Supplementary Information for "Selective Knowledge Sharing for Privacy-Preserving Federated Distillation"

Jiawei Shao[1], **Fangzhao Wu**[2,*], **and Jun Zhang**[1,*]

[1]Hong Kong University of Science and Technology, Hong Kong
[2]Microsoft Research Asia, Beijing
[*]correspondce: Jun Zhang (eejzhang@ust.hk), Fangzhao Wu (wufangzhao@gmail.com)

Supplementary Material

Proof of Theorem 2

Prior to proving Theorem 2, we first present two Lemmas.

Lemma 1. *Given hypothesis spaces* $\mathscr{H} := \{\hat{\boldsymbol{h}} : \mathscr{X} \to V(\Delta^{C-1})\}$ *and* $\mathscr{G} := \{g : \mathscr{X} \to \{0,1\}\}$ *with* $g(\boldsymbol{x}) = \frac{1}{2}\|\hat{\boldsymbol{h}}(\boldsymbol{x}) - \hat{\boldsymbol{h}}'(\boldsymbol{x})\|_1$ *for* $\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}' \in \mathscr{H}$*, we have* $|\mathscr{L}_{\mathscr{D}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}') - \mathscr{L}_{\mathscr{D}'}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}')| \leq d_{\mathscr{G}}(\mathscr{D}, \mathscr{D}')$ *for* $\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}' \in \mathscr{H}$.

Proof.

$$
\begin{aligned}
d_{\mathscr{G}}(\mathscr{D}, \mathscr{D}') &= 2 \sup_{g \in \mathscr{G}} \left| \Pr_{\mathscr{D}}[g(\boldsymbol{x}) = 1] - \Pr_{\mathscr{D}'}[g(\boldsymbol{x}) = 1] \right|, \\
&= \sup_{\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}' \in \mathscr{H}} 2 \left| \frac{1}{2}\mathbb{E}_{\boldsymbol{x} \sim \mathscr{D}}[\hat{\boldsymbol{h}}(\boldsymbol{x}) - \hat{\boldsymbol{h}}'(\boldsymbol{x})] - \frac{1}{2}\mathbb{E}_{\boldsymbol{x} \sim \mathscr{D}'}[\hat{\boldsymbol{h}}(\boldsymbol{x}) - \hat{\boldsymbol{h}}'(\boldsymbol{x})] \right| \geq |\mathscr{L}_{\mathscr{D}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}') - \mathscr{L}_{\mathscr{D}'}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}')|.
\end{aligned}
$$

□

Lemma 2. *For any* $\delta \in (0,1)$*, with probability at least* $1 - \delta$ *over the choice of the samples, we have*

$$
\mathscr{L}_{\mathscr{D}_{test}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*) \leq \mathscr{L}_{\hat{\mathscr{D}}_k \cup \hat{\mathscr{D}}_{proxy}}(\hat{\boldsymbol{h}}) + \sqrt{\left(\frac{2\alpha^2}{m_k} + \frac{2(1-\alpha)^2}{m_{proxy}} \right) \log \frac{2}{\delta}}. \tag{1}
$$

Proof. The loss $\mathscr{L}_{\hat{\mathscr{D}}_k \cup \hat{\mathscr{D}}_{\text{proxy}}}(\hat{\boldsymbol{h}})$ can be written as

$$
\begin{aligned}
&\mathscr{L}_{\hat{\mathscr{D}}_k \cup \hat{\mathscr{D}}_{\text{proxy}}}(\hat{\boldsymbol{h}}) = \alpha \mathscr{L}_{\hat{\mathscr{D}}_k}(\hat{\boldsymbol{h}}) + (1-\alpha)\mathscr{L}_{\hat{\mathscr{D}}_{\text{proxy}}}(\hat{\boldsymbol{h}}), \\
&= \frac{1}{m_k + m_{\text{proxy}}} \left[\sum_{\boldsymbol{x} \in \hat{\mathscr{D}}_k} \frac{\alpha(m_k + m_{\text{proxy}})}{m_k} \|\hat{\boldsymbol{h}}(\boldsymbol{x}) - \hat{\boldsymbol{h}}^*(\boldsymbol{x})\|_1 + \sum_{\boldsymbol{x} \in \hat{\mathscr{D}}_{\text{proxy}}} \frac{(1-\alpha)(m_k + m_{\text{proxy}})}{m_{\text{proxy}}} \|\hat{\boldsymbol{h}}(\boldsymbol{x}) - \hat{\boldsymbol{h}}^*(\boldsymbol{x})\|_1 \right].
\end{aligned} \tag{2}
$$

Let $X_1^{(k)}, \ldots, X_{m_k}^{(k)}$ and $X_1^{(\text{proxy})}, \ldots, X_{m_{\text{proxy}}}^{(\text{proxy})}$ be independent random variables that take on the values of $\frac{\alpha(m_k + m_{\text{proxy}})}{m_k}\|\hat{\boldsymbol{h}}(\boldsymbol{x}) - \hat{\boldsymbol{h}}^*(\boldsymbol{x})\|_1$ for $\boldsymbol{x} \in \hat{\mathscr{D}}_k$ and $\frac{(1-\alpha)(m_k + m_{\text{proxy}})}{m_{\text{proxy}}}\|\hat{\boldsymbol{h}}(\boldsymbol{x}) - \hat{\boldsymbol{h}}^*(\boldsymbol{x})\|_1$ for $\boldsymbol{x} \in \hat{\mathscr{D}}_{\text{proxy}}$, respectively. We define $\overline{X}$ as the mean value of these variables, which represents the empirical loss in (2). By linearity of expectations, $\mathbb{E}[\overline{X}]$ is equal to the loss $\mathscr{L}_{\mathscr{D}_k \cup \mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}})$. According to Hoeffding's inequality, for any $\varepsilon > 0$, we have

$$
\Pr\left[|\mathscr{L}_{\mathscr{D}_k \cup \mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}) - \mathscr{L}_{\hat{\mathscr{D}}_k \cup \hat{\mathscr{D}}_{\text{proxy}}}(\hat{\boldsymbol{h}})| \geq \varepsilon \right] \leq 2\exp\left(\frac{-\varepsilon^2}{\frac{2\alpha^2}{m_k} + \frac{2(1-\alpha)^2}{m_{\text{proxy}}}} \right). \tag{3}
$$

Let the right-hand side of (3) be δ, we can derive the inequality in Theorem 2. □

We are now ready to prove Theorem 2.

Proof. The following derives the upper bound of $|\mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*) - \mathscr{L}_{\mathscr{D}_k \cup \mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}})|$:

$$
\begin{aligned}
&|\mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*) - \mathscr{L}_{\mathscr{D}_k \cup \mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}})| = |\mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*) - \alpha\mathscr{L}_{\mathscr{D}_k}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*) - (1-\alpha)\mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}, \boldsymbol{h}^*_{\text{proxy}})|, \\
\leq &\alpha|\mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*) - \mathscr{L}_{\mathscr{D}_k}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*)| + (1-\alpha)|\mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*) - \mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}, \boldsymbol{h}^*_{\text{proxy}})|, \\
\leq &\alpha\left[|\mathscr{L}_{\mathscr{D}_k}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*) - \mathscr{L}_{\mathscr{D}_k}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}')| + |\mathscr{L}_{\mathscr{D}_k}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}') - \mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}')| + |\mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}') - \mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*)| \right], \\
&+ (1-\alpha)\left[|\mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}, \boldsymbol{h}^*_{\text{proxy}}) - \mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}')| + |\mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}') - \mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}')| + |\mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}') - \mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}^*)| \right], \\
\leq &\alpha\left[\mathscr{L}_{\mathscr{D}_k}(\hat{\boldsymbol{h}}', \hat{\boldsymbol{h}}^*) + |\mathscr{L}_{\mathscr{D}_k}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}') - \mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}')| + \mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}', \hat{\boldsymbol{h}}^*) \right], \qquad (4) \\
&+ (1-\alpha)\left[\mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}', \boldsymbol{h}^*_{\text{proxy}}) + |\mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}') - \mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}, \hat{\boldsymbol{h}}')| + \mathscr{L}_{\mathscr{D}_{\text{test}}}(\hat{\boldsymbol{h}}', \hat{\boldsymbol{h}}^*) \right], \qquad (5) \\
\leq &\alpha\left[\lambda_k + d_{\mathscr{G}_k}(\mathscr{D}_k, \mathscr{D}_{\text{test}}) \right] + (1-\alpha)\left[\lambda_{k,\text{proxy}} + d_{\mathscr{G}_k}(\mathscr{D}_{\text{proxy}}, \mathscr{D}_{\text{test}}) + \mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}', \boldsymbol{h}^*_{\text{proxy}}) - \mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\hat{\boldsymbol{h}}', \hat{\boldsymbol{h}}^*) \right], \qquad (6) \\
\leq &\alpha\left[\lambda_k + d_{\mathscr{G}_k}(\mathscr{D}_k, \mathscr{D}_{\text{test}}) \right] + (1-\alpha)\left[\lambda_{k,\text{proxy}} + d_{\mathscr{G}_k}(\mathscr{D}_{\text{proxy}}, \mathscr{D}_{\text{test}}) + \mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\boldsymbol{h}^*_{\text{proxy}}, \hat{\boldsymbol{h}}^*) \right], \qquad (7)
\end{aligned}
$$

where

$$\mathscr{L}_{\mathscr{D}_{\text{proxy}}}(\boldsymbol{h}^*_{\text{proxy}}, \hat{\boldsymbol{h}}^*) = p^{(1)}_{\text{proxy}} \mathscr{L}_{\mathscr{D}^{(1)}_{\text{proxy}}}(\hat{\boldsymbol{h}}^*, \boldsymbol{h}^*_{\text{proxy}}) + p^{(2)}_{\text{proxy}} \mathscr{L}_{\mathscr{D}^{(2)}_{\text{proxy}}}(\hat{\boldsymbol{h}}^*_{\text{proxy}}, \boldsymbol{h}^*_{\text{proxy}}). \tag{8}$$

The triangle inequality gives rise to (4), (5), and (7), while Lemma 1 underlies the inequality (6). By combining (7), (8), and Lemma 2, we obtain the upper bound stated in Theorem 2.

□

More Details of Experiments

In the experiments, we use stochastic gradient descent (SGD) with a learning rate of 0.1 as the optimizer. During federated distillation, instead of directly minimizing the empirical loss $\mathscr{L}_{\hat{\mathscr{D}}_k \cup \hat{\mathscr{D}}_{\text{proxy}}}(\hat{\boldsymbol{h}})$ in Theorem 2, clients begin by training their local models independently based on their datasets for 200 SGD steps. In each communication round, clients then fine-tune the models on local samples and proxy samples for 1 and 10 SGD steps, respectively. This training scheme makes the knowledge distillation process more stable. Table 1 provides a summary of datasets, and the architectures of local models are shown in Table 2, Table 3, and Table 4. The convolutional layer with output channel o, kernel size k, and padding p is denoted as $\text{Conv}(o,k,p)$. The fully-connected layer with output dimension o is defined as linear(o), and the max-pooling layer with kernel size k is denoted as MaxPool(k). ReLU represents the rectified linear unit function.

In addition, our method has adopted a portion of the local data as a validation set to determine the threshold τ_{client} of the client-side selectors. The value of τ_{client} corresponds to the quantile of the estimated ratio over the aforementioned validation set. In the weak non-IID scenario, clients are given local training sets containing two classes. As modeling the joint distribution of these classes is challenging and results in high sampling complexity, we utilize an alternative approach where we construct two density-ratio estimators, denoted as $\boldsymbol{w}^{(1)}_k$ and $\boldsymbol{w}^{(2)}_k$, for each class. To determine whether a proxy sample should be treated as an in-distribution sample, we apply the same method for selecting the thresholds $\tau^{(1)}_k$ and $\tau^{(2)}_k$ for $\boldsymbol{w}^{(1)}_k$ and $\boldsymbol{w}^{(2)}_k$, respectively. If any of the estimators output a density ratio higher than its corresponding threshold, the local prediction of the proxy sample will not be filtered out.

Supplementary Tables

Table 1. Summary of datasets.

	COVIDx	MNIST	Fashion MNIST	CIFAR-10
Number of local data	$1000 \sim 2000$	5,400	5,400	4,000
Local training batch size	64	64	64	64
Number of proxy data	1,500	6,000	6,000	10,000
Distillation batch size	128	512	512	32

Table 2. The architectures of local models on the pneumonia detection task.

Client 1	Client 2	Client 3	Client 4
Conv(10, 5, 0) + ReLU	Conv (10, 3, 1) + ReLU	Linear(1024) + ReLU	Linear(1024) + ReLU
MaxPool(2)	MaxPool(2)	Linear(512) + ReLU	Linear(1024) + ReLU
Conv(20, 5, 0) + ReLU	Conv (20, 3, 1) + ReLU	Linear(256) + ReLU	Linear(3)
MaxPool(2)	MaxPool(2)	Linear(3)	
Linear(50) + ReLU	Linear(128) + ReLU		
Linear(3)	Linear(3)		

Table 3. The architectures of local models on the MNIST and Fashion MNIST datasets. Note that the MNIST and Fashion MNIST datasets have images of different sizes, which consequently leads to variations in the input dimensions for the fully-connected layers.

Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8	Client 9	Client 10
Conv(10, 5, 0)	Conv(10, 5, 0)	Conv(10, 3, 1)	Conv(10, 3, 1)	Conv(10, 5, 0)	Conv(10, 5, 0)	Linear(1024)	Linear(1024)	Linear(1024)	Linear(1024)
ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU
MaxPool(2)	MaxPool(2)	MaxPool(2)	MaxPool(2)	MaxPool(2)	MaxPool(2)	Linear(512)	Linear(512)	Linear(1024)	Linear(1024)
Conv(20, 5, 0)	Conv(20, 5, 0)	Conv(20, 3, 1)	Conv(20, 3, 1)	Conv(20, 3, 1)	Conv(20, 3, 1)	ReLU	ReLU	ReLU	ReLU
ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	Linear(256)	Linear(256)	Linear(10)	Linear(10)
MaxPool(2)	MaxPool(2)	MaxPool(2)	MaxPool(2)	MaxPool(2)	MaxPool(2)	ReLU	ReLU		
Linear(50)	Linear(50)	Linear(128)	Linear(128)	Linear(64)	Linear(64)	Linear(10)	Linear(10)		
ReLU	ReLU	ReLU	ReLU	ReLU	ReLU				
Linear(10)	Linear(10)	Linear(10)	Linear(10)	Linear(10)	Linear(10)				

Table 4. The architectures of local models on the CIFAR-10 dataset. ResNet* represents the layers of ResNet before the fully connected layer.

Client 1	Client 2	Client 3	Client 4	Client 5	Client 6	Client 7	Client 8	Client 9	Client 10
ResNet*-18	ResNet*-18	ResNet*-18	ResNet*-18	ResNet*-34	ResNet*-34	ResNet*-34	ResNet*-34	ResNet*-50	ResNet*-50
Linear(128)	Linear(128)	Linear(256)	Linear(256)	Linear(128)	Linear(128)	Linear(256)	Linear(256)	Linear(256)	Linear(256)
ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU	ReLU
Linear(64)	Linear(64)	Linear(10)	Linear(10)	Linear(64)	Linear(64)	Linear(10)	Linear(10)	Linear(10)	Linear(10)
ReLU	ReLU			ReLU	ReLU				
Linear(10)	Linear(10)			Linear(10)	Linear(10)				