing some related pre-trained models, often without accessing their original data [Zhou, 2016]. Heterogeneous model reuse (HMR) for multiparty learning [Wu *et al.*, 2019] is the closest work to ours. Based on the idea of learn-

ware [Zhou, 2016], the black-box construction of a global model from the heterogeneous local models is performed. In HMR [Wu *et al.*, 2019], a global model is built based on the idea of Max-Model Predictor and then the model is carefully calibrated using a designed multiparty multiclass margin (MPMC-margin) loss. However, the accuracy is usually unsatisfactory under the zero-shot setting (model ensemble without any parameter or architecture calibration) due to the lack of exploitation of prior information. In RKME [Wu *et al.*, 2023], each local classifier is assumed to be associated with a reduced kernel mean embedding as model specification, which largely improves the zero-shot test accuracy, but complicated model selection is required. Our method makes use of the data density specification (with privacy guarantee presented in section 3.4), and a cheap model ensemble strategy is adopted to achieve very promising performance, even without calibration. Figure 1 is a comparison of our method with HMR and RKME.

The major difference between model reuse and some other related paradigms such as federated learning is that for the latter, information is exchanged among different parties in privacy-preserving ways during the training phase. While for model reuse, the training process of different parties is independent, and information is only exchanged and exploited in the form of models during the deployment phase [Yang *et al.*, 2019; McMahan *et al.*, 2017; Ding *et al.*, 2020].

3 The Proposed Method

In this section, we first introduce the main notations and preliminaries of heterogeneous model reuse for the multiparty learning problem.

3.1 Notations and Preliminaries

We consider that there are N participants in a multiparty learning system, and each participant $i \in [N]$ is known as a party and has its own local dataset $S_i = \{(x, y) \in \mathcal{X} \times \mathcal{Y}_i\}$ containing data samples and corresponding labels, where the labels are in $\mathcal{Y}_i \subseteq \mathcal{Y}$. Here, data exist in the form of isolated islands. Each party can only access its local dataset, so the underlying global dataset $S = \cup_{i=1}^N S_i$ cannot be directly observed by any parties. The participants attempt to cooperate in bridging the gap between model accuracy and training data accessibility, and obtaining a reliable global model. The whole model reuse progress is diagrammed as figure 1(d).

For a multiparty classification problem, each party i holds its local classifier $\mathbf{F}_i : \mathcal{X} \rightarrow \mathcal{Y}_i$ which is trained on its local dataset S_i and the types of classifiers can vary among parties. The first challenge of learning the global model arises from the potential sample selection bias or covariate shift. A local classifier may misclassify an unseen input sample into the wrong class. In fact, a local classifier would never be able to predict correctly if the local label space is not equal to the full label space, i.e. when \mathbf{F}_i can only estimate posterior class probabilities $p(C_k|x, S_i)$ given x for class $C_k \in \mathcal{Y}_i \subsetneq \mathcal{Y}$, we simply assign zero to $p(C_k|x, S_i)$ for $C_k \in \mathcal{Y} \setminus \mathcal{Y}_i$.

As for our method, in addition to the local classifier, each party should also fit a local density estimator $\mathbf{G}_i : \mathcal{X} \rightarrow \mathbb{R}$ on S_i in an unsupervised manner. The density estimator \mathbf{G}_i

is a generative probability model that learns to approximate the log-likelihood probability of the observations. As we shall see in the following section, the log-likelihood and the dataset prior constitute the transition matrix that transforms the local class posterior probability to the global class posterior probability. Therefore, the density estimators participate in the ensemble of local models in our model reuse framework together with the classifiers. Besides, since \mathbf{G}_i only provides the function of estimating the log-likelihood for given samples and does not need to generate samples, the privacy of the local dataset is guaranteed.

3.2 Heterogeneous Model Reuse Aided by Density Estimation

We tackle the multiparty learning problem by utilizing some pre-trained local models to train a reliable global one. Before combining local models, we shall dive into the decision theory of the multiparty learning problem to gain some insight. We denote the joint probability distribution on the underlying global dataset as $p(x, C_k)$, and local joint distribution as conditional probability $p(x, C_k|S_i)$ given local dataset S_i . The underlying global dataset is inaccessible and hence a directly estimation of $p(x, C_k)$ is intractable. A possible solution is to marginalize out S_i to obtain $p(x, C_k)$:

$$p(x, C_k) = \mathbb{E}_{S_i \sim S} [p(x, C_k|S_i)]. \quad (1)$$

For a classification task, we need to assign each observation x to a certain class C_k . Such operation will divide the input space \mathcal{X} into adjacent decision regions $\{\mathcal{R}_k\}$. Our ultimate goal is to find the optimal decision policy $f^* \in \mathcal{X} \mapsto \mathcal{Y}$ that maximizes the probability of correct classification, i.e.,

$$P(\text{correct}) = \sum_k \int_{\mathcal{R}_k} p(x, f^*(x)) dx. \quad (2)$$

It is straightforward to see that we can maximize the probability of correct classification if and only if we assign each x to the class with the most considerable joint probability, since we can only assign x to one class at a time. Since the upper bound of Eq. (2) is $\int \max_{C_k} p(x, C_k) dx$ [Bishop and Nasrabadi, 2006], we have $f^* = \text{argmax}_{C_k} p(\cdot, C_k)$. By further expanding out $p(x, C_k)$ using marginalization Eq. (1), we can reformulate Eq. (2) as

$$P_{\max} = \int \sum_i p(C_k^*|x, S_i) p(x|S_i) p(S_i) dx, \quad (3)$$

where $C_k^* = \text{argmax}_{C_k} p(x, C_k)$. In this way, we construct the global joint distribution by exploiting information about prior dataset distribution $p(S_i)$, local density/likelihood $p(x|S_i)$ and local class posterior $p(C_k|x, S_i)$. To gain further insight into the global joint function, we multiply and divide the global likelihood $p(x)$ inner the right-hand integral, and rewrite Eq. (3) equivalently as

$$\int p(x) \underbrace{\left(\sum_i p(C_k^*|x, S_i) \frac{p(x|S_i)p(S_i)}{p(x)} \right)}_{p(C_k|x)} dx \quad (4)$$

$$= \int p(x) \left(\sum_i p(C_k^*|x, S_i) \lambda_i \right) dx, \quad (5)$$

where $\sum_{i=1}^N \lambda_i = 1$ and $\lambda_i = p(S_i|x)$ according to Bayes' theorem. Compared with the original joint function Eq. (2), we now represent the global posterior probability $p(C_k|x)$ as a weighted sum of local posteriors. Evidently, when there is only one party, $\lambda_1 = 1$, this joint representation degenerates to the familiar standard classification case, i.e., assigning x to class $\text{argmax}_{C_k} p(C_k|x)$.

When the dimension of input space \mathcal{X} is small, estimation of the density $p(x|S_i)$ is trivial, and some popular and vanilla density estimation techniques, such as Gaussian mixture and kernel density estimators, from the classical unsupervised learning community can be directly adopted. However, when the input dimension is high, such as in the case of image classification, landscape of the log-likelihood function $\mathbf{G}_i(x)$ for approximating the density can be extremely sharp due to the sparse sample distribution and thus intractable in practice. We address this issue by extracting a sizeable common factor inner the summation, and rewriting the integrand in Eq. (3) equivalently as

$$p(x|S_j) \sum_i p(C_k^*|x, S_i) p(S_i) e^{\log p(x|S_i) - \log p(x|S_j)}, \quad (6)$$

where $j = \text{argmax}_j \log p(x|S_j)$. In this way, we normalize the likelihood estimation to a reasonable interval $[0, 1]$ without loss of information.

We then ensemble the local models according to Eq. (6), as illustrated in Figure 1(c), where $p(S_i)$ is proportional to the size of local dataset and sum up to 1, so that $p(S_i) = |S_i| / \sum_j |S_j|$. Moreover, the class posterior $p(C_k|x, S_i)$ and density $p(x|S_i)$ can be approximated by the discriminate model \mathbf{F}_i and generative model \mathbf{G}_i , respectively. Finally, the global model can make a final decision, dropping the common factor $p(x|S_j)$, and the final decision policy can be written in a compact form by matrices as:

$$\text{argmax}_C \left\langle \mathbf{F}^{(C)}(\cdot), \|\mathbf{S}\|_1 \odot \exp(\mathbf{G}(\cdot) - \bar{\mathbf{G}}(\cdot)) \right\rangle, \quad (7)$$

where $\bar{\mathbf{G}} = \text{argmax}_i \mathbf{G}_i(\cdot)$ and \odot is the Hadamard product. Hereafter, we denote this inner product as the decision objective function $J(C)$ for simplicity. The main procedure of our model reuse algorithm is summarized in Algorithm 1.

The following claim shows that our global model can be considered as a more general version of the max-model predictor defined in [Wu *et al.*, 2019].

Claim 1. Let $\lambda_i = \delta_{\text{argmax}_i p(x|S_i)}^i$, Eq.(5) would degenerate to a max-model predictor.

Proof. If we assign λ_i to $\delta_{\text{argmax}_i p(x|S_i)}^i$, then according to Eq. (5), we have

$$P = \int p(x) p(C_k^*|x, \text{argmax}_{S_i} p(x|S_i)) dx. \quad (8)$$

Recall the definition of C_k^* and by dropping the common factor $p(x)$, the decision policy Eq. (8) can be characterized as $\text{argmax}_{C_k} \max_{S_i} p(C_k|x, S_i)$. \square

By sharing selected samples among parties, HMR [Wu *et al.*, 2019] makes $p(\cdot|S_i)$ get closer to each other in the Hilbert space so that $p(C_k|x, \text{argmax}_{S_i} p(x|S_i)) \rightarrow p(C_k|x)$.

Algorithm 1 Heterogeneous Model Reuse aided by Density Estimation (without Calibration).

Input:

- Local classifiers $\mathbf{F}_1, \mathbf{F}_2, \dots, \mathbf{F}_N$
 \triangleright e.g. CART, SVM, MLP, CNN
- Local log-likelihood estimators $\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_N$
 \triangleright e.g. Kernel Density, Real-NVP, VAE
- Sizes of local datasets $|S_1|, |S_2|, \dots, |S_N|$
- Query samples x_1, x_2, \dots, x_m

Output: Labels of classification

- 1: **for** $j = 1, 2, \dots, N$ **do**
 - 2: initialize dataset prior probability by normalization:
 $p_j := |S_j| / \sum_i |S_i|$
 - 3: **for** $i = 1, 2, \dots, m$ **do**
 - 4: for each class k calculate local posterior probability, fill zeros for missing entries:
 $\mathcal{G}_{ijk} := \mathbf{F}_j^{(C_k)}(x_i)$ or 0
 - 5: calculate local log-likelihood:
 $\mathcal{F}_{ij} := \mathbf{G}_j(x_i)$
 - 6: **end for**
 - 7: **end for**
 - 8: **for** $i = 1, 2, \dots, m$ **do**
 - 9: calculate objective function for each class k :
 $J_{ik} := \sum_j p_j \mathcal{F}_{ijk} \exp(\mathcal{G}_{ijk} - \max_j \mathcal{G}_{ijk})$
 - 10: make decision for query sample x_i :
 $C_i := \text{argmax}_k J_{ik}$
 - 11: **end for**
 - 12: **return** $\mathbf{C} = C_1, C_2, \dots, C_m$
-

3.3 Multiparty Cross-Entropy Loss

In this subsection, we design a novel *multiparty cross-entropy loss* (MPCE loss), which enables us to calibrate the classifiers in the compositional deep global model in an end-to-end manner. We use θ and μ to denote the sets of classifiers' and generative models' parameters respectively, and we aim to find optimal θ so that we approximate the actual class posterior function well. A popular way to measure the distance between two probabilities is to compute the Kullback-Leibler (KL) divergence between them. With a slight abuse of notation, we characterize the KL divergence between true class posterior and approximated class posterior as

$$\text{KL}(p||p_\theta) = \sum_{C \in \mathcal{C}} p(C|x) \log \frac{p(C|x)}{p_\theta(C|x)} \quad (9)$$

$$= \mathbb{E}_{C \sim p} [\log p(C|x)] - \mathbb{E}_{C \sim p} [\log p_\theta(C|x)]. \quad (10)$$

The first term in Eq. (10) is fixed and associated with the dataset. Besides, as for a classification task, $p(\cdot|x)$ is a Kronecker delta function of class C , so this expectation term is 0. We define the *MPCE loss* as the second term in Eq. (10), that is

$$\mathcal{L}_{\text{mpce}}(\hat{y}, y) = -\mathbb{E}_{C \sim p} [\log p_\theta(C|x)] \quad (11)$$

$$= -\sum_k \delta_k^y \log p_\theta(C_k|x), \quad (12)$$

where δ is the Kronecker delta function, δ_k^y is 1 if k and y are equal, and 0 otherwise. By utilizing the global posterior presented in Eq. (4), we can further expand out the loss to get

$$\mathcal{L}_{\text{mpce}}(\hat{y}, y) = -\log \left\{ \sum_i p_\theta(C_y|x, S_i) \frac{p_\mu(x|S_i)p(S_i)}{p(x)} \right\}. \quad (13)$$

Claim 2. For single party case, the MPCE loss degenerates to the standard cross-entropy loss.

Proof. Evidently, when there is only one party, we have $p(x|S_i)p(S_i) = p(x, S_i) = p(x)$. \square

Next, We follow the same argument about the high dimensional situation and apply the normalization trick presented in Eq. (6), to obtain

$$\begin{aligned} \mathcal{L}_{\text{mpce}}(\hat{y}, y) = & \left\{ -\bar{\mathbf{G}}_\mu(x) + \log p(x) \right\} \\ & - \log \left\langle \mathbf{F}_\theta^{(C_y)}(x), \|\mathbf{S}\|_1 \odot \exp(\mathbf{G}_\mu(x) - \bar{\mathbf{G}}_\mu(x)) \right\rangle. \end{aligned} \quad (14)$$

Notice that in Eq. (14), the last term in the log operation is the same as that in Eq. (7), minimizing the negative-log MPCE loss will maximize the policy objective function. At step t , we can update the model parameters $[\theta, \mu]^\top$ using $-\eta g_t$, where η is the learning rate hyperparameter and g_t is the gradient defined as

$$g_t = \left[\nabla_\theta \mathcal{L}_{\text{mpce}}(\hat{y}, y) \quad \sum_{\forall_i y \in \mathcal{Y}_i} \nabla_\mu \mathcal{L}_{\text{gen}}^{(i)}(x) \right]^\top. \quad (15)$$

Here, $\mathcal{L}_{\text{gen}}^{(i)}$ is some unsupervised loss (such as negative log-likelihood for normalizing flows) of the i -th density estimator. This can be conducted in a differential privacy manner by clipping and adding noise to g_t

$$\bar{g}_t = g_t / \max\{1, \|g_t\|_2/C\} + \mathcal{N}(0, \sigma^2 C^2 I). \quad (16)$$

By optimizing the MPCE loss, the gradient information is back-propagated along a path weighted by the density estimation. The party that provides more significant density estimation for calibration would obtain larger gradients and faster parameter updates.

3.4 Privacy Guarantee

In this paper, the data density is utilized as model specification, and this may lead to privacy issue. However, since we can conduct density estimation in a differential privacy manner, the privacy can be well protected.

In particular, differential privacy [Dwork *et al.*, 2006; Dwork, 2011; Dwork *et al.*, 2014], which is defined as follows, has become a standard for privacy analysis.

Definition 1 ([Dwork *et al.*, 2006]). A randomized algorithm $\mathcal{M} : \mathcal{D} \mapsto \mathcal{R}$ satisfies (ϵ, δ) -differential privacy (DP) if and only if for any two adjacent input datasets D, D' that differ in a single entry and for any subset of outputs $S \subset \mathcal{R}$ it holds that

$$\Pr[\mathcal{M}(D) \in S] \leq e^\epsilon \Pr[\mathcal{M}(D') \in S] + \delta. \quad (17)$$

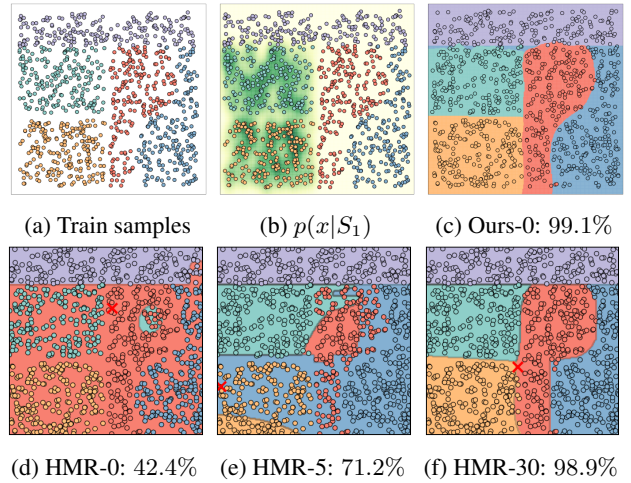


Figure 2: Visualization results of 2D toy example. (a) The five-class 2D dataset. (b) The estimated density by party 1. (c) Decision boundary and accuracy of our method without calibration (iteration 0) on the testing data. (d-f) Decision boundary and test accuracy of HMR at iteration 0, 5 and 30.

It has been demonstrated that density estimators can be trained in a differential privacy manner to approximate arbitrary, high-dimensional distributions based on the DP-SGD algorithm [Abadi *et al.*, 2016; Waites and Cummings, 2021]. Therefore, the proposed model aggregation strategy is guaranteed to be (ϵ, δ) -differentially private when local models are pre-trained in a differential privacy manner, where ϵ and δ are the training privacy budget and training privacy tolerance hyperparameters, respectively.

4 Experiments

In this section, we evaluate our proposed method using a two-dimensional toy dataset and a popular benchmark dataset. The basic experimental setup for our 2D toy and benchmark experiments is similar to that adopted in [Wu *et al.*, 2019]. Experiments on the benchmark data demonstrate our model reuse algorithm and end-to-end calibration process on various biased data distribution scenarios. The code is available at <https://github.com/tanganke/HMR>.

4.1 Toy Experiment

We first visualize our proposed method with a toy example. Here, we create a 2D toy dataset with 2000 points, each associated with a label from 5 classes denoted by different colors. The dataset is equally split into a training set and a test set, as shown in Figure 2a.

There are three parties in this toy example, each equipped with different local models. The three parties use logistic regression, Gaussian kernel SVM, and gradient boosting decision tree for classification, respectively. In addition, they all use kernel density estimator with bandwidth set to 0.1 to estimate the log-likelihood function. We implement the local models using the scikit-learn package [Pedregosa *et al.*, 2011]. Each party can only access a subset of the complete training set as the local dataset. The accessible samples are

Setting	A	B	C	D	Average
RKME	87.3 ± 1.4	80.4 ± 4.2	68.7 ± 3.2	70.6 ± 3.8	76.7
HMR-1	87.3 ± 0.4	54.2 ± 3.0	37.6 ± 2.3	56.0 ± 1.3	58.8
HMR-10	88.3 ± 0.6	66.0 ± 3.1	66.5 ± 4.3	59.4 ± 4.0	70.1
HMR-50	91.2 ± 0.3	85.1 ± 1.9	82.1 ± 0.9	77.6 ± 1.4	84.0
HMR-100	91.6 ± 0.1	87.4 ± 1.1	84.4 ± 0.6	82.1 ± 0.8	86.4
Ours	91.3 ± 0.2	88.4 ± 0.4	84.6 ± 0.3	84.7 ± 0.4	87.2

Table 1: Accuracy on benchmark data under four multiparty settings (avg. \pm std.%). Here HMR-X represents the HMR method that has been calibrated for X rounds.

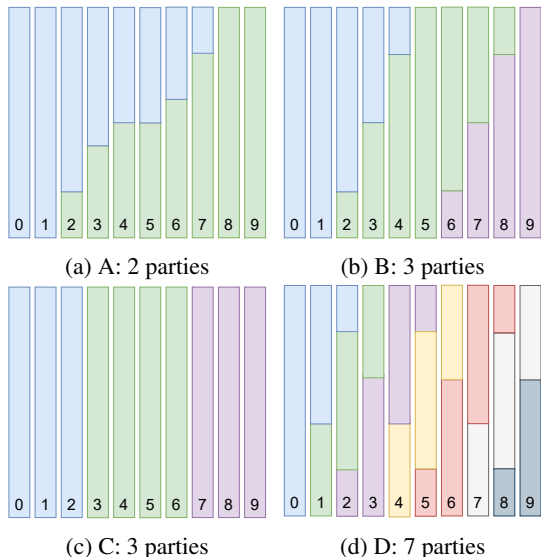


Figure 3: Four experiment settings with different sample selection biases by dividing the training set of Fashion-MNIST. Each color represents a local dataset associated with a party.

all the green and orange ones for party 1, all the red and blue ones for party 2, and all the blue and purple ones for party 3.

We first train the classifiers in a supervised manner and the kernel density estimators in an unsupervised manner on the corresponding local dataset. Then we reuse these trained local models according to Algorithm 1 to make final decisions on test samples. Lastly, we analyze the zero-shot composition performance (without calibration) and compare with the most related work HMR [Wu *et al.*, 2019]. The results are shown by Figure 2. From the results, we can see that the zero-shot composition accuracy reaches 99.1%, and the decision boundary is shown in figure 2c. In contrast, the zero-shot accuracy of HMR is only 42.4% and the performance is comparable to our method after 30 rounds of calibrations.

4.2 Benchmark Experiment

In this set of experiments, we aim to understand how well our method compares to the state-of-the-art heterogeneous model reuse approaches for multiparty learning and the strength and weakness of our calibration procedure. Specifically, we mainly compare our method with HMR [Wu *et al.*, 2019] and RKME [Wu *et al.*, 2023].

- HMR uses a max-model predictor as the global model

together with a designed multiparty multiclass margin loss function for further calibration.

- RKME trains local classifiers and computes the *reduced kernel mean embedding* (RKME) specification in the upload phase, assigns weights to local classifiers based on RKME specification, and trains a model selector for future tasks in the deployment phase.

In addition to multiparty learning, we train a centralized model on the entire training set for comparison.

We evaluate our method, HMR, and RKME on Fashion-MNIST [Xiao *et al.*, 2017], a popular benchmark dataset in the machine learning community, containing 70,000 28×28 gray-scale fashion product images, each associated with a label from 10 classes. The complete training set is split into a training set of 60,000 examples and a test set of 10,000 examples. To simulate the multiparty setting, we separate the training set into different parties with biased sample distribution. The resulting four cases are shown as figure 3, and we refer to [Wu *et al.*, 2019] for a detailed description.

We set the training batch size to be 128, and the learning rate of all local models to $1e-4$ during the local training. The learning rate is $1e-5$ during the calibration step. All local classifiers have the same 3-layer convolutional network and all local density estimators are the same 12-layer real non-volume preserving (real NVP) flow network [Dinh *et al.*, 2016]. The real NVP network is a class of invertible functions and both the forward and its inverse computations are quite efficient. This enables exact and tractable density evaluation. As For RKME, we set the reduced dimension size to 10, and the number of generated samples to 200.

Firstly, we test the zero-shot composition accuracy of the compared approaches, and if possible, evaluate the subsequent calibration performance. Due to the difference in the calibration mechanism, for HMR, a new neuron is added at the last layer of the classifiers to add reserved class output. In contrast, for our method, the calibration is end-to-end, and the structure of the classifiers is fixed. Therefore our method is more simple to implement. HMR retrains each local model on the augmented data set for one epoch during calibration. As for our method, the calibration operation is performed directly on the global model. Only a batch of 64 data samples is randomly selected from the training set to perform gradient back-propagation. We run 20 times for each setting to mitigate randomness and display the standard deviation bands. Experimental results including the centralized ones are visualized in Figure 4, and reported in Table 1.

From Figure 4 and Table 1, we can see that for sufficient

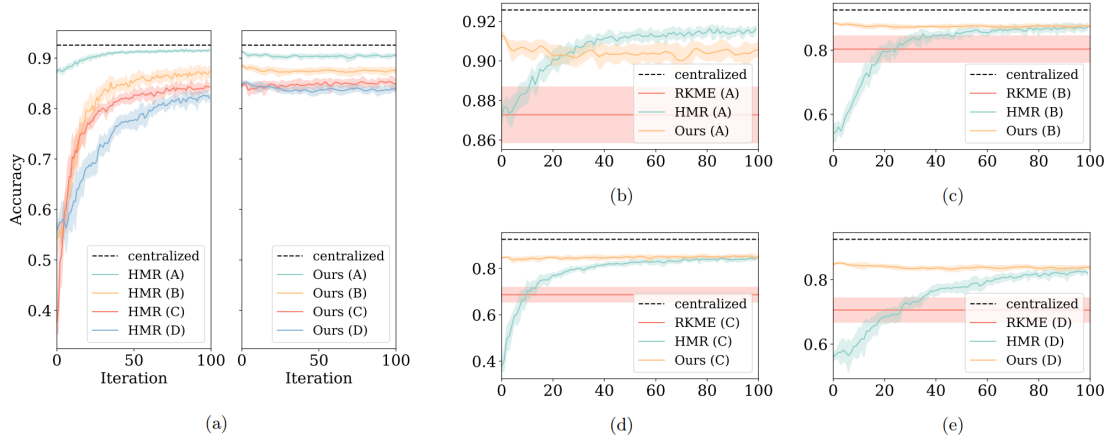


Figure 4: The test accuracy curves over calibration iterations (avg. \pm std.). (a) overall performance of HMR (left) and our method (right) under the four multiparty settings. (b-e) performance of different compared approaches under each multiparty setting, where RKME is a constant value since it is inherently a method that cannot be subsequently calibrated.

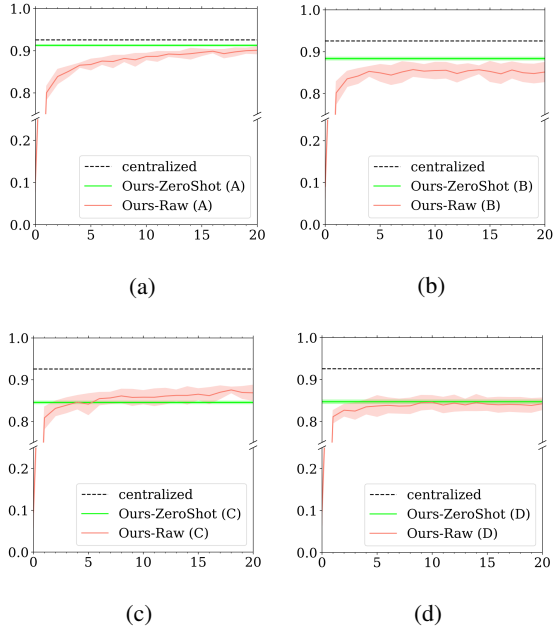


Figure 5: A comparison of the Raw models and the ZeroShot models (avg. \pm std.). Here, the Raw models represent the global models directly calibrated from random initialization without training, and the ZeroShot models represent the zero-shot composition from sufficient trained local models.

trained local models, our model reuse method achieves relatively superior accuracy from the beginning and outperforms all other model reuse counterparts. At the same time, subsequent calibrations do not improve performance or, even worse, slightly degrade performance. This may be because that the local models are well trained, the further calibration may lead to slight over-fitting. This demonstrates the effectiveness of our method that exploring data density for reuse.

Then we demonstrate that our calibration procedure is indeed effective when the local models are not well trained. In particular, we compare the test accuracy of the above zero-shot composition with the global model directly calibrated from random initialization without training (denoted as Raw). We fit the raw global models on the full training set for 20 epochs with the learning rate set to be $1e-4$ and runs 20 times for each multiparty setting. The results are shown as figure 5. We can observe from the results that during our calibration, the Raw model consistently achieves higher performance and eventually converges to the zero-shot composition accuracy.

5 Conclusions

In this paper, we propose a novel heterogeneous model reuse method for multiparty learning, where an auxiliary density estimator is designed to help the reuse. In practical deployment, the pre-trained locals model can be provided as web query services, which is secure and privacy-friendly. Besides, we propose a multiparty cross-entropy criteria to measure the distance between the true global posterior and the approximation. Experimental results on both synthetic and benchmark data demonstrate the superiority of our method. From the results, we mainly conclude that: 1) exploring more prior knowledge on the private local data during the training phase can lead to higher performance during the deployment phase; 2) substantial performance boost can be obtained by using the designed simple and easy-to-implement calibration strategy. To the best of our knowledge, this is the first work to directly consider the multiparty learning problem from a decision theory perspective.

In the future, we plan to investigate the feature space to characterize and manipulate the knowledge learned from specific data. In addition to the popular image classification task, the proposed method can also be applied to tasks in other fields such as machine translation and speech recognition.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2021YFC3300200, the Special Fund of Hubei LuoJia Laboratory under Grant 220100014, the National Natural Science Foundation of China (Grant No. 62276195 and 62272354). Prof Dacheng Tao is partially supported by Australian Research Council Project FL-170100117.

References

- [Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.
- [Bishop and Nasrabadi, 2006] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.
- [Ding *et al.*, 2020] Yao-Xiang Ding, Zhi-Hua Zhou, Sinno Jialin Pan, and Masashi Sugiyama. Boosting-Based Reliable Model Reuse. *Proceedings of Machine Learning Research*, 129(1):145–160, 2020.
- [Dinh *et al.*, 2016] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–32, may 2016.
- [Dwork *et al.*, 2006] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer, 2006.
- [Dwork *et al.*, 2014] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.
- [Dwork, 2008] Cynthia Dwork. Differential privacy: A survey of results. In *International conference on theory and applications of models of computation*, pages 1–19. Springer, 2008.
- [Dwork, 2011] Cynthia Dwork. A firm foundation for private data analysis. *Communications of the ACM*, 54(1):86–95, 2011.
- [Juvekar *et al.*, 2018] Chiraag Juvekar, Vinod Vaikuntanathan, and Anantha Chandrakasan. {GAZELLE}: A low latency framework for secure neural network inference. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1651–1669, 2018.
- [Knott *et al.*, 2021] Brian Knott, Shobha Venkataraman, Awni Hannun, Shubho Sengupta, Mark Ibrahim, and Laurens van der Maaten. Crypten: Secure multi-party computation meets machine learning. *Advances in Neural Information Processing Systems*, 34:1–13, 2021.
- [Kumar *et al.*, 2020] Nishant Kumar, Mayank Rathee, Nishanth Chandran, Divya Gupta, Aseem Rastogi, and Rahul Sharma. Cryptflow: Secure tensorflow inference. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 336–353. IEEE, 2020.
- [Li *et al.*, 2012] Nan Li, Ivor W Tsang, and Zhi-Hua Zhou. Efficient optimization of performance measures by classifier adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1370–1382, 2012.
- [Li *et al.*, 2019] Liping Li, Wei Xu, Tianyi Chen, Georgios B Giannakis, and Qing Ling. Rsa: Byzantine-robust stochastic aggregation methods for distributed learning from heterogeneous datasets. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1544–1551, 2019.
- [Lindell, 2005] Yehuda Lindell. Secure multiparty computation for privacy preserving data mining. In *Encyclopedia of Data Warehousing and Mining*, pages 1005–1009. IGI global, 2005.
- [McMahan *et al.*, 2016] H Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, pages 1–11, 2016.
- [McMahan *et al.*, 2017] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.
- [Mohassel and Rindal, 2018] Payman Mohassel and Peter Rindal. Aby3: A mixed protocol framework for machine learning. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 35–52, 2018.
- [Pathak *et al.*, 2010] Manas Pathak, Shantanu Rane, and Bhiksha Raj. Multiparty differential privacy via aggregation of locally trained classifiers. In *Advances in Neural Information Processing Systems*, volume 23, page 1876–1884. Curran Associates, Inc., 2010.
- [Pedregosa *et al.*, 2011] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Andreas Müller, Joel Nothman, Gilles Louppe, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [Rajkumar *et al.*, 2019] Alvin Rajkumar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019.
- [Rajkumar and Agarwal, 2012] Arun Rajkumar and Shivani Agarwal. A differentially private stochastic gradient descent algorithm for multiparty classification. In *Artificial Intelligence and Statistics*, pages 933–941. PMLR, 2012.
- [Savage, 2017] Neil Savage. Calculating disease. *Nature*, 550(7676):S115–S117, 2017.

- [Shokri and Shmatikov, 2015] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.
- [Sweeney, 2002] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(05):557–570, 2002.
- [Waites and Cummings, 2021] Chris Waites and Rachel Cummings. Differentially private normalizing flows for privacy-preserving density estimation. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 1000–1009, 2021.
- [Warnat-Herresthal *et al.*, 2021] Stefanie Warnat-Herresthal, Hartmut Schultze, Krishnaprasad Lingadahalli Shastry, Sathyanarayanan Manamohan, Saikat Mukherjee, Vishesh Garg, Ravi Sarveswara, Kristian Händler, Peter Pickkers, N Ahmad Aziz, et al. Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594(7862):265–270, 2021.
- [Wu *et al.*, 2019] Xi Zhu Wu, Song Liu, and Zhi Hua Zhou. Heterogeneous model reuse via optimizing multiparty multiclass margin. *36th International Conference on Machine Learning, ICML 2019*, 2019-June:11862–11871, 2019.
- [Wu *et al.*, 2023] Xi-Zhu Wu, Wenkai Xu, Song Liu, and Zhi-Hua Zhou. Model reuse with reduced kernel mean embedding specification. *IEEE Transactions on Knowledge and Data Engineering*, 35(01):699–710, jan 2023.
- [Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. (arXiv:1708.07747):1–6, September 2017.
- [Yang *et al.*, 2017] Yang Yang, De-Chuan Zhan, Ying Fan, Yuan Jiang, and Zhi-Hua Zhou. Deep Learning for Fixed Model Reuse. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1):2831–2837, February 2017.
- [Yang *et al.*, 2019] Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.
- [Yao, 1986] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *27th Annual Symposium on Foundations of Computer Science (sfcs 1986)*, pages 162–167. IEEE, 1986.
- [Zhou, 2012] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.
- [Zhou, 2016] Zhi Hua Zhou. Learnware: on the future of machine learning. *Frontiers of Computer Science*, 10(4):589–590, 2016.