Data-Free Knowledge Distillation for Heterogeneous Federated Learning: Supplementary Document

8. Theoretical Derivations

8.1. Notations and Preliminaries

Let $\mathcal{X} \subset \mathbb{R}^p$ be the *input* space, $\mathcal{Z} \subset \mathbb{R}^d$ be the *latent* feature space, and $\mathcal{Y} \subset \mathbb{R}$ be the output space. $\mathcal{R} : \mathcal{X} \to \mathcal{Z}$ denotes a ***representation function*** that maps inputs into features. $\mathcal{T}$ denotes a ***domain*** (or *task*), which consists of a data distribution $\mathcal{D}$ over $\mathcal{X}$ and a ground-truth *labeling* function $c^* : \mathcal{X} \to \mathcal{Y}$. Given a domain $\mathcal{T} := \langle \mathcal{D}, c^* \rangle$ and a representation function $\mathcal{R}$, we use $\tilde{\mathcal{D}}$ to denote the *induced image* of $\mathcal{D}$ under $\mathcal{R}$ (Ben-David et al., 2007), *s.t.* given a probability event $\mathcal{B}$,

$$\mathbb{E}_{z\sim\tilde{\mathcal{D}}}[\mathcal{B}(z)] = \mathbb{E}_{x\sim\mathcal{D}}[\mathcal{B}(\mathcal{R}(x))].$$

Accordingly, $\tilde{c}^*$ denotes the *induced* labeling function under $\mathcal{R}$:

$$\tilde{c}^*(z) := \mathbb{E}_{x\sim D}\left[c^*(x)|\mathcal{R}(x) = z\right].$$

Let $h : \mathcal{Z} \to \mathcal{Y}$ denote a ***hypothesis*** that maps features to predicted labels, and $\mathcal{H} \subseteq \{h : \mathcal{Z} \to \mathcal{Y}\}$ denote a hypothesis class. For our analysis, we assume the FL tasks are for binary classification, *i.e.* $\mathcal{Y} = \{0, 1\}$, and the loss function is 0-1 bounded, with $l(\hat{y}, y) = |\hat{y} - y|$. Same assumptions have been adopted by various prior art (Ben-David et al., 2007; Blitzer et al., 2008; Ben-David et al., 2010; Lin et al., 2020; Ben-David et al., 2007).

Given two distributions $\mathcal{D}$ and $\mathcal{D}'$, $d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}')$ is defined as the $\mathcal{H}$-divergence between $\mathcal{D}$ and $\mathcal{D}'$, *i.e.*:

$$d_{\mathcal{H}}(\mathcal{D}, \mathcal{D}') := 2 \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}}} |\Pr_{\mathcal{D}}(\mathcal{A}) - \Pr_{\mathcal{D}'}(\mathcal{A})|\},$$

where $\mathcal{A}_{\mathcal{H}}$ is a set of measurable subsets under $\mathcal{D}$ and $\mathcal{D}'$ for certain $h \in \mathcal{H}$. Moreover, $\mathcal{H}\triangle\mathcal{H}$ is defined as the symmetric difference hypothesis space (Blitzer et al., 2008), *i.e.*:

$$\mathcal{H}\triangle\mathcal{H} := \{h(z) \oplus h'(z), h, h' \in \mathcal{H}\}$$

where $\oplus$ denotes the XOR operator, so that $h(z) \oplus h'(z)$ indicates that h and h' disagrees with each other. Accordingly, $\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}$ is a set of measurable subsets for $\forall\, h(z) \oplus h'(z) \in \mathcal{H}\triangle\mathcal{H}$. Then $d_{\mathcal{H}\triangle\mathcal{H}}(\cdot, \cdot)$ is defined as the *distribution divergence* induced by the symmetric difference hypothesis space (Blitzer et al., 2008):

$$d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}, \mathcal{D}') := 2 \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}} |\Pr_{\mathcal{D}}(\mathcal{A}) - \Pr_{\mathcal{D}'}(\mathcal{A})|\}.$$

Specifically, let $\mathcal{D}, \mathcal{D}'$ be two arbitrary distributions on the input space $\mathcal{X}$, and let $\tilde{\mathcal{D}}, \tilde{\mathcal{D}}'$ be their induced images over $\mathcal{R}$. Then based on the definition of $d_{\mathcal{H}\triangle\mathcal{H}}(\cdot, \cdot)$, one can have:

$$\begin{aligned} d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}, \tilde{\mathcal{D}}') &= 2 \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}} \left|\mathbb{E}_{x\sim\mathcal{D}}\left[\Pr(\mathcal{A}(\mathcal{R}(x)))\right] - \mathbb{E}_{x\sim\mathcal{D}'}\left[\Pr(\mathcal{A}(\mathcal{R}(x)))\right]\right| \\ &= 2 \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}} \left|\mathbb{E}_{z\sim\tilde{\mathcal{D}}}\left[\Pr(\mathcal{A}(x))\right] - \mathbb{E}_{z\sim\tilde{\mathcal{D}}'}\left[\Pr(\mathcal{A}(z))\right]\right| \\ &= 2 \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}} |\Pr_{\tilde{\mathcal{D}}}(\mathcal{A}) - \Pr_{\tilde{\mathcal{D}}'}(\mathcal{A})|\}. \end{aligned}$$

8.2. Derivations of Remark 1

Remark. *Let $p(y)$ be the prior distribution of labels, and $r(z|y) : \mathcal{Y} \to \mathcal{Z}$ be the conditional distribution derived from generator $G_{\boldsymbol{w}}$. Then regulating a user model $\boldsymbol{\theta}_k$ using samples from $r(z|y)$ can minimize the conditional KL-divergence*

between two distributions, derived from the user and from the generator, respectively:

$$\max_{\boldsymbol{\theta}_k} \mathbb{E}_{y\sim p(y), z\sim r(z|y)} \left[\log p(y|z;\boldsymbol{\theta}_k)\right] \equiv \min_{\boldsymbol{\theta}_k} D_{\mathrm{KL}}[r(z|y)\|p(z|y;\boldsymbol{\theta}_k)],$$

Proof. Expanding the KL-divergence, we have

$$\begin{aligned}
\because\ D_{\mathrm{KL}}[r(z|y)\|p(z|y;\boldsymbol{\theta}_k)] &\equiv \mathbb{E}_{y\sim p(y)}\left[\mathbb{E}_{z\sim r(z|y)}\left[\log \frac{r(z|y)}{p(z|y;\boldsymbol{\theta})}\right]\right] \\
&= \mathbb{E}_{y\sim p(y)}\mathbb{E}_{z\sim r(z|y)}\left[\log r(z|y)\right] - \mathbb{E}_{y\sim p(y)}\mathbb{E}_{z\sim r(z|y)}\left[\log p(z|y;\boldsymbol{\theta})\right] \\
&= \underbrace{-H(r(z|y))}_{\text{constant w.r.t }\boldsymbol{\theta}_k} - \mathbb{E}_{y\sim p(y)}\mathbb{E}_{z\sim r(z|y)}[\log p(z|y;\boldsymbol{\theta})].
\end{aligned}$$

where $H(r(z|y))$ is constant w.r.t $\boldsymbol{\theta}_k$. Therefore when optimizing $\boldsymbol{\theta}_k$ we have:

$$\begin{aligned}
&\min_{\boldsymbol{\theta}_k} D_{\mathrm{KL}}[r(z|y)\|p(z|y;\boldsymbol{\theta}_k)] \\
\equiv &\min_{\boldsymbol{\theta}_k} -\mathbb{E}_{y\sim p(y), z\sim r(z|y)}\left[\log p(z|y;\boldsymbol{\theta}_k)\right] \\
\equiv &\max_{\boldsymbol{\theta}_k} \mathbb{E}_{y\sim p(y)}\mathbb{E}_{z\sim r(z|y)}[\log \frac{p(y|z;\boldsymbol{\theta}_k)p(z)}{p(y)}] \\
\equiv &\max_{\boldsymbol{\theta}_k} \mathbb{E}_{y\sim p(y)}\mathbb{E}_{z\sim r(z|y)}[\log p(y|z;\boldsymbol{\theta}_k) + \log p(z) - \log p(y)] \\
\equiv &\max_{\boldsymbol{\theta}_k} \mathbb{E}_{y\sim p(y)}\mathbb{E}_{z\sim r(z|y)}[\log p(y|z;\boldsymbol{\theta}_k)].
\end{aligned}$$

where $H(r(z|y))$ denotes the entropy of the probability distribution $r(z|y)$ which is *not* optimizable w.r.t $\boldsymbol{\theta}_k$, and $p(z|y;\boldsymbol{\theta}_k) := \frac{p(y|z;\boldsymbol{\theta}_k)p(z)}{p(y)}$ is defined as the probability that the input representation to the predictor is z if it yields a label y. □

8.3. Derivations of Theorem 1

Before deriving Theorem 1, we first present an upper-bound for the generalization performance from prior art (Ben-David et al., 2007), which analyzes the role of a feature representation function in the context of *domain adaptation*:

Lemma 1. Generalization Bounds for Domain Adaptation (Ben-David et al., 2007; Blitzer et al., 2008):

Let $\mathcal{T}_S$ and $\mathcal{T}_T$ be the source and target domains, whose data distributions are $\mathcal{D}_S$ and $\mathcal{D}_T$. Let $\mathcal{R}: \mathcal{X} \to \mathcal{Z}$ be a feature representation function, and $\tilde{\mathcal{D}}_S, \tilde{\mathcal{D}}_T$ be the induced images of $\mathcal{D}_S$ and $\mathcal{D}_T$ over $\mathcal{R}$, respectively. Let $\mathcal{H}$ be a set of hypothesis with VC-dimension d. Then with probability at least $1-\delta$, $\forall\ h \in \mathcal{H}$:

$$\mathcal{L}_{\mathcal{T}_T}(h) \le \hat{\mathcal{L}}_{\mathcal{T}_S}(h) + \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4}{\delta}\right)} + d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_S, \tilde{\mathcal{D}}_T) + \lambda, \tag{8}$$

where e is the base of the natural logarithm, $\hat{\mathcal{L}}_{\mathcal{T}_S}$(h) is the empirical risk of the source domain given m observable samples, and $\lambda = \min_{h\in\mathcal{H}}\left(\mathcal{L}_{\mathcal{T}_T}(h) + \mathcal{L}_{\mathcal{T}_S}(h)\right)$ is the optimal risk on the two domains.

One insight from Lemma 1 is that a good representation function plays a tradeoff between minimizing the empirical risk ($\hat{\mathcal{L}}_{\mathcal{T}_S}(h)$) and the induced distributional discrepancy ($d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_S, \tilde{\mathcal{D}}_T)$). Based on Lemma 1, one can establish Theorem 1 as the following:

Theorem. (**Generalization Bounds for FL**) *Consider an FL system with K users. Let $\mathcal{T}_k = \langle \mathcal{D}_k, c^*\rangle$ and $\mathcal{T} = \langle \mathcal{D}, c^*\rangle$ be the k-th local domain and the global domain, respectively. Let $\mathcal{R}: \mathcal{X} \to \mathcal{Z}$ be a feature extraction function that is simultaneously shared among users. Let h_k denote the hypothesis learned on domain $\mathcal{T}_k$, and $h = \frac{1}{K}\sum_{k=1}^{K} h_k$ be the global ensemble of user predictors. Then with probability at least $1-\delta$:*

$$\mathcal{L}_{\mathcal{T}}(h) \le \frac{1}{K}\sum_{k\in[K]} \hat{\mathcal{L}}_{\mathcal{T}_k}(h_k) + \frac{1}{K}\sum_{k\in[K]}\left(d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}) + \lambda_k\right) + \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4K}{\delta}\right)},$$

*where $\hat{\mathcal{L}}_{\mathcal{T}_k}(h_k)$ is the empirical risk of h_k, $\lambda_k := \min_h(\mathcal{L}_{\mathcal{T}_k}(h) + \mathcal{L}_{\mathcal{T}}(h))$ denotes an oracle performance on $\mathcal{T}_k$ and $\mathcal{T}$, and $\tilde{\mathcal{D}}_k$ and $\tilde{\mathcal{D}}$ is the **induced** image of $\mathcal{D}_k$ and $\mathcal{D}$ from $\mathcal{R}$, respectively, s.t. $\mathbb{E}_{z\sim\tilde{\mathcal{D}}_k}[\mathcal{B}(z)] = \mathbb{E}_{x\sim\mathcal{D}_k}[\mathcal{B}(\mathcal{R}(x))]$ given a probability event $\mathcal{B}$, and so for $\tilde{\mathcal{D}}$.*

Proof. By treating each one of the local domains $k \in [K]$ as the *source* and the global domain as the *target*, one can have that, $\forall\ \delta > 0$, with probability $1 - \frac{\delta}{K}$:

$$\mathcal{L}_{\mathcal{T}}(h_k) \leq \hat{\mathcal{L}}_{\mathcal{T}_k}(h_k) + d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}) + \lambda_k + \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4K}{\delta}\right)}.$$

Also, due to the convexity of risk function and Jesen inequality, one can have:

$$\mathcal{L}_{\mathcal{T}}(h) \equiv \mathcal{L}_{\mathcal{T}}\left(\frac{1}{K}\sum_{k\in[K]} h_k\right) \leq \frac{1}{K}\sum_{k\in[K]}\mathcal{L}_{\mathcal{T}}(h_k).$$

Therefore,

$$\begin{aligned}
&\Pr\left[\mathcal{L}_{\mathcal{T}}(h) > \frac{1}{K}\sum_{k\in[K]}\left(\hat{\mathcal{L}}_{\mathcal{T}_k}(h_k) + \sum_{k\in[K]}(d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}) + \lambda_k) + \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4K}{\delta}\right)}\right)\right] \\
\leq &\Pr\left[\frac{1}{K}\sum_{k\in[K]}\mathcal{L}_{\mathcal{T}}(h_k) > \frac{1}{K}\sum_{k\in[K]}\left(\hat{\mathcal{L}}_{\mathcal{T}_k}(h_k) + \sum_{k\in[K]}(d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}) + \lambda_k) + \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4K}{\delta}\right)}\right)\right] \\
\leq &\Pr\left[\bigvee_{k\in[K]}\mathcal{L}_{\mathcal{T}}(h_k) > \hat{\mathcal{L}}_{\mathcal{T}_k}(h_k) + d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}) + \lambda_k + \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4K}{\delta}\right)}\right] \\
\leq &\sum_{k\in[K]}\frac{\delta}{K} = \delta.
\end{aligned}$$

□

Theorem 1 shows that the performance of the aggregated hypothesis is upper-bounded by: 1) the local performance of each user hypothesis ($\hat{\mathcal{L}}_{\mathcal{T}_k}(h_k)$), 2) the dissimilarity between the global and local distributions over the feature space ($d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}})$), 3) the oracle performance (λ_k), and 4) the numerical constraints regarding the number of empirical samples m and the VC-dimension d.

8.4. Derivations of Corollary 1

Corollary. *Let $\mathcal{T}$, $\mathcal{T}_k$, $\mathcal{R}$ defined as in Theorem 1. $\mathcal{D}_{\mathrm{A}}$ denotes an **augmented** data distribution, and $\mathcal{D}'_k = \frac{1}{2}(\mathcal{D}_k + \mathcal{D}_{\mathrm{A}})$ is a **mixture** of distributions. Accordingly, $\tilde{\mathcal{D}}_{\mathrm{A}}$ and $\tilde{\mathcal{D}}'_k$ denote the **induced** image of $\mathcal{D}_{\mathrm{A}}$ and $\mathcal{D}'_k$ over $\mathcal{R}$, respectively. Let $\hat{\mathcal{D}}'_k = \hat{\mathcal{D}}_k \cup \hat{\mathcal{D}}_{\mathrm{A}}$ be an empirical dataset of $\mathcal{D}'_k$, with $|\hat{\mathcal{D}}_k|$=m, $|\hat{\mathcal{D}}'_k| = |\hat{\mathcal{D}}_k| + |\hat{\mathcal{D}}_{\mathrm{A}}| = m'$. Assume the discrepancy between $\tilde{\mathcal{D}}_{\mathrm{A}}$ and $\tilde{\mathcal{D}}$ is bounded, s.t $\exists\ \epsilon > 0, d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_A, \tilde{\mathcal{D}}) \leq \epsilon$, then with probability $1 - \delta$:*

$$\mathcal{L}_{\mathcal{T}}(h) \leq \frac{1}{K}\sum_k \mathcal{L}_{\mathcal{T}'_k}(h_k) + \frac{1}{K}\sum_k (d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}'_k, \tilde{\mathcal{D}})) + \frac{1}{K}\sum_k \lambda'_k + \sqrt{\frac{4}{m'}\left(d\log\frac{2em'}{d} + \log\frac{4K}{\delta}\right)}, \tag{9}$$

where $\mathcal{T}'_k = \{\mathcal{D}'_k, c^*\}$ is the updated local domain, $\lambda'_k = \min_h(\mathcal{L}_{\mathcal{T}'_k}(h) + \mathcal{L}_{\mathcal{T}}(h))$ denotes the oracle performance, and $d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}'_k, \tilde{\mathcal{D}}) \leq d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}})$ when ϵ is small.

Proof. Equation 9 can be directly derived by Theorem 1. We now focus on analyzing the relation between $d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}})$ and $d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}'_k, \tilde{\mathcal{D}})$, which is the data dissimilarity ***before*** and ***after*** data augmentation using samples from distribution $\mathcal{D}_A$, respectively.

Based on the definition of $d_{\mathcal{H}\triangle\mathcal{H}}(\cdot,\cdot)$, one can derive that:

$$
\begin{aligned}
& d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}'_k, \tilde{\mathcal{D}}) \\
=& 2 \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}} \left| \mathbb{E}_{z\sim\tilde{\mathcal{D}}'_k} \left[\Pr(\mathcal{A}(z))\right] - \mathbb{E}_{z\sim\tilde{\mathcal{D}}} \left[\Pr(\mathcal{A}(z))\right] \right| \\
=& 2 \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}} \left| \mathbb{E}_{z\sim\frac{1}{2}(\tilde{\mathcal{D}}_k+\tilde{\mathcal{D}}_A)} \left[\Pr(\mathcal{A}(z))\right] - \mathbb{E}_{z\sim\tilde{\mathcal{D}}} \left[\Pr(\mathcal{A}(z))\right] \right| \\
=& 2 \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}} \left| \frac{1}{2}\mathbb{E}_{z\sim\tilde{\mathcal{D}}_K} \left[\Pr(\mathcal{A}(z))\right] + \frac{1}{2}\mathbb{E}_{z\sim\tilde{\mathcal{D}}_A} \left[\Pr(\mathcal{A}(z))\right] - \mathbb{E}_{z\sim\tilde{\mathcal{D}}} \left[\Pr(\mathcal{A}(z))\right] \right| \\
\leq & \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}} \left| \mathbb{E}_{z\sim\tilde{\mathcal{D}}_K} \left[\Pr(\mathcal{A}(z))\right] - \mathbb{E}_{z\sim\tilde{\mathcal{D}}} \left[\Pr(\mathcal{A}(z))\right] \right| + \sup_{\mathcal{A}\in\mathcal{A}_{\mathcal{H}\triangle\mathcal{H}}} \left| \mathbb{E}_{z\sim\tilde{\mathcal{D}}_A} \left[\Pr(\mathcal{A}(z))\right] - \mathbb{E}_{z\sim\tilde{\mathcal{D}}} \left[\Pr(\mathcal{A}(z))\right] \right| \\
=& \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}) + \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_A, \tilde{\mathcal{D}}).
\end{aligned}
$$

It is clear that $\frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_A, \tilde{\mathcal{D}})$, which is bounded by ϵ, affects the dissimilarity between the *induced* image of local and the global distribution, therefore plays a key role in upper-bounding the global performance ($\mathcal{L}_{\mathcal{T}}(h)$ in Equation 9). Next, we discuss different scenarios when FL can benefit from such augmented data, and when the quality of augmented distribution $\mathcal{D}_A$ can limit the generalization performance of the aggregated model.

$\mathcal{D}_A$ can benefit local users when ϵ is small: To see this, one can assume that:

$$
d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_A, \tilde{\mathcal{D}}) = \epsilon \leq \min_k d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}),
$$

of which the intuition is that, after feature mapping, the discrepancy between the augmented distribution and the global distribution is smaller than the discrepancy between an individual user and the global. Based on this assumption, one can conclude that $\forall\ \mathcal{T}_k \in \boldsymbol{\mathcal{T}}$:

$$
\begin{aligned}
d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}'_k, \tilde{\mathcal{D}}) =& \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}) + \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_A, \tilde{\mathcal{D}}) \\
\leq & \frac{1}{2} d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}) + \min_j d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_j, \tilde{\mathcal{D}}) \\
\leq & d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}),
\end{aligned}
$$

Therefore, a small $d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_A, \tilde{\mathcal{D}})$ benefits local users w.r.t their generalization performance, by both reducing the data discrepancy and enriching the empirical samples, in that:

$$
\mathcal{L}_{\mathcal{T}}(h_k) \leq \mathcal{L}_{\mathcal{T}'_k}(h_k) + \lambda'_k + \underbrace{\leq d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}'_k, \tilde{\mathcal{D}})}_{\leq d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}})} + \underbrace{\sqrt{\frac{4}{m'}\left(d\log\frac{2em'}{d} + \log\frac{4}{\delta}\right)}}_{\leq\sqrt{\frac{4}{m}\left(d\log\frac{2em}{d}+\log\frac{4}{\delta}\right)}} \qquad \text{(Derived from Lemma 1).}
$$

$\mathcal{D}_A$ has positive effects on the generalization performance when ϵ is moderate: Instead, one might as well assume that

$$
d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_A, \tilde{\mathcal{D}}) = \epsilon \leq \frac{1}{K}\sum_{k=1}^{K} d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}),
$$

which implies that, after feature mapping over $\mathcal{R}$, the dissimilarity between $\mathcal{D}_A$ and the global distribution $\mathcal{D}$ is at least as small as the *average* dissimilarity between local users and the global. Based on this assumption, one can derive that:

$$
\sum_k d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}'_k, \tilde{\mathcal{D}}) \leq \sum_k d_{\mathcal{H}\triangle\mathcal{H}}(\tilde{\mathcal{D}}_k, \tilde{\mathcal{D}}),\ \sqrt{\frac{4}{m'}\left(d\log\frac{2em'}{d} + \log\frac{4}{\delta}\right)} \leq \sqrt{\frac{4}{m}\left(d\log\frac{2em}{d} + \log\frac{4}{\delta}\right)},
$$

which can still contribute to a tighter upper-bound for the global performance in Equation 9, compared with not using the augmented data.

Conversely, when ϵ is over-large, which implies that $\mathcal{D}_A$ is not relevant to the original FL task, it may have negative impacts on the generalization performance. □

9. Extended Experiments

We first discuss some practical considerations for implementing our algorithm:

- **Weighting user models:** User models vary in their ability to predict certain labels over others due to their statistical heterogeneity. Therefore, we use the number of training labels available to users to summarize a weight matrix $\mathbf{\Lambda} = \{\lambda_k^c | c \in \mathcal{Y}, k \in \{1, 2, \cdots, K\}\}$, *s.t.* $\forall c, i, j,\ \frac{\lambda_i^c}{\lambda_j^c} = \frac{n_i^c}{n_j^c}$ indicates the ratio of training samples for label c between two users i and j, and $\sum_k \lambda_k^c = 1\ \forall c \in \mathcal{Y}$. We then apply this weight matrix to adjust the generator objective as the following:

$$\min_{\boldsymbol{w}} J(\boldsymbol{w}) := \mathbb{E}_{y\sim\hat{p}(y)}\mathbb{E}_{z\sim G_{\boldsymbol{w}}(z|y)}\left[\lambda_k^y l\left(\sigma\left(\frac{1}{K}\sum\nolimits_{k=1}^{K} g(z;\boldsymbol{\theta}_k^p)\right), y\right)\right].$$

  We found that this weighted objective can further mitigate the impact of negative ensemble, especially when a teacher model is too weak to predict certain labels due to lacking training samples of that category.

- **Stochastic generative learning:** Built upon prior arts on generative learning (Kingma & Welling, 2014), we use an auxiliary noise vector with dimension d_n to infer the desirable feature representation for a given label y, *s.t.* $z \sim G_{\boldsymbol{w}}(\cdot|y) \equiv G_{\boldsymbol{w}}(y, \epsilon|\epsilon \sim \mathcal{N}(0, I))$. To further increase the diversity of the generator output, we also leverage the idea of *diversity loss* from prior work (Mao et al., 2019) to train the generator model.

9.1. Prototype Results

We adopt an one-round FL setting for the prototype experiment, for which the dataset distributions of local users, as well as their model decision boundaries *before* and *after* knowledge distillation, are illustrated in Figure 9. Accuracy of user models on the global dataset is also summarized in Table 5, from which one can observe that the generalization performance of user models have been notably improved by the distilled knowledge.

	User 1	User 2	User 3	Oracle
Before	97.1	81.3	81.2	98.4
After	98.6	98.3	98.2	

Table 5. Accuracy (%) before and after KD.

9.2. Experimental Setup

We provide the network architecture for the generator and the classifier in Table 6 and Table 7. For the generator $G_{\boldsymbol{w}}$, we adopt a two-MLP layer network. It takes a noise vector ϵ and an one-hot label vector y as the input, which, after a hidden layer with dimension d_h, outputs a feature representation with dimension d. For the classifier, we adopt a network architecture with a CNN module followed by a MLP module. Hyperparameter settings for the experiments are provided in Table 8.

Dataset	Hyperparameter	Value
CELEBA	d_n, d_h, d	32, 128, 32
MNIST& EMNIST	d_n, d_h, d	32, 256, 32

Table 6. Network architecture for the generator $G_{\boldsymbol{w}}$.

Dataset	Hyperparameter	Value
CELEBA	CNN Module	[16, M, 32, M, 64]
	MLP Module	[784, 32]
MNIST & EMNIST	CNN Module	[6, 16]
	MLP Module	[784, 32]

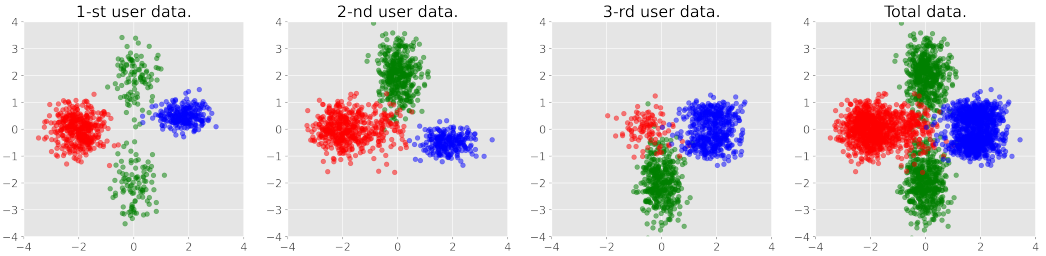
Table 7. Network architecture for the classification model.

9.3. FEDGEN with Partial Parameter Sharing

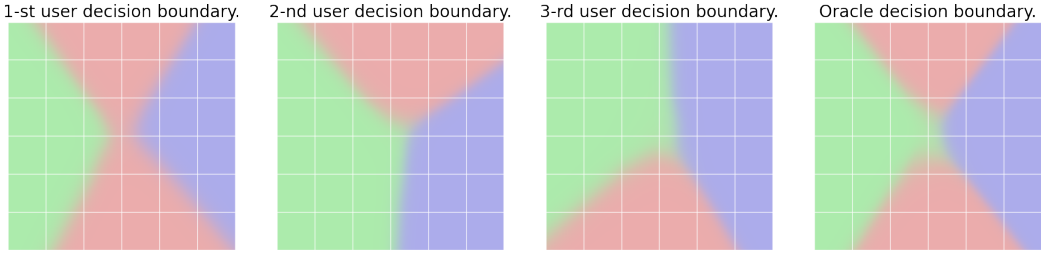
Algorithm 2 summarizes an variant approach of FEDGEN for a specific FL setting, where only the last prediction layer is shared among users while keeping the feature extraction layers localized.

9.4. Extended Experimental Results

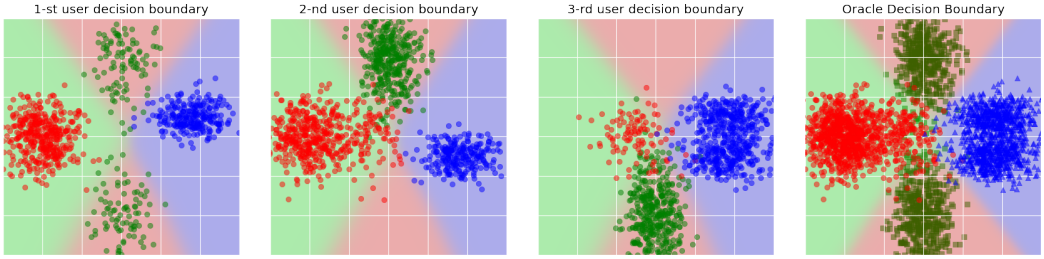
We elaborate the learning curves trained on the MNIST, CELEBA, and EMNIST dataset in Figure 10, Figure 11, and Figure 12, respectively, with their performance summarized in Table 9.



(a) Local user data distribution and total data distribution.



(b) User models generate biased decision boundaries **before** KD, provided with incomplete local data.



(c) Decision Boundaries of user models are improved **after** KD.

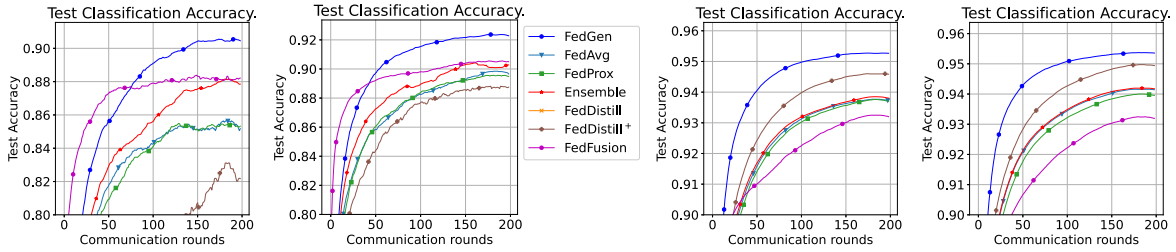
Figure 9. Knowledge distillation process for the prototype experiment.

Algorithm 2 FEDGEN with Partial Parameter Sharing

Require: Tasks $\mathcal{T}_k, k \in \{1, \cdots, K\}$;
Global predictor $\boldsymbol{\theta}^p$, local parameters $\{\boldsymbol{\theta}_k = [\boldsymbol{\theta}_k^f; \boldsymbol{\theta}_k^p]\}_{k=1}^K$;
Generator parameter $\boldsymbol{w}$; $\hat{p}(y)$ uniformly initialized;
Learning rate α, β, local steps T, batch size B, local label counter c_k.
repeat
  Server selects active users $\mathcal{A}$ uniformly at random, then broadcast $\boldsymbol{w}, \boldsymbol{\theta}^p$, $\hat{p}(y)$ to $\mathcal{A}$.
  for all user $k \in \mathcal{A}$ in parallel **do**
    $\boldsymbol{\theta}_k^p \leftarrow \boldsymbol{\theta}^p$,
    for $t = 1, \ldots, T$ **do**
      $\{x_i, y_i\}_{i=1}^B \sim \mathcal{T}_k$, $\{\hat{z}_i \sim G_{\boldsymbol{w}}(\cdot|\hat{y}_i), \hat{y}_i \sim \hat{p}(y)\}_{i=1}^B$.
      Update label counter c_k.
    $\boldsymbol{\theta}_k \leftarrow \boldsymbol{\theta}_k - \beta \nabla_{\boldsymbol{\theta}_k} J(\boldsymbol{\theta}_k)$.
    end for
    User sends $\boldsymbol{\theta}_k^p$, c_k back to server.
  end for
  Server updates $\boldsymbol{\theta}^p \leftarrow \frac{1}{|\mathcal{A}|} \sum_{k \in \mathcal{A}} \boldsymbol{\theta}_k^p$, and $\hat{p}(y)$ based on $\{c_k\}_{k \in \mathcal{A}}$.
  $\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \nabla_{\boldsymbol{w}} J(\boldsymbol{w})$.
until training stop

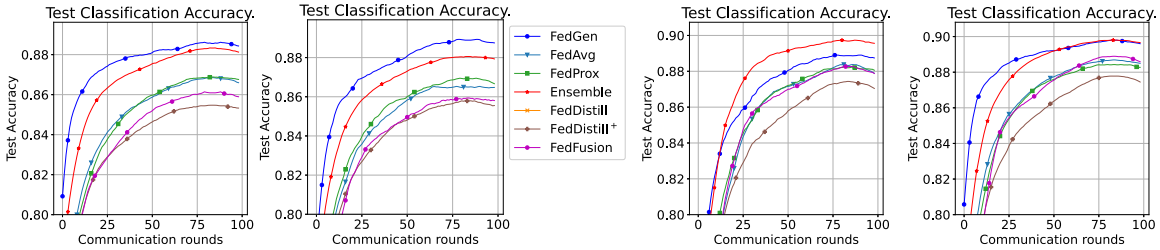
	Hyperparameter	Value
Shared Parameters	Learning rate	0.01
	Optimizer	sgd
	Local update steps (T)	20
	Batch size (B)	32
	Communication rounds	200
	# of total users	20
	# of active users	10
FEDDFUSION	Ensemble Optimizer	adam
	Generator learning rate	10^{-4}
	Ensemble batch size	128
FEDGEN	Generator Optimizer	adam
	Generator learning rate	10^{-4}
	Generator inference size	128
	User distillation batch size	32
FEDDISTILL& FEDDISTILL$^+$	Distillation coefficient	0.1
FEDPROX	Proximal coefficient	0.1

Table 8. We use the above configurations for experiments unless mentioned otherwise.



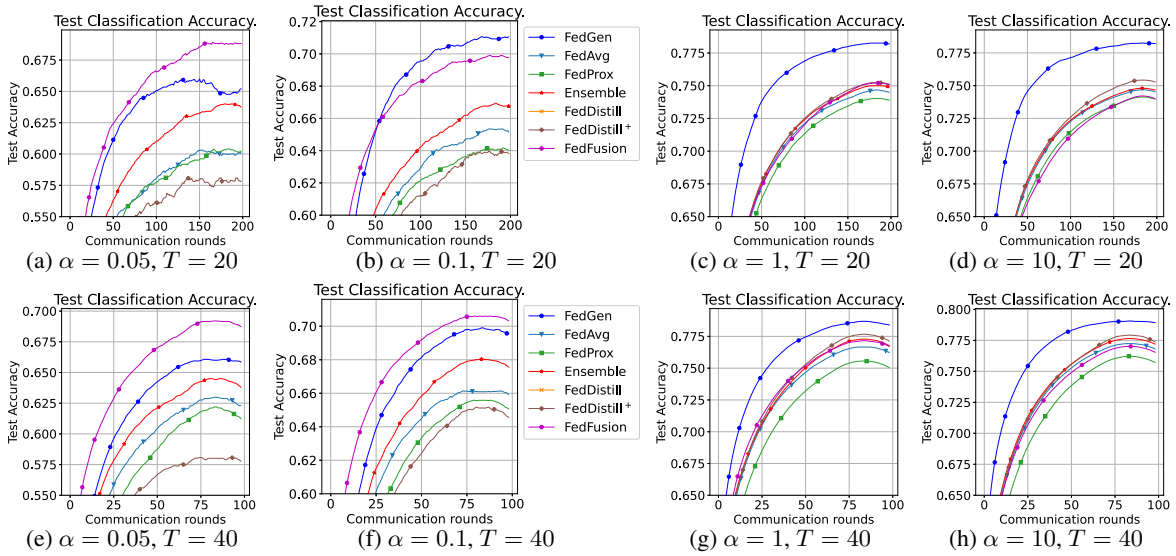
(a) $\alpha = 0.05$ (b) $\alpha = 0.1$ (c) $\alpha = 1$ (d) $\alpha = 10$

Figure 10. Performance curves on MNIST dataset, where a smaller α denotes larger data heterogeneity.



(a) CELEBA, $r = 9/10$. (b) CELEBA, $r = 5/10$. (c) CELEBA, $r = 5/25$. (d) CELEBA, $r = 10/25$.

Figure 11. Performance curves on CELEBA dataset.



(a) $\alpha = 0.05, T = 20$ (b) $\alpha = 0.1, T = 20$ (c) $\alpha = 1, T = 20$ (d) $\alpha = 10, T = 20$

(e) $\alpha = 0.05, T = 40$ (f) $\alpha = 0.1, T = 40$ (g) $\alpha = 1, T = 40$ (h) $\alpha = 10, T = 40$

Figure 12. Performance curves on EMNIST dataset, under different data heterogeneity and communication frequencies.

Top-1 Test Accuracy.								
Dataset	Setting	FEDAVG	FEDPROX	FEDENSEMBLE	FEDDISTILL	FEDDISTILL$^+$	FEDDFUSION	FEDGEN
MNIST	$\alpha = 0.05$	87.70±2.07	87.49±2.05	88.85±0.68	70.56±1.24	86.70±2.27	90.02±0.96	**91.30±0.74**
	$\alpha = 0.1$	90.16±0.59	90.10±0.39	90.78±0.39	64.11± 1.36	90.28±0.89	91.11±0.43	**93.03±0.32**
	$\alpha = 1$	93.84±0.25	93.83 ± 0.29	93.91±0.28	79.88±0.66	94.73±0.15	93.37±0.40	**95.52±0.07**
	$\alpha = 10$	94.23±0.13	94.06±0.10	94.25±0.11	89.21±0.26	95.04±0.21	93.36±0.45	**95.79±0.10**
CELEBA	$r = 5/10$	87.48±0.39	87.67±0.39	88.48±0.23	76.68±1.23	86.37±0.41	87.01±1.00	**89.70±0.32**
	$r = 5/25$	89.13±0.25	88.84±0.19	**90.22±0.31**	74.99±1.57	88.05± 0.43	88.93±0.79	89.62±0.34
	$r = 10/25$	89.12±0.20	89.01±0.33	90.08±0.24	75.88±1.17	88.14±0.37	89.25±0.56	**90.29±0.47**
EMNIST, T=20	$\alpha = 0.05$	62.25±2.82	61.93±2.31	64.99±0.35	60.49±1.27	61.56±2.15	**70.40±0.79**	68.53±1.17
	$\alpha = 0.1$	66.21±2.43	65.29±2.94	67.53±1.19	50.32±1.39	66.06±3.18	70.94±0.76	**72.15±0.21**
	$\alpha = 1$	74.83±0.99	74.12±0.88	75.12±1.07	46.19±0.70	75.41±1.05	75.43±0.37	**78.48±1.04**
	$\alpha = 10$	74.83± 0.69	74.24±0.81	74.90±0.80	54.77±0.33	75.55 ±0.94	74.36±0.40	**78.43±0.74**
EMNIST, T=40	$\alpha = 0.05$	64.51±1.13	63.60±0.69	65.74±0.45	60.73±1.62	60.73±1.06	**70.46±1.16**	67.64 ±0.75
	$\alpha = 0.1$	67.71±1.31	66.79±0.77	68.96±0.66	49.54±1.18	67.01±0.38	**71.55±0.43**	70.90 ±0.49
	$\alpha = 1$	77.02±1.09	75.93 ±0.95	77.68±0.98	46.72±0.73	78.12±0.90	77.58±0.37	**78.92± 0.73**
	$\alpha = 10$	77.52±0.66	76.54±0.71	77.92±0.62	54.85±0.44	78.37±0.76	77.31±0.45	**79.29±0.53**

Table 9. Performance overview under different data heterogeneity settings. For MNIST and EMNIST, user data follows the Dirichlet distribution with hyperparameter α, with a **smaller** α indicating higher heterogeneity. For CELEBA, r denotes the ratio between active users and total users. T denotes the local training steps (communication delay). All above experiments use batch size B=32.