A PROOFS OF THEOREM 2.1 AND THEOREM 2.2

A.1 PRELIMINARIES

FedTrans integrates two client utility estimation strategies: weight-based estimation and performance-based estimation. We formulate a Bayesian framework shown in Figure 3, wherein the parameters are estimated by employing the maximum likelihood estimate (MLE) approach. We first briefly introduce MLE under latent variables settings, and from that derive the parameter estimate rules for our framework. The general MLE problem is formulated as follows:

$$\phi^*_{MLE} = \arg\max_{\phi \in \Phi} p(\mathbf{O}; \phi), \tag{17}$$

where ϕ are the parameters of the probabilistic model and $\mathbf{O}$ are a set of observations.

When it is infeasible to directly model the likelihood function $p(\mathbf{O}; \phi)$ (as in our problem), we introduce latent variables $\mathbf{L}$ to connect the observations to unknown parameters. In this way, the likelihood function is transformed into

$$p(\mathbf{O}; \phi) = \int p(\mathbf{O}, \mathbf{L}; \phi) d\mathbf{L}. \tag{18}$$

The expectation maximization (EM) algorithm is employed for MLE with latent variables. The algorithm iteratively updates the elements in the likelihood function. Specifically, the E-step iteratively optimizes the latent variables $\mathbf{L}$, and the M-step iteratively optimizes the parameters ϕ.

For the problem in this paper, the observations refer to the round reputation matrix $\mathbf{R}$ conditioned on the weight parameters of top-layer $\mathbf{X}$; the latent variables are $\mathbf{r}$ and $\mathbf{s}$; and the parameters are $\{\mathcal{W}_d, A, B\}$, hereafter denoted as $\mathcal{P}$. A detailed expression of the likelihood function is as follows:

$$\begin{aligned} p(\mathbf{R}|\mathbf{X}; \mathcal{P}) &= \prod_i \prod_j p(\mathbf{R}_{i,j}|\mathbf{x}_j; \mathcal{P}) \\ &= \prod_i \prod_j \int_{r_i, s_j} p(\mathbf{R}_{i,j}, r_i, s_j|\mathbf{x}_j; \mathcal{P}) dr_i, s_j \\ &= \int p(\mathbf{R}, \mathbf{r}, \mathbf{s}|\mathbf{X}; \mathcal{W}_d) d\mathbf{r}, \mathbf{s}. \end{aligned} \tag{19}$$

We first decompose the likelihood function in Equation 6 into two terms as shown in Equation 7 of Section 2.2. We then employ variational EM to optimize variables in the Bayesian framework, wherein the E-step and M-step are designed to iteratively update the variables related to both the two modules. The reason for choosing the variational EM instead of the closed-form EM will also be discussed in the following section.

After applying the logarithm on Equation 19, we get

$$\begin{aligned} \log p(\mathbf{R}|\mathbf{X}; \mathcal{P}) &= \log \left(\prod_i \prod_j \int_{r_i, s_j} p(\mathbf{R}_{i,j}, r_i, s_j|\mathbf{x}_j; \mathcal{P}) dr_i, s_j \right) \\ &= \sum_i \sum_j \log \left(\int_{r_i, s_j} p(\mathbf{R}_{i,j}, r_i, s_j|\mathbf{x}_j; \mathcal{P}) dr_i, s_j \right) \\ &= \sum_i \sum_j \log \left(\int_{r_i, s_j} q(r_i, s_j) \frac{p(\mathbf{R}_{i,j}, r_i, s_j|\mathbf{x}_j; \mathcal{P})}{q(r_i, s_j)} dr_i, s_j \right), \end{aligned} \tag{20}$$

where the approximation term $q(r_i, s_j)$ can be any probability density function.

According to Jensen's Inequality, we have described the entire process of

$$\begin{aligned}
&\log\left(\int_{r_i,s_j} q(r_i,s_j)\frac{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}{q(r_i,s_j)}dr_i,s_j\right)\\
&=\log\left(\mathbb{E}\left[\frac{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}{q(r_i,s_j)}\right]\right)\\
&\geq\mathbb{E}\left[\log\left(\frac{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}{q(r_i,s_j)}\right)\right]\\
&=\int_{r_i,s_j} q(r_i,s_j)\log\left(\frac{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}{q(r_i,s_j)}\right)dr_i,s_j.
\end{aligned}\tag{21}$$

Therefore, we have

$$\log p(\mathbf{R}|\mathbf{X};\mathcal{P})\geq\sum_i\sum_j\int_{r_i,s_j} q(r_i,s_j)\log\left(\frac{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}{q(r_i,s_j)}\right)dr_i,s_j.\tag{22}$$

The difference between the two sides of the inequality is

$$\begin{aligned}
\Delta&=\log p(\mathbf{R}|\mathbf{X};\mathcal{P})-\sum_i\sum_j\int_{r_i,s_j} q(r_i,s_j)\log\left(\frac{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}{q(r_i,s_j)}\right)dr_i,s_j\\
&=\log(\prod_i\prod_j p(\mathbf{R}_{i,j}|\mathbf{x}_j;\mathcal{P}))-\sum_i\sum_j\int_{r_i,s_j} q(r_i,s_j)\log\left(\frac{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}{q(r_i,s_j)}\right)dr_i,s_j\\
&=\sum_i\sum_j\log(p(\mathbf{R}_{i,j}|\mathbf{x}_j;\mathcal{P}))-\sum_i\sum_j\int_{r_i,s_j} q(r_i,s_j)\log\left(\frac{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}{q(r_i,s_j)}\right)dr_i,s_j\\
&=\sum_i\sum_j\int_{r_i,s_j}\left[q(r_i,s_j)\log\left(p(\mathbf{R}_{i,j}|\mathbf{x}_j;\mathcal{P})\right)-q(r_i,s_j)\log\left(\frac{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}{q(r_i,s_j)}\right)\right]dr_i,s_j\\
&=\sum_i\sum_j\int_{r_i,s_j} q(r_i,s_j)\log\left(\frac{p(\mathbf{R}_{i,j}|\mathbf{x}_j;\mathcal{P})q(r_i,s_j)}{p(\mathbf{R}_{i,j},r_i,s_j|\mathbf{x}_j;\mathcal{P})}\right)dr_i,s_j\\
&=\sum_i\sum_j\int_{r_i,s_j} q(r_i,s_j)\log\left(\frac{p(\mathbf{R}_{i,j}|\mathbf{x}_j;\mathcal{P})q(r_i,s_j)}{p(r_i,s_j|\mathbf{R}_{i,j},\mathbf{x}_j;\mathcal{P})p(\mathbf{R}_{i,j}|\mathbf{x}_j;\mathcal{P}))}\right)dr_i,s_j\\
&=\sum_i\sum_j\int_{r_i,s_j} q(r_i,s_j)\log\left(\frac{q(r_i,s_j)}{p(r_i,s_j|\mathbf{R}_{i,j},\mathbf{x}_j;\mathcal{P}))}\right)dr_i,s_j\\
&=\sum_i\sum_j KL(q||p(r_i,s_j|\mathbf{R}_{i,j},\mathbf{x}_j;\mathcal{P})).
\end{aligned}\tag{23}$$

The gap Δ refers to the first term in Equation 7, which we need to minimize in the E-step. However, the closed-form EM updates do not work in the case of discrete-continuous variables. We develop a mean field variational inference, following the idea of approximating the posterior distribution of latent variables $p(r_i,s_j|\mathbf{R}_{i,j},\mathbf{x}_j;\mathcal{P})$ with the variational distribution $q(\mathbf{r},\mathbf{s})=\prod_i\prod_j q(r_i,s_j)$.

In the mean-field approach, we assume that

$$q(\mathbf{r},\mathbf{s})=q(\mathbf{r})q(\mathbf{s})=\prod_i q(r_i)\prod_j q(s_i),\tag{24}$$

where $q(r_i)=Beta(\alpha_i,\beta_i)$ (i.e., Equation 9) and $q(s_j)=Ber(\theta_j)$ (i.e., Equation 10).

Next, we will derive the update rule of $q(\mathbf{s})$ and $q(\mathbf{r})$ corresponding to Theorem 2.1 and Theorem 2.2

A.2 Proof of Theorem 2.1

For $q(\mathbf{s})$, we have

$$q(\mathbf{s})\propto\exp\{\mathbb{E}_{q(r_i)}[\log(p(\mathbf{r},\mathbf{s},\mathbf{R},\mathbf{X};\mathcal{P}))]\}.\tag{25}$$

For the j-th client, there is a set of rounds $\mathcal{I}^j$ involving the j-th client. In other words, $\mathbf{R}_{i,j}$ is not blank for $i \in \mathcal{I}^j$. Equation 25 is formulated as

$$\begin{aligned} q(s_j) &\propto \exp\{\mathbb{E}_{q(r_i)}[\log(\prod_{i\in\mathcal{I}^j} p(r_i, s_j, \mathbf{R}_{i,j}, \mathbf{x}_j; \mathcal{P}))]\} \\ &\propto \exp\{\mathbb{E}_{q(r_i)}[\sum_{i\in\mathcal{I}^j} \log(p(r_i, s_j, \mathbf{R}_{i,j}, \mathbf{x}_j; \mathcal{P}))]\} \\ &\propto \exp\{\sum_{i\in\mathcal{I}^j} \mathbb{E}_{q(r_i)}[\log(p(r_i, s_j, \mathbf{R}_{i,j}, \mathbf{x}_j; \mathcal{P}))]\} \\ &\propto \prod_{i\in\mathcal{I}^j} \exp\{\mathbb{E}_{q(r_i)}[\log(p(r_i, s_j, \mathbf{R}_{i,j}, \mathbf{x}_j; \mathcal{P}))]\}. \end{aligned} \tag{26}$$

After applying the chain rule on $p(r_i, s_j, \mathbf{R}_{i,j}, \mathbf{x}_j; \mathcal{P})$, we can get

$$\begin{aligned} p(r_i, s_j, \mathbf{R}_{i,j}, \mathbf{x}_j; \mathcal{P}) &= p(r_i|\mathbf{x}_j; \mathcal{P}) \times p(s_j|r_i, \mathbf{x}_j; \mathcal{P}) \times p(\mathbf{R}_{i,j}|s_j, r_i, \mathbf{x}_j; \mathcal{P}) \\ &= p(r_i) \times p(s_j|\mathbf{x}_j; \mathcal{P}) \times p(\mathbf{R}_{i,j}|r_i, s_j). \end{aligned} \tag{27}$$

This is because s_j only depends on $\mathbf{x}_j$ and $\mathcal{P}$, and $\mathbf{R}_{i,j}$ does not depend on $\mathbf{x}_j$ and $\mathcal{P}$ given s_j and r_i.

Substituting Equation 27 into Equation 26, we have

$$\begin{aligned} q(s_j) &\propto \prod_{i\in\mathcal{I}^j} \exp\{\mathbb{E}_{q(r_i)}[\log(p(r_i) \times p(s_j|\mathbf{x}_j; \mathcal{P}) \times p(\mathbf{R}_{i,j}|r_i, s_j))]\} \\ &\propto \prod_{i\in\mathcal{I}^j} \exp\{\mathbb{E}_{q(r_i)}[\log(p(r_i)] + \mathbb{E}_{q(r_i)}[\log(p(s_j|\mathbf{x}_j; \mathcal{P}))] + \mathbb{E}_{q(r_i)}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\}. \end{aligned} \tag{28}$$

We remove the irrelevant term related to $q(s_j)$, i.e., $\mathbb{E}_{q(r_i)}[\log(p(r_i)]$, then we get

$$\begin{aligned} q(s_j) &\propto \prod_{i\in\mathcal{I}^j} \exp\{\mathbb{E}_{q(r_i)}[\log(p(s_j|\mathbf{x}_j; \mathcal{P}))] + \mathbb{E}_{q(r_i)}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\} \\ &\propto \prod_{i\in\mathcal{I}^j} \exp\{\mathbb{E}_{q(r_i)}[\log(p(s_j|\mathbf{x}_j; \mathcal{P}))]\} \times \exp\{\mathbb{E}_{q(r_i)}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\}. \end{aligned} \tag{29}$$

Since $\log(p(s_j|\mathbf{x}_j; \mathcal{P}))$ dose not contain the variable $q(r_i)$, we have

$$\exp\{\mathbb{E}_{q(r_i)}[\log(p(s_j|\mathbf{x}_j; \mathcal{P}))]\} = \exp\{\log(p(s_j|\mathbf{x}_j; \mathcal{P}))\} = p(s_j|\mathbf{x}_j; \mathcal{P}). \tag{30}$$

Substituting Equation 30 into Equation 29, we have

$$\begin{aligned} q(s_j) &\propto \prod_{i\in\mathcal{I}^j} p(s_j|\mathbf{x}_j; \mathcal{P}) \times \exp\{\mathbb{E}_{q(r_i)}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\} \\ &\propto p(s_j|\mathbf{x}_j; \mathcal{P}) \prod_{i\in\mathcal{I}^j} \exp\{\mathbb{E}_{q(r_i)}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\}. \end{aligned} \tag{31}$$

Equation 31 is equivalent to Equation 11 of Section 2.2 when we define $\mathbb{E}_{q(r_i)}[\log(\cdot)] = g_{q(r_i)}(\cdot)$.

Based on Equation 31, we now derive the update rule of $q(s_j)$ given the variational parameters θ_j, α_i, and β_i from last iteration. We first show the proof for $s_j = 1$; the proof for $s_j = 0$ follows similarly.

Equations 2 and 3 of Section 2.1 parameterize the latent variable s_j with the utility θ_j: the selection s_j of j-th client follows a Bernoulli distribution parameterized by client utility θ_j that is the output of a machine learning model $f^{\mathcal{W}_d}(\cdot)$ with input topmost layer $\mathbf{x}_j$ in the j-th local model. We have

$$p(s_j = 1|\mathbf{x}_j; \mathcal{P}) = \theta_j. \tag{32}$$

In Equation 5, we connect the round informativeness r_i and client selection s_j under the assumption that rounds with higher informativeness have more entries $R_{i,j}$ satisfying actual client utility, which can be formulated to

$$p(\mathbf{R}_{i,j}|r_i, s_j) = {r_i}^{(1-|s_j-\mathbf{R}_{i,j}|)} \times (1-r_i)^{|s_j-\mathbf{R}_{i,j}|}. \tag{33}$$

In the case when $s_j = 1$, Equation 33 is equivalent to

$$p(\mathbf{R}_{i,j}|r_i, s_j) = \begin{cases} 1-r_i & (\mathbf{R}_{i,j}=0) \\ r_i & (\mathbf{R}_{i,j}=1). \end{cases} \tag{34}$$

After substituting the probabilities $p(s_j|\mathbf{x}_j;\mathcal{P})$ and $p(\mathbf{R}_{i,j}|r_i,s_j)$ into Equation 31, we get

$$q(s_j=1) \propto \begin{cases} \theta_j \prod_{i\in\mathcal{I}^j} \exp\{g_{q(r_i)}(1-r_i)\} & (\mathbf{R}_{i,j}=0) \\ \theta_j \prod_{i\in\mathcal{I}^j} \exp\{g_{q(r_i)} r_i\} & (\mathbf{R}_{i,j}=1). \end{cases} \tag{35}$$

By computing the geometric mean of the beta distribution, we can evaluate the expectations $g_x(\cdot)$ as follows:

$$g_{q(r_i)}(1-r_i) = \Psi(\beta_i) - \Psi(\alpha_i+\beta_i), \tag{36}$$

$$g_{q(r_i)} r_i = \Psi(\alpha_i) - \Psi(\alpha_i+\beta_i). \tag{37}$$

Substituting Equations 36 and 37 into Equation 35, we can obtain the update rule of $q(s_j=1)$, as given in Equation 13 of Section 2.2 as well as shown below:

$$q(s_j=1) \propto \begin{cases} \theta_j \prod_{i\in\mathcal{I}^j} \exp\{\Psi(\beta_i) - \Psi(\alpha_i+\beta_i)\} & (\mathbf{R}_{i,j}=0) \\ \theta_j \prod_{i\in\mathcal{I}^j} \exp\{\Psi(\alpha_i) - \Psi(\alpha_i+\beta_i)\} & (\mathbf{R}_{i,j}=1). \end{cases}$$

Similarly, for $s_j = 0$, we have

$$p(s_j=0|\mathbf{x}_j;\mathcal{P}) = 1-\theta_j, \tag{38}$$

and

$$p(\mathbf{R}_{i,j}|r_i,s_j) = \begin{cases} r_i & (\mathbf{R}_{i,j}=0) \\ 1-r_i & (\mathbf{R}_{i,j}=1). \end{cases} \tag{39}$$

Equation 31 is then equivalent to

$$q(s_j=0) \propto \begin{cases} (1-\theta_j) \prod_{i\in\mathcal{I}^j} \exp\{g_{q(r_i)} r_i\} & (\mathbf{R}_{i,j}=0) \\ (1-\theta_j) \prod_{i\in\mathcal{I}^j} \exp\{g_{q(r_i)}(1-r_i)\} & (\mathbf{R}_{i,j}=1). \end{cases} \tag{40}$$

Again, substituting Equation 36 and 37 into Equation 40, we can obtain the update rule of $q(s_j=0)$ as Equation 14 of Section 2.2:

$$q(s_j=0) \propto \begin{cases} (1-\theta_j) \prod_{i\in\mathcal{I}^j} \exp\{\Psi(\alpha_i) - \Psi(\alpha_i+\beta_i)\} & (\mathbf{R}_{i,j}=0) \\ (1-\theta_j) \prod_{i\in\mathcal{I}^j} \exp\{\Psi(\beta_i) - \Psi(\alpha_i+\beta_i)\} & (\mathbf{R}_{i,j}=1). \end{cases}$$

We now conclude the proof of Theorem 2.1.

A.3 Proof of Theorem 2.2

For $q(\mathbf{r})$, we have

$$q(\mathbf{r}) \propto \exp\{\mathbb{E}_{q(s_j)}[\log(p(\mathbf{r},\mathbf{s},\mathbf{R},\mathbf{X};\mathcal{P}))]\}. \tag{41}$$

In the i-th round, there is a set of participating clients $\mathcal{J}^i$ corresponding to entries in the i-th row of matrix $\mathbf{R}$. Equation 41 is formulated as

$$q(r_i) \propto \exp\{\mathbb{E}_{q(s_j)}[\log(\prod_{j\in\mathcal{J}^i} p(r_i, s_j, \mathbf{R}_{i,j}, \mathbf{x}_j; \mathcal{P}))]\}. \tag{42}$$

Following the transformation of $q(s_j)$, we can also get a simplified version of Equation 42, similar to Equation 31:

$$q(r_i) \propto p(r_i) \prod_{j \in \mathcal{J}^i} \exp\{\mathbb{E}_{q(s_j)}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\} \tag{43}$$

Since $q(s_j) = Ber(\theta_j)$, we have

$$\begin{aligned}
\mathbb{E}_{q(s_j)}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))] &= \sum_{s_j} q(s_j) \log(p(\mathbf{R}_{i,j}|r_i, s_j)) \\
&= q(s_j = 0) \times \log(p(\mathbf{R}_{i,j}|r_i, s_j)) + q(s_j = 1) \times \log(p(\mathbf{R}_{i,j}|r_i, s_j)) \\
&= (1 - \theta_j) \times \log(p(\mathbf{R}_{i,j}|r_i, s_j)) + \theta_j \times \log(p(\mathbf{R}_{i,j}|r_i, s_j)) \\
&\stackrel{def}{=} \mathbb{E}_{\theta_j}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))].
\end{aligned} \tag{44}$$

Therefore, Equation 43 is finally transformed into

$$q(r_i) \propto p(r_i) \prod_{j \in \mathcal{J}^i} \exp\{\mathbb{E}_{\theta_j}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\}. \tag{45}$$

Equation 45 is equivalent to Equation 12 of Section 2.2 when we define $\mathbb{E}_{\theta_j}[log(\cdot)] = g_{\theta'}(\cdot)$.

Based on Equation 45, we now derive the update rule of $q(r_i)$ given the variational parameters θ_j, α_i, and β_i from last iteration.

We replace the probability $p(r_i)$ in Equation 45 by the Beta distribution with parameters α_i and β_i from the previous iteration:

$$q(r_i) \propto Beta(\alpha_i, \beta_i) \prod_{j \in \mathcal{J}^i} \exp\{\mathbb{E}_{\theta_j}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\}. \tag{46}$$

According to Equation 33, the $\exp\{\mathbb{E}_{\theta_j}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\}$ in Equation 46 can be reformulated to

$$\begin{aligned}
&\exp\{\mathbb{E}_{\theta_j}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\} \\
&= \exp\{\mathbb{E}_{\theta_j}[\log({r_i}^{(1-|s_j - \mathbf{R}_{i,j}|)} \times (1 - r_i)^{|s_j - \mathbf{R}_{i,j}|})]\} \\
&= \exp\{(1 - \theta_j) \times \log({r_i}^{(1-|\mathbf{R}_{i,j}|)} \times (1 - r_i)^{|\mathbf{R}_{i,j}|}) + \theta_j \times \log({r_i}^{(1-|1-\mathbf{R}_{i,j}|)} \times (1 - r_i)^{|1-\mathbf{R}_{i,j}|})\} \\
&= \exp\{(1 - \theta_j) \times \log({r_i}^{(1-\mathbf{R}_{i,j})} \times (1 - r_i)^{\mathbf{R}_{i,j}}) + \theta_j \times \log({r_i}^{(\mathbf{R}_{i,j})} \times (1 - r_i)^{(1-\mathbf{R}_{i,j})})\} \\
&= \exp\{(1 - \theta_j) \times \log({r_i}^{(1-\mathbf{R}_{i,j})} \times (1 - r_i)^{\mathbf{R}_{i,j}})\} \times \exp\{\theta_j \times \log({r_i}^{(\mathbf{R}_{i,j})} \times (1 - r_i)^{(1-\mathbf{R}_{i,j})})\} \\
&= ({r_i}^{(1-\mathbf{R}_{i,j})} \times (1 - r_i)^{\mathbf{R}_{i,j}})^{(1-\theta_j)} \times ({r_i}^{\mathbf{R}_{i,j}} \times (1 - r_i)^{(1-\mathbf{R}_{i,j})})^{\theta_j} \\
&= {r_i}^{(1+2\mathbf{R}_{i,j}\theta_j - \theta_j - \mathbf{R}_{i,j})} (1 - r_i)^{(\mathbf{R}_{i,j} - 2\mathbf{R}_{i,j}\theta_j + \theta_j)}.
\end{aligned} \tag{47}$$

Therefore, the expectation term in Equation 46 can be evaluated as follows:

$$\exp\{\mathbb{E}_{\theta_j}[\log(p(\mathbf{R}_{i,j}|r_i, s_j))]\} = \begin{cases} {r_i}^{(1-\theta_j)}(1 - r_i)^{\theta_j} & (\mathbf{R}_{i,j} = 0) \\ {r_i}^{\theta_j}(1 - r_i)^{(1-\theta_j)} & (\mathbf{R}_{i,j} = 1). \end{cases} \tag{48}$$

In the case when $\mathbf{R}_{i,j} = 1$, Equation 46 is equivalent to :

$$q(r_i) \propto Beta(\alpha_i, \beta_i) \prod_{j \in \mathcal{J}^i} {r_i}^{\theta_j (1-r_i)^{(1-\theta_j)}}. \tag{49}$$

The probability density function of r_i's distribution is given by:

$$Beta(\alpha_i, \beta_i) \propto {r_i}^{(\alpha_i - 1)}(1 - r_i)^{(\beta_i - 1)}. \tag{50}$$

Substituting Equation 50 into Equation 49, we get

$$\begin{aligned} q(r_i) &\propto {r_i}^{(\alpha_i-1)}(1-r_i)^{(\beta_i-1)} \prod_{j\in\mathcal{J}^i} {r_i}^{\theta_j}(1-r_i)^{(1-\theta_j)} \\ &\propto \prod_{j\in\mathcal{J}^i} {r_i}^{(\alpha_i-1)}(1-r_i)^{(\beta_i-1)} {r_i}^{\theta_j}(1-r_i)^{(1-\theta_j)} \\ &\propto \prod_{j\in\mathcal{J}^i} {r_i}^{(\alpha_i+\theta_j-1)}(1-r_i)^{(\beta_i+(1-\theta_j)-1)} \\ &\propto {r_i}^{(\alpha_i+\sum_{j\in\mathcal{J}^i}\theta_j-1)}(1-r_i)^{(\beta_i+\sum_{j\in\mathcal{J}^i}(1-\theta_j)-1)} \\ &\propto Beta\left(\alpha_i+\sum_{j\in\mathcal{J}^i}\theta_j, \beta_i+\sum_{j\in\mathcal{J}^i}(1-\theta_j)\right). \end{aligned} \tag{51}$$

Similarly, for $\mathbf{R}_{i,j}=0$, we have

$$q(r_i) \propto Beta(\alpha_i,\beta_i) \prod_{j\in\mathcal{J}^i} {r_i}^{(1-\theta_j)}(1-r_i)^{\theta_j}. \tag{52}$$

Again, substituting Equation 50 into Equation 52, we complete the proof of Theorem 2.2 as follows:

$$\begin{aligned} q(r_i) &\propto {r_i}^{(\alpha_i-1)}(1-r_i)^{(\beta_i-1)} \prod_{j\in\mathcal{J}^i} {r_i}^{(1-\theta_j)}(1-r_i)^{\theta_j} \\ &\propto \prod_{j\in\mathcal{J}^i} {r_i}^{(\alpha_i-1)}(1-r_i)^{(\beta_i-1)} {r_i}^{(1-\theta_j)}(1-r_i)^{\theta_j} \\ &\propto \prod_{j\in\mathcal{J}^i} {r_i}^{(\alpha_i+(1-\theta_j)-1)}(1-r_i)^{(\beta_i+\theta_j-1)} \\ &\propto {r_i}^{(\alpha_i+\sum_{j\in\mathcal{J}^i}(1-\theta_j)-1)}(1-r_i)^{(\beta_i+\sum_{j\in\mathcal{J}^i}\theta_j-1)} \\ &\propto Beta\left(\alpha_i+\sum_{j\in\mathcal{J}^i}(1-\theta_j), \beta_i+\sum_{j\in\mathcal{J}^i}\theta_j\right). \end{aligned} \tag{53}$$

B Convergence Analysis

B.1 Preliminaries

Before the convergence analysis on FedTrans, we first provide a detailed formulation of the local training and global aggregation in FedTrans with selective client participation.

In the i-th communication round, the cloud server randomly selects a set of participating clients $\mathcal{J}^i$. Each client $j\in\mathcal{J}^i$ performs the stochastic gradient descent (SGD) on the local data instances:

$$\mathcal{L}(f_j^{\mathcal{W}_{\tau,j}}, \xi_\tau^j) = \frac{1}{|\xi_\tau^j|}\sum_{x\in\xi_\tau^j} l(f_j^{\mathcal{W}_{\tau,j}}; x), \tag{54}$$

$$\hat{\mathcal{W}}_{\tau+1,j} = \mathcal{W}_{\tau,j} - \eta_\tau \nabla\mathcal{L}(f_j^{\mathcal{W}_{\tau,j}}, \xi_\tau^j), \tag{55}$$

where τ indexes the local SGD step on the objective $\mathcal{L}(\cdot)$ regarding the loss function $l(\cdot)$ (e.g., cross-entropy loss). ξ_τ^j is a batch of samples randomly selected from local data $\mathcal{D}_j$ at step τ. Client updates local model to $f_j^{\hat{\mathcal{W}}_{\tau+1,j}}$ after one-step gradient descent with the learning rate η_τ. Suppose that client reports the model updates every $E\in\mathbb{Z}^+$ steps, and we have the update rule

$$\mathcal{W}_{\tau+1,j} = \begin{cases} \hat{\mathcal{W}}_{\tau+1,j}, & \tau+1\neq iE \\ \sum_{j\in\mathcal{J}^i} p_j \hat{\mathcal{W}}_{\tau+1,j}, & \tau+1=iE. \end{cases} \tag{56}$$

We define the local updates of client j offloading to the cloud server in round i as

$$\mathcal{W}_{i,j}^{*} \triangleq \hat{\mathcal{W}}_{iE,j}. \tag{57}$$

Under the guidance of client utility estimated by FedTrans, the server aggregates a subset of updates from clients $\hat{\mathcal{J}}^i \subseteq \mathcal{J}^i$ for the global model

$$\bar{\mathcal{W}}_i^{*} \triangleq \sum_{j \in \hat{\mathcal{J}}^i} \hat{p}_j \mathcal{W}_{i,j}^{*} = \sum_{j \in \hat{\mathcal{J}}^i} \hat{p}_j \hat{\mathcal{W}}_{iE,j}. \tag{58}$$

In the following section, we simplify the objective function as $\mathcal{L}_j(\mathcal{W}, \xi) = \mathcal{L}(f_j^{\mathcal{W}}, \xi)$ and gradients as $g_j(\mathcal{W}, \xi) = \nabla \mathcal{L}_j(\mathcal{W}, \xi)$ for a more concise expression. Specifically, $\mathcal{L}_j(\mathcal{W}) = \mathcal{L}(f_j^{\mathcal{W}}, \mathcal{D}_j)$ and $g_j(\mathcal{W}) = \nabla \mathcal{L}_j(\mathcal{W})$.

B.2 Convergence Guarantee

A rigorous convergence analysis of FedTrans is non-trivial since the tendentious selection of clients with high utility compared with random client selection (e.g., FedAvg). Borrowing from the theoretical analysis in (Cho et al., 2022) that considers the effect of biased client participation on convergence, we can provide the convergence guarantee of FedTrans under assumptions

Assumption 1 *$\mathcal{L}_1, \cdots, \mathcal{L}_{|\mathcal{J}|}$ are all L-smooth, i.e., for all $\mathcal{W}$ and $\mathcal{W}'$,*

$$\mathcal{L}_j(\mathcal{W}) \leq \mathcal{L}_j(\mathcal{W}') + \langle \nabla \mathcal{L}_j(\mathcal{W}'), \mathcal{W} - \mathcal{W}' \rangle + \frac{L}{2} \|\mathcal{W} - \mathcal{W}'\|_2^2. \tag{59}$$

Assumption 2 *$\mathcal{L}_{\mathcal{D}_1}, \cdots, \mathcal{L}_{\mathcal{D}_{|\mathcal{J}|}}$ are all μ-strongly convex, i.e., for all $\mathcal{W}$ and $\mathcal{W}'$,*

$$\mathcal{L}_j(\mathcal{W}) \geq \mathcal{L}_j(\mathcal{W}') + \langle \nabla \mathcal{L}_j(\mathcal{W}'), \mathcal{W} - \mathcal{W}' \rangle + \frac{\mu}{2} \|\mathcal{W} - \mathcal{W}'\|_2^2. \tag{60}$$

Assumption 3 *For mini-batch ξ_τ^j uniformly sampled at random from local data of j-th client $\mathcal{D}_j$, the resulting stochastic gradient is unbiased, that is, $\mathbb{E}[g_j(\mathcal{W}_{\tau,j}, \xi_\tau^j)] = g_j(\mathcal{W}_{\tau,j})$. Also, the variance of the stochastic gradient is bounded, i.e., for all $j = 1, \cdots, |\mathcal{J}|$ and $\forall \tau$,*

$$\mathbb{E}[\|g_j(\mathcal{W}_{\tau,j}, \xi_\tau^j) - g_j(\mathcal{W}_{\tau,j})\|^2] \leq \sigma^2. \tag{61}$$

Assumption 4 *The stochastic gradient's expected squared norm is uniformly bounded, i.e., for all $j = 1, \cdots, |\mathcal{J}|$ and $\forall \tau$,*

$$\mathbb{E}[\|g_j(\mathcal{W}_{\tau,j}, \xi_\tau^j)\|^2] \leq G^2. \tag{62}$$

In line with (Cho et al., 2022), we then define two metrics: local-global objective gap and selection skew. The local-global objective gap is formulated as

$$\Gamma \triangleq \sum_{j}^{|\mathcal{J}|} p_j (\mathcal{L}_j(\bar{\mathcal{W}}^{*}) - \mathcal{L}_j(\mathcal{W}_j^{*})), \tag{63}$$

where p_j refers to the fraction of data at j-th client to the overall data volume. The $\bar{\mathcal{W}}^{*}$ and $\mathcal{W}_j^{*}$ are the optimal weight parameters of the global objective $\mathcal{L}(\cdot)$ and local objective $\mathcal{L}_j(\cdot)$ respectively. To be more specific,

$$\mathcal{W}_j^{*} = \arg\min_{\mathcal{W}} \mathcal{L}_j(\mathcal{W}), \tag{64}$$

$$\bar{\mathcal{W}}^{*} = \arg\min_{\mathcal{W}} \mathcal{L}(\mathcal{W}) = \arg\min_{\mathcal{W}} \sum_{j}^{|\mathcal{J}|} p_j \mathcal{L}_j(\mathcal{W}). \tag{65}$$

Further, we denote $\mathcal{L}_j^{*} = \mathcal{L}_j(\mathcal{W}_j^{*})$ and $\mathcal{L}^{*} = \mathcal{L}(\bar{\mathcal{W}}^{*})$.

The selection skew is formulated as

$$\rho(\mathcal{S}(\pi,\mathcal{W}),\mathcal{W}') = \frac{\mathbb{E}_{\mathcal{S}(\pi,\mathcal{W})}[\frac{1}{m}\sum_{j\in\mathcal{S}(\pi,\mathcal{W})}(\mathcal{L}_j(\mathcal{W}')-\mathcal{L}_j^*)]}{\mathcal{L}(\mathcal{W}')-\sum_j^{|\mathcal{J}|}p_j\mathcal{L}_j^*} \tag{66}$$

where $\mathcal{S}(\pi,\mathcal{W})$ is a set of selected m clients given selection strategy π according to weight parameters $\mathcal{W}$, and $\mathcal{W}'$ is the observing point where we evaluate the local objective $\mathcal{L}_j(\mathcal{W}')$ and global objective $\mathcal{L}(\mathcal{W}')$.

We then define two related metrics

$$\bar{\rho} \triangleq \min_{\mathcal{W},\mathcal{W}'} \rho(\mathcal{S}(\pi,\mathcal{W}),\mathcal{W}'), \tag{67}$$

$$\tilde{\rho} \triangleq \max_{\mathcal{W}} \rho(\mathcal{S}(\pi,\mathcal{W}),\mathcal{W}^*). \tag{68}$$

Under the above-mentioned three assumptions, for learning rate $\eta_t = \frac{1}{\mu(t+\gamma)}$ and $\gamma = \frac{4L}{\mu}$, we have the convergence with any selection strategy π after T local steps as

$$\mathbb{E}[\mathcal{L}(\bar{\mathcal{W}}_T^*)]-\mathcal{L}^* \leqslant \frac{1}{T+\Gamma}[\frac{4L(32\tau^2G^2+\sigma^2/m)}{3\mu^2\bar{\rho}}+\frac{8L^2\Gamma}{\mu^2}+\frac{L\gamma\|\bar{\mathcal{W}}_0^*-\bar{\mathcal{W}}^*\|^2}{2}]+\frac{8L\Gamma}{3\mu}(\frac{\tilde{\rho}}{\bar{\rho}}-1) \tag{69}$$

where T is the local SGD interactions, and G is the upper bound of the stochastic gradient's expected squared norm in assumption (3).

Similarly, for a fixed learning rate $\eta \leqslant \min\{\frac{1}{2\mu B},\frac{1}{4L}\}$ where is $B = 1+\frac{3\bar{\rho}}{8}$, we have the convergence with any selection strategy π as

$$\begin{aligned}\mathbb{E}[\mathcal{L}(\bar{\mathcal{W}}_T^*)]-\mathcal{L}^* &\leqslant \frac{4L\eta(32\tau^2G^2+\frac{\sigma^2}{m})+6\bar{\rho}L\Gamma)}{\mu(8+3\bar{\rho})}+\frac{8L\Gamma(\tilde{\rho}-\bar{\rho})}{\mu(8+3\bar{\rho})}\\ &+\frac{L}{\mu}\left(1-\eta\mu(1+\frac{3\bar{\rho}}{8})\right)^T\left(\mathcal{L}(\bar{\mathcal{W}}_0^*)-\mathcal{L}^*-\frac{4\left(\eta(32\tau^2G^2+\frac{\sigma^2}{m}+6\bar{\rho}L\Gamma)+2\Gamma(\tilde{\rho}-\bar{\rho})\right)}{8+3\bar{\rho}}\right).\end{aligned} \tag{70}$$

For a small η, both fixed-learning rate case and decaying-learning rate case have the same upper bound by $\frac{8L\Gamma}{3\mu}(\frac{\tilde{\rho}}{\bar{\rho}}-1)$.

In FedTrans, the selection strategy π_{util} chooses clients $\hat{\mathcal{J}}^i \subseteq \mathcal{J}^i$ from the participating clients based on the estimated utilities. Therefore, in Equation 66, $\mathcal{S}(\pi,\mathcal{W}) = \hat{\mathcal{J}}^i$ and $m = |\hat{\mathcal{J}}^i|, \forall i \in \mathcal{I}$. According to the analysis in (Cho et al., 2022), a larger selection skew ρ results in faster convergence. We provide the lower bound of convergence rate at $\mathcal{O}(\frac{1}{T\bar{\rho}})$ where T denotes the accumulated local SGD steps, since $\bar{\rho}$ calculates the minimum of the selection skew, as shown in Equation 67. In practice, the varying weights parameters of the global model $\bar{\mathcal{W}}_i^*$ and the local model $\mathcal{W}_{i,j}^*$ cause the selection skew $\rho(\mathcal{S}(\pi,\bar{\mathcal{W}}_i^*),\mathcal{W}_{i,j}^*)$ to change with the FL proceeding while maintaining convergence rate of at least of $\bar{\rho}$.

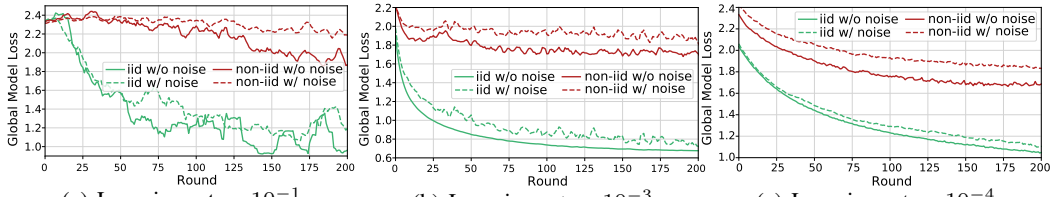
C More Experimental Details

Implementation Details. The parameters of variational utility inference and those for discriminator $f^{\mathcal{W}_d}$ training are empirically set. We adopt $f^{\mathcal{W}_d}$ with Multi-Layer Perception (MLP) having 2 hidden layers of 128 and 64 dimensions respectively. In discriminator training, we select the learning rate as $1e-3$, and we set the priors A and B by sampling from a uniform distribution $\sim [0,10]$ and update them in E-step according to Theorem 2.1 and Theorem 2.2.

Comparison Methods. We implement all the comparison methods in Python and the neural networks with PyTorch, running on an NVIDIA 2080Ti GPU. In local training, local epochs are set to 5 and the learning rate is $1e-2$. We use SGD with momentum factor = 0.9 as the local optimizer.

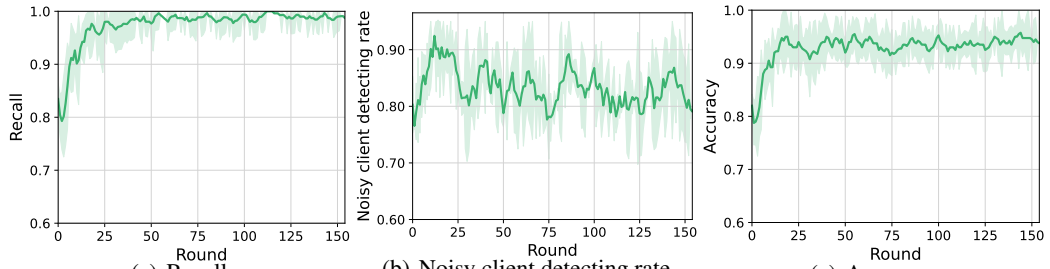
To evaluate the resilience of FedTrans to data noise, we compare it with SOTA baselines. RHFL (Fang & Ye, 2022) considers symmetrically using cross-entropy loss and reverse cross-entropy loss to ameliorate the negative effect of internal local model noise. Robust-FL (Yang et al., 2022b) copes with the noisy federated setting by interchanging additional information called class-wise centroids. FLDebugger (Li et al., 2021) utilizes the 2-norm distance between local weight parameters and global weight parameters to distinguish noisy clients. FedCorr (Xu et al., 2022) calculates the average Local Intrinsic Dimension (LID) of local prediction vectors for each client. The server applies a Gaussian Mixture Model on received LID scores to partition involved clients into two subsets: noisy clients and clean clients. As a client selection method, Oort (Lai et al., 2021) achieves enhanced time-to-accuracy by constructing both statistical utility and system utility for clients. DivFL (Balakrishnan et al., 2022) selects a subset of clients whose weight updates closely mimic the information gained from aggregating updates across all clients, to improve learning efficiency. Note that FLDebugger and FedCorr are sample-wise noisy correction methods where we partially compare two baselines with regard to the identification of noisy clients. We implement the component about data utility in Oort for comparison.

Observation details. Figure 1 illustrates the negative impact of noisy clients under two different local data distributions, emphasizing more severe degradation in global model performance when local data is non-IID across clients. For the completeness of experiments, we also explore the performance of the global model varying with learning rates other than 10^{-2} in Figure 1. Compared with results in Figure 9, under both IID and non-IID settings, the global model achieves the best performance when the learning rate is 10^{-2} as used in Figure 1. Furthermore, we observe a more pronounced degradation of the model performance introduced by data noise in non-IID data compared to IID data across all learning rates ranging from 10^{-1} to 10^{-4}, which necessitates inferring client utility in more realistic non-IID scenarios.



(a) Learning rate = 10^{-1} (b) Learning rate = 10^{-3} (c) Learning rate = 10^{-4}

Figure 9: Global model performance of FL with noisy clients (in Hybrid *(across-)* local noise). We consider two data distributions and three different learning rates.



(a) Recall (b) Noisy client detecting rate (c) Accuracy

Figure 10: The client selection performance in the presence of $\epsilon = 30\%$ noisy clients.

Client selection performance. In the experiments, we adopt a threshold-based selection strategy, excluding the updates of the clients with the utility θ below 0.5 from the global model aggregation. As shown in Figure 10, we evaluate the client selection performance in the setting where $\epsilon = 30\%$ clients corrupted by Hybrid *(across-)* noise. We observe the high recall performance, stabilizing at

approximately 99% after 50 communication rounds. It indicates FedTrans's effective recognition of clean (positive) clients on the server side. The success rate of detecting noisy clients (true negative rate) reaches an average of 85% as the rounds progress. Furthermore, the overall accuracy of client selection according to the utilities estimated by FedTrans is around 95%, given that the clean clients are the majority participants (at approximately 70%) in each communication round.

Correlation between utility and noise rate. As inferring client utility is the main focus of this paper, we investigate the performance of utility estimation by FedTrans across communication rounds. In contrast to evaluating the success rate of detecting noisy clients, our assessment measures the strength of the monotonic relationship between estimated utility and actual local noise. Disregarding the specific selection strategy (e.g., the threshold-based method) according to client utilities, we employ Spearman's rank-order correlation coefficient (PROCC) to quantify the statistical dependence between two variables (i.e., θ_j and local noise rate). As shown in Figure 11, the utility exhibits a negative relationship with local noise, and this relationship becomes more significant along with FL progress. This observation indicates that utility estimation becomes more accurate and aligned with the client noise degree, primarily attributed to the incremental updates on the discriminator $f^{\mathcal{W}_d}$ across communication rounds.

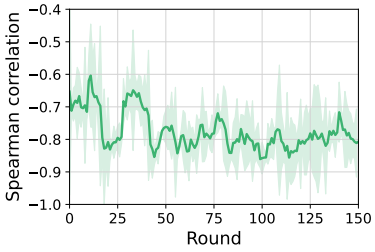


Figure 11: The correlation between estimated client utility and its actual local noise rate varies with the communication rounds.

D TRANSPARENT CLIENT UTILITY ESTIMATION IN FEDERATED LEARNING

Here, we provide the pseudo-code of the overall FL process with FedTrans as a general algorithmic framework (see Algorithm 2). In each round, local models in participating clients are first updated as in standard FL settings (**rows 3-6**). The server then updates the round-reputation matrix and infers client utility (**row 8-11**) by calling the variational utility inference algorithm (see Algorithm 1). After that, it can perform an arbitrary client selection strategy guided by estimated client utility (**row 12-14**). It finally obtains the global model by aggregating the weight parameters from selected clients (**row 15**). FedTrans, as a module that does not require any additional information from the client, can be coupled to any existing aggregation and local training schemes.

Algorithm 2 Federated Learning with FedTrans

1: **Require:** A set of clients with self-contained data: $\mathcal{C}$; Server auxiliary dataset: $\mathcal{D}_a$; Client selection rate: γ.
2: **for** each round $i = 1, 2, \cdots, N$ **do**
3: $\quad\mathcal{J}^i \leftarrow$ randomly select $max(|\mathcal{C}| \times \gamma, 1)$ clients from $\mathcal{C}$
4: $\quad$**for** $j \in \mathcal{J}^i$ **in parallel do** ▷ Local Training
5: $\quad\quad\mathcal{W}^*_{i,j} \leftarrow$*ClientUpdate* $(C_j, \bar{\mathcal{W}}^*_{i-1})$
6: $\quad$**end for**
7: $\quad$**Server Executes FedTrans:**
8: $\quad\mathbf{R}_i \leftarrow$*MatrixUpdate* $(\{\mathcal{W}^*_{i,j}\}_{j\in\mathcal{J}^i}, \mathbf{R}_{i-1}, \mathcal{D}_a)$
9: $\quad\mathcal{W}_d \leftarrow$*VariationalInference* $(\{\mathcal{W}^*_{i,j}\}_{j\in\mathcal{J}^i}, \bar{\mathcal{W}}^*_{i-1}, \mathbf{R}_i)$ ▷ Utility Estimation
10: $\quad\{\mathbf{x}_j\}_{j\in\mathcal{J}^i} \leftarrow$ top-layer of $\{\mathcal{W}^*_{i,j}\}_{j\in\mathcal{J}^i}$
11: $\quad\{\theta_j\}_{j\in\mathcal{J}^i} \leftarrow f^{\mathcal{W}_d}(\{\mathbf{x}_j\}_{j\in\mathcal{J}^i})$
12: $\quad$**for** $j \in \mathcal{J}^i$ **do**
13: $\quad\quad s_j \leftarrow$ client selection guided by θ_j ▷ Client Selection
14: $\quad$**end for**
15: $\quad\bar{\mathcal{W}}^*_i \leftarrow$*Aggregation* $(\{\mathcal{W}^*_{i,j}\}_{j\in\mathcal{J}^i}, \{s_j\}_{j\in\mathcal{J}^i})$ ▷ Server Aggregation
16: **end for**

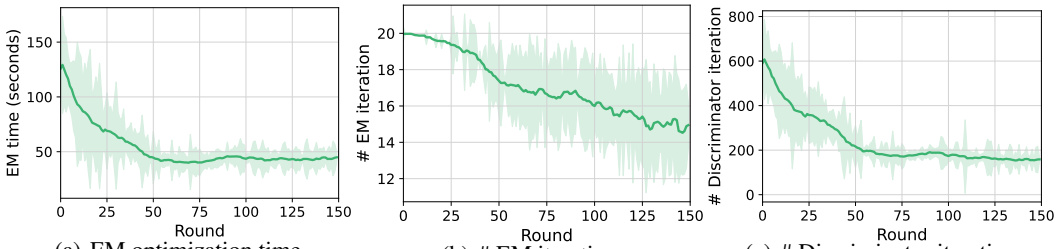
E TIME CONSUMPTION

Although the FedTrans is running on the server with relatively rich computing resources, this inevitably incurs extra training time overheads. We present a detailed time consumption analysis under noisy clients (i.e., Hybrid (cross-)) on the CIFAR10 dataset with distribution Dir(0.5), as shown in Table 2. Note that FedCorr Xu et al. (2022) requires involving all clients to estimate a set of noisy clients before Federated Learning (FL) training starts. Therefore, we do not investigate the per-round time consumption of FedCorr in this context.

Table 2: Time consumption per round or when achieving the target accuracy for each method.

	CIFAR-10, MobileNetV2, *Dir*(0.5)						
	FedAvg	**FLDebugger**	**Oort**	**Robust-FL**	**RHFL**	**DivFL**	**FedTrans**
Per round (seconds)	113 ± 3	115 ± 2	114 ± 3	129 ± 8	115 ± 3	114 ± 3	173 ± 5
Target accuracy (minutes)	118.7 ± 3.2	280.8 ± 4.9	385.8 ± 10.2	60.9 ± 3.3	63.3 ± 1.7	49.4 ± 0.9	21.7 ± 0.4

In Table 2, we provide the maximum time consumption for a single round of FedTrans and other baselines across 300 communication rounds of Federated Learning (FL) communication. Additionally, it includes cumulative time consumption when reaching the target accuracy of 63% (i.e., the maximum accuracy achievable by all methods). The results reveal that, although FedTrans requires more time for each round compared to the baselines, the overall time consumption to reach the target accuracy with our proposed FedTrans is the shortest, showing an impressive speedup of more than 56%. This is primarily because FedTrans requires fewer rounds to attain the targeted accuracy.



(a) EM optimization time (b) # EM iteration (c) # Discriminator iteration

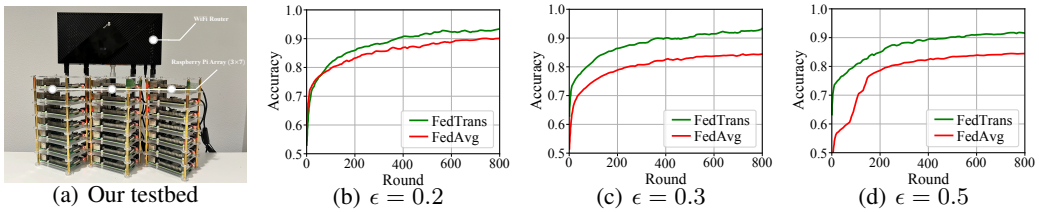
Figure 12: The varying optimization overhead in FedTrans across communication round. From left to right: total optimization time, total EM iterations, and the Discriminator iteration required in variational inference.

Furthermore, we also explore the optimization overhead of the variational EM algorithm that changes as FL proceeds. Figure 12 shows the EM optimization time, EM iterations, and Discriminator (i.e., M-step) iterations varying with communication rounds. The overall optimization time significantly decreases as FL progresses, attributed to the diminishing trend observed in both EM iterations and discriminator iterations. Moreover, the time cost of training the discriminator takes a larger proportion of the total optimization time. As a result, the overall optimization time across communication rounds exhibits a similar trend as the discriminator iterations.

F EVALUATION ON PRACTICAL RESOURCE-CONSTRAINED TESTBED

We also build a testbed consisting of 21 Raspberry Pi 4B embedded devices, as shown in Figure 13(a), to evaluate the performance of FedTrans in practical resource-constrained scenarios.

HAR Dataset. We conduct the experiments on the Human Activity Recognition (HAR) dataset collected from onboard inertial sensors (accelerometer and gyroscope) for distinguishing six activities of daily living: *walking, walking upstairs/downstairs, sitting, standing, and laying.* HAR dataset shows inherent suitability for the FL scenarios since the data is naturally split by different participants. To be more specific, HAR contains data collected from 30 different users, and we take 21 users as the training data and augment the number of clients from 21 to 42 by assigning the data of each user to two clients. We run the FedTrans on a testbed and each Raspberry Pi simulates



(a) Our testbed (b) $\epsilon = 0.2$ (c) $\epsilon = 0.3$ (d) $\epsilon = 0.5$

Figure 13: Evaluation of FedTrans on the HAR dataset implemented on resource-constrained devices, across three settings of client flip rate ($\epsilon_c = 0.2, 0.3, 0.5$). **Our built testbed: 21 Raspberry Pi 4B embedded devices connect to an FL server wirelessly via a TP-Link WiFi Router.**

two clients and connects to the laptop wirelessly through a TP-Link WiFi Router. Here, we employ a shallow CNN with two convolutional layers for the HAR task and only consider the *Random Flipping* noise with three different proportions of corrupted clients ranging from 30% to 50%.

Results in Figure 13 show FedTrans can already outperform FedAvg by a margin from the beginning. We notice that the gap becomes larger when the number of corrupted clients increases. While FedAvg performance increases when the training proceeds to later rounds, it remains lower than FedTrans. This shows the advantage of selecting clean clients even though the number of much fewer than the clients selected by FedAvg.

Energy-to-accuracy. We report the energy consumption (measured by the power meter) when the global model achieves the same performance (i.e., the best accuracy of FedAvg) utilizing the CIFAR10 dataset on our testbed shown in Figure 9(a). FedTrans is more energy-efficient even with the extra overhead at the server: it reduces the energy consumption by up to 66.1%. This is mainly due to the fewer training rounds required when FL is training with FedTrans: it requires only 52.2%, 79.2%, and 74.1% rounds to reach FedAvg's best performance for three client noise ratios, respectively.

G PERFORMANCE ON OTHER DATA DISTRIBUTIONS

IID setting. We conduct experiments under the IID setting using the CIFAR10 and FMNIST dataset, evaluating the performance of baselines in mixed noise scenarios, The results are presented in Table 3 and Table 4. The data reported in the tables is based on five trials.

Table 3: Global model accuracy under six types of noise configurations using CIFAR10 dataset. Distribution of the local data is followed by IID setting, and $\epsilon = 30\%$ of participating 100 clients are corrupted.

CIFAR-10, MobileNetV2, *IID*						
	Hybrid (*across-*)	**Hybrid** (*intra-*)	**Label** (*across-*)	**Label** (*intra-*)	**Image** (*across-*)	**Image** (*intra-*)
FedAvg (McMahan et al., 2017)	82.2% ± 0.1%	80.2% ± 0.2%	81.6% ± 0.1%	81.3% ± 0.0%	82.5% ± 0.0%	81.6% ± 0.1%
FLDebugger (Li et al., 2021)	82.1% ± 0.0%	81.5% ± 0.3%	82.0% ± 0.2%	81.4% ± 0.4%	81.9% ± 0.1%	82.2% ± 0.2%
Oort (Lai et al., 2021)	66.7% ± 0.4%	63.4% ± 0.5%	64.7% ± 0.7%	64.8% ± 0.0%	70.2% ± 0.3%	69.2% ± 0.5%
Robust-FL (Yang et al., 2022b)	78.4% ± 0.0%	77.5% ± 0.6%	77.9% ± 0.1%	78.2% ± 0.4%	78.0% ± 0.0%	76.9% ± 0.2%
RHFL (Fang & Ye, 2022)	83.1% ± 0.2%	82.3% ± 0.1%	83.4% ± 0.1%	83.4% ± 0.2%	82.9% ± 0.5%	82.1% ± 0.1%
DivFL (Balakrishnan et al., 2022)	79.4% ± 0.1%	78.4% ± 0.2%	77.5% ± 0.2%	77.6% ± 0.0%	80.1% ± 0.0%	78.7% ± 0.2%
FedCorr (Xu et al., 2022)	79.1% ± 0.0%	79.6% ± 0.1%	78.7% ± 0.1%	78.3% ± 0.1%	79.8% ± 0.3%	78.9% ± 0.2%
FedTrans	83.9% ± 0.3%	83.8% ± 0.1%	84.2% ± 0.2%	84.0% ± 0.0%	84.1% ± 0.1%	83.8% ± 0.2%

Table 4: Global model accuracy under IID setting with varying ratios of the corrupted clients using FMNIST dataset.

The Ratio of Corrupted Clients					
	30%	**40%**	**50%**	**60%**	**70%**
FedAvg w/o noise (McMahan et al., 2017)	91.2% ± 0.0%	91.2% ± 0.0%	91.2% ± 0.0%	91.2% ± 0.0%	91.2% ± 0.0%
FedAvg w/ noise (McMahan et al., 2017)	90.4% ± 0.2%	90.3% ± 0.1%	90.0% ± 0.1%	89.7% ± 0.1%	87.6% ± 0.0%
FedTrans	90.5% ± 0.0%	90.9% ± 0.1%	90.3% ± 0.1%	90.4% ± 0.7%	90.2% ± 0.0%

Results in these two tables demonstrate that the negative of unreliable clients could be exacerbated by the complexity of the task. Even with mixed noise, a significant accuracy drop only becomes discernible when the ratio of corrupted clients surpasses 60% in the relatively straightforward task (i.e., FMNIST dataset). FedAvg suffers from significant performance degradation under mixed noise on the complex task (i.e., CIFAR10 dataset), while FedTrans outperforms other baselines in effectively mitigating the effects of such complex noise.

H2C setting. We also explored a more severe non-IID experimental setting where each client holds only two classes of data, denoted as H2C. Table 5 reports the results of the compared methods in different noise configurations under 30% corrupted clients.

Table 5: Global model accuracy under six types of noise configurations. Distribution of the local data is followed by H2C setting, and $\epsilon = 30\%$ of participating 100 clients are corrupted.

CIFAR-10, MobileNetV2, *H2C*						
	Hybrid (*across-*)	**Hybrid** (*intra-*)	**Label** (*across-*)	**Label** (*intra-*)	**Image** (*across-*)	**Image** (*intra-*)
FedAvg (McMahan et al., 2017)	43.6% ± 0.1%	37.9% ± 0.3%	43.1% ± 0.3%	40.3% ± 0.5%	43.5% ± 0.2%	44.9% ± 0.5%
FLDebugger (Li et al., 2021)	37.7% ± 0.1%	40.7% ± 0.7%	37.1% ± 0.7%	37.7% ± 0.2%	38.2% ± 0.9%	40.2% ± 1.0%
Oort (Lai et al., 2021)	33.2% ± 0.2%	18.7% ± 0.7%	34.9% ± 0.4%	28.3% ± 0.7%	44.6% ± 0.8%	41.1% ± 0.1%
Robust-FL (Yang et al., 2022b)	44.4% ± 0.4%	36.1% ± 0.3%	44.3% ± 0.3%	43.1% ± 0.4%	46.3% ± 0.5%	43.9% ± 1.0%
RHFL (Fang & Ye, 2022)	43.8% ± 0.5%	36.0% ± 0.8%	46.6% ± 1.1%	43.7% ± 0.8%	44.9% ± 0.8%	43.2% ± 0.3%
DivFL (Balakrishnan et al., 2022)	45.1% ± 0.9%	40.4% ± 0.4%	42.64% ± 0.2%	41.3% ± 0.0%	44.1% ± 0.1%	45.4% ± 0.2%
FedCorr (Xu et al., 2022)	48.0% ± 0.5%	32.1% ± 0.1%	47.7% ± 0.3%	45.0% ± 0.3%	40.1% ± 1.1%	40.4% ± 0.3%
FedTrans	50.0% ± 0.7%	48.4% ± 0.6%	45.3% ± 0.9%	48.8% ± 0.7%	46.6% ± 0.6%	46.4% ± 0.8%
FMNIST, LeNet-5, *H2C*						
FedAvg (McMahan et al., 2017)	76.1% ± 0.1%	75.7% ± 0.1%	76.0% ± 0.5%	75.7% ± 0.6%	76.7% ± 0.2%	76.9% ± 0.2%
FLDebugger (Li et al., 2021)	68.2% ± 0.2%	69.3% ± 0.2%	72.1% ± 0.1%	69.0% ± 0.3%	57.9% ± 0.2%	56.8% ± 0.2%
Oort (Lai et al., 2021)	72.3% ± 0.3%	48.0% ± 0.7%	66.2% ± 0.9%	45.7% ± 0.8%	78.9% ± 0.1%	78.1% ± 0.8%
Robust-FL (Yang et al., 2022b)	79.8% ± 0.2%	79.9% ± 0.0%	79.8% ± 0.8%	78.3% ± 0.2%	79.3% ± 0.3%	80.3% ± 0.4%
RHFL (Fang & Ye, 2022)	80.7% ± 0.1%	81.2% ± 0.2%	80.6% ± 0.2%	80.0% ± 0.3%	80.8% ± 0.2%	80.9% ± 0.2%
DivFL (Balakrishnan et al., 2022)	76.8% ± 0.7%	76.4% ± 0.9%	75.1% ± 0.4%	75.2% ± 0.2%	76.0% ± 0.3%	77.3% ± 0.5%
FedCorr (Xu et al., 2022)	82.1% ± 0.1%	81.7% ± 0.1%	82.2% ± 0.4%	81.3% ± 0.1%	79.3% ± 0.2%	79.3% ± 0.2%
FedTrans	82.4% ± 0.2%	84.2% ± 0.6%	84.3% ± 0.1%	84.7% ± 0.3%	83.5% ± 0.2%	83.2% ± 0.2%