The Appendix part is organized as follows:

- All related work are provided in Appendix A.
- Additional details of prior work of BBSE and MLLS are in Appendix B.
- Mathematical proof for label shifts with multiple nodes and IW-ERM is given in Appendix C.
- General algorithmic description is in Appendix D.
- Proof of Theorem 5.1 is in Appendix E.
- Proof of Theorem 5.2 and Convergence-Communication-Privacy guarantees for IW-ERM in Equation (IW-ERM) are provided in Appendix F.
- Complexity analysis is in Appendix G.
- Mathematical notations are summarized in Appendix H.
- Limitations are discussed in Appendix I.
- Additional experiments and experimental details are provided in Appendix J.

A RELATED WORK

In the context of distributed learning with label shifts, importance ratio estimation is tackled either by solving a linear system as in (Lipton et al., 2018; Azizzadenesheli et al., 2019) or by minimizing distribution divergence as in (Garg et al., 2020). In this section, we overview complete related work.

Federated learning (FL). Much of the current research in FL predominantly centers around the minimization of empirical risk, operating under the assumption that each node maintains the same training/test data distribution (Li et al., 2020a; Kairouz et al., 2021; Wang et al., 2021b). Prominent methods in FL (Kairouz et al., 2021; Li et al., 2020a; Wang et al., 2021b) include FedAvg (McMahan et al., 2017), FedBN (Li et al., 2021b), FedProx (Li et al., 2020b) and SCAFFOLD (Karimireddy et al., 2020a). FedAvg and its variants such as (Huang et al., 2021; Karimireddy et al., 2020b) have been the subject of thorough investigation in optimization literature, exploring facets such as communication efficiency, node participation, and privacy assurance (Ramezani-Kebrya et al., 2023).Subsequent work, such as the study by de Luca et al. (2022), explores Federated Domain Generalization and introduces data augmentation to the training. This model aims to generalize to both in-domain datasets from participating nodes and an out-of-domain dataset from a non-participating node. Additionally, Gupta et al. (2022) introduces FL Games, a game-theoretic framework designed to learn causal features that remain invariant across nodes. This is achieved by employing ensembles over nodes' historical actions and enhancing local computation, under the assumption of consistent training/test data distribution across nodes. The existing strategies to address statistical heterogeneity across nodes during training primarily rely on heuristic-based personalization methods, which currently lack theoretical backing in statistical learning (Smith et al., 2017; Khodak et al., 2019; Li et al., 2021a). In contrast, we aim to minimize overall test error amid both intra-node and inter-node distribution shifts, a situation frequently observed in real-world scenarios. Techniques ensuring communication efficiency, robustness, and secure aggregations serve as complementary.

Importance ratio estimation Classical Empirical Risk Minimization (ERM) seeks to minimize the expected loss over the training distribution using finite samples. When faced with distribution shifts, the goal shifts to minimizing the expected loss over the target distribution, leading to the development of Importance-Weighted Empirical Risk Minimization (IW-ERM)(Shimodaira, 2000; Sugiyama et al., 2006; Byrd & C. Lipton, 2019; Fang et al., 2020). Shimodaira (2000) established that the IW-ERM estimator is asymptotically unbiased. Moreover, Ramezani-Kebrya et al. (2023) introduced FTW-ERM, which integrates density ratio estimation.

Label shift and MLLS family For theoretical analysis, the conditional distribution $p(\boldsymbol{x}|\boldsymbol{y})$ is held strictly constant across all distributions (Lipton et al., 2018; Garg et al., 2020; Saerens et al., 2002). Both BBSE (Lipton et al., 2018) and RLLS (Azizzadenesheli et al., 2019) designate a discrete latent space $\boldsymbol{z}$ and introduce a confusion matrix-based estimation method to compute the ratio $\boldsymbol{w}$ by solving a linear system (Saerens et al., 2002; Lipton et al., 2018). This approach is straightforward and has been proven consistent, even when the predictor is not calibrated. However, its subpar performance is attributed to the information loss inherent in the confusion matrix (Garg et al., 2020).

Consequently, MLLS (Garg et al., 2020) introduces a continuous latent space, resulting in a significant enhancement in estimation performance, especially when combined with a post-hoc calibration method (Shrikumar et al., 2019). It also provides a consistency guarantee with a canonically calibrated predictor. This EM-based MLLS method is both concave and can be solved efficiently.

Discrepancy Measure In information theory and statistics, discrepancy measures play a critical role in quantifying the differences between probability distributions. One such measure is the Bregman Divergence (Banerjee et al., 2005), defined as

$$D_\phi(\boldsymbol{x}\|\boldsymbol{y}) = \phi(\boldsymbol{x}) - \phi(\boldsymbol{y}) - \langle \nabla\phi(\boldsymbol{y}), \boldsymbol{x} - \boldsymbol{y} \rangle,$$

which encapsulates the difference between the value of a convex function ϕ at two points and the value of the linear approximation of ϕ at one point, leveraging the gradient at another point.

Discrepancy measures are generally categorized into two main families: Integral Probability Metrics (IPMs) and f-divergences. IPMs, including Maximum Mean Discrepancy (Gretton et al., 2012) and Wasserstein distance (Villani, 2009), focus on distribution differences $P - Q$. In contrast, f-divergences, such as KL-divergence (Kullback & Leibler, 1951) and Total Variation distance, operate on ratios P/Q and do not satisfy the triangular inequality. Interconnections and variations between these families are explored in studies like (f, Γ)-Divergences (Birrell et al., 2022), which interpolate between f-divergences and IPMs, and research outlining optimal bounds between them (Agrawal & Horel, 2020).

MLLS (Garg et al., 2020) employs f-divergence, notably the KL divergence, which is not a metric as it doesn't satisfy the triangular inequality, and requires distribution P to be absolutely continuous with respect to Q. Concerning IPMs, while MMD is reliant on a kernel function, it can suffer from the curse of dimensionality when faced with high-dimensional data. On the other hand, the Wasserstein distance can be reformulated using Kantorovich-Rubinstein duality (Dedecker et al., 2006; Arjovsky et al., 2017) as a maximization problem subject to a Lipschitz constrained function $f : \mathbb{R}^d \to \mathbb{R}$.

B BBSE AND MLLS FAMILY

In this section, we summarize the contributions of BBSE (Lipton et al., 2018) and MLLS (Garg et al., 2020). Our objective is to estimate the ratio $p^{\text{te}}(y)/p^{\text{tr}}(y)$. We consider a scenario with m possible label classes, where $y = c$ for $c \in [m]$. Let $\boldsymbol{r}^\star = [r_1^\star, \ldots, r_m^\star]^\top$ represent the true ratios, with each $r_c^\star$ defined as $r_c^\star = \frac{p^{\text{te}}(y=c)}{p^{\text{tr}}(y=c)}$ (Garg et al., 2020). We then define a family of distributions over $\mathcal{Z}$, parameterized by $\boldsymbol{r} = [r_1, \ldots, r_m]^\top \in \mathbb{R}^m$, where r_c is the c-th element of the ratio vector.

$$p_{\boldsymbol{r}}(\boldsymbol{z}) := \sum_{c=1}^{m} p^{\text{te}}(\boldsymbol{z}|y=c) \cdot p^{\text{tr}}(y=c) \cdot r_c \tag{9}$$

Here, $r_c \geq 0$ for $c \in [m]$ and $\sum_{c=1}^m r_c \cdot p^{\text{tr}}(y=c) = \sum_{c=1}^m p^{\text{te}}(y=c) = 1$ as constraints. When $\boldsymbol{r} = \boldsymbol{r}^\star$, e.g., $r_c = r_c^\star$ for $c \in [m]$, we have $p_{\boldsymbol{r}}(\boldsymbol{z}) = p_{\boldsymbol{r}^\star}(\boldsymbol{z}) = p^{\text{te}}(\boldsymbol{z})$ (Garg et al., 2020). So our task is to find $\boldsymbol{r}$ such that

$$\begin{aligned} &\sum_{c=1}^{m} p^{\text{te}}(\boldsymbol{z}|y=c) \cdot p^{\text{tr}}(y=c) \cdot r_c \boldsymbol{x} \\ &= \sum_{c=1}^{m} p^{\text{tr}}(\boldsymbol{z}, y=c) \cdot r_c = p^{\text{te}}(\boldsymbol{z}) \end{aligned} \tag{10}$$

Lipton et al. (2018) introduced Black Box Shift Estimation (BBSE) to address this issue. With a pre-trained classifier f for the classification task, BBSE assumes that the latent space $\mathcal{Z}$ is discrete and defines $p(\boldsymbol{z}|\boldsymbol{x}) = \delta_{\arg\max f(\boldsymbol{x})}$, where the output of $f(\boldsymbol{x})$ is a probability vector (or a simplex) over m classes. BBSE estimates $p^{\text{te}}(\boldsymbol{z}|y)$ as a confusion matrix, using both the training and validation data. It calculates $p^{\text{tr}}(y=c)$ from the training set and $p^{\text{te}}(\boldsymbol{z})$ from the test data. The problem then reduces to solving the following equation:

$$\boldsymbol{A}\boldsymbol{w} = \boldsymbol{B} \tag{11}$$

where $|\mathcal{Z}| = m$, $\boldsymbol{A} \in \mathbb{R}^{m \times m}$ with $A_{jc} = p^{\text{te}}(z=j|y=c) \cdot p^{\text{tr}}(y=c)$, and $\boldsymbol{B} \in \mathbb{R}^m$ with $B_j = p^{\text{te}}(z=j)$ for $c, j \in [m]$.

The estimation of the confusion matrix in terms of $p^{\text{te}}(\boldsymbol{z}|y)$ leads to the loss of calibration information (Garg et al., 2020). Furthermore, when defining $\mathcal{Z}$ as a continuous latent space, the confusion matrix becomes intractable since $\boldsymbol{z}$ has infinitely many values. Therefore, MLLS directly minimizes the divergence between $p^{\text{te}}(\boldsymbol{z})$ and $p_{\boldsymbol{r}}(\boldsymbol{z})$, instead of solving the linear system in Equation (11).

Within the f-divergence family, MLLS seeks to find a weight vector $\boldsymbol{r}$ by minimizing the KL-divergence $D_{\text{KL}}\left(p^{\text{te}}(\boldsymbol{z}), p_{\boldsymbol{r}}(\boldsymbol{z})\right) = \mathbb{E}_{\text{te}}\left[\log p^{\text{te}}(\boldsymbol{z})/p_{\boldsymbol{r}}(\boldsymbol{z})\right]$, for $p_{\boldsymbol{r}}(\boldsymbol{z})$ defined in Equation (9). Leveraging on the properties of the logarithm, this is equivalent to maximizing the log-likelihood: $\boldsymbol{r} := \arg\max_{\boldsymbol{r}\in\mathbb{R}} \mathbb{E}_{\text{te}}\left[\log p_{\boldsymbol{r}}(\boldsymbol{z})\right]$. Expanding $p_{\boldsymbol{r}}(\boldsymbol{z})$, we have

$$\begin{aligned} \mathbb{E}_{\text{te}}\left[\log p_{\boldsymbol{r}}(\boldsymbol{z})\right] &= \mathbb{E}_{\text{te}}\left[\log(\sum_{c=1}^{m} p^{\text{tr}}(\boldsymbol{z}, y=c) r_c)\right] \\ &= \mathbb{E}_{\text{te}}\left[\log(\sum_{c=1}^{m} p^{\text{tr}}(y=c \mid \boldsymbol{z}) r_c) + \log p^{\text{tr}}(\boldsymbol{z})\right]. \end{aligned} \tag{12}$$

Therefore the unified form of MLLS can be formulated as:

$$\boldsymbol{r} := \arg\max_{\boldsymbol{r}\in\mathbb{R}} \mathbb{E}_{\text{te}}\left[\log(\sum_{c=1}^{m} p^{\text{tr}}(y=c \mid \boldsymbol{z}) r_c)\right]. \tag{13}$$

This is a convex optimization problem and can be solved efficiently using methods such as EM, an analytic approach, and also iterative optimization methods like gradient descent with labeled training data and unlabeled test data. MLLS defines the $p(\boldsymbol{z}|\boldsymbol{x})$ as $\delta_{\boldsymbol{x}}$, plugs in the pre-defined f to approximate $p^{\text{tr}}(y|\boldsymbol{x})$ and optimizes the following objective:

$$\boldsymbol{r}_f := \arg\max_{\boldsymbol{r}\in\mathbb{R}} \ell(\boldsymbol{r}, f) := \arg\max_{\boldsymbol{r}\in\mathbb{R}} \mathbb{E}_{\text{te}} \left[\log(f(\boldsymbol{x})^T \boldsymbol{r})\right]. \tag{14}$$

With the Bias-Corrected Calibration (BCT) (Shrikumar et al., 2019) strategy, they adjust the logits $\hat{f}(\boldsymbol{x})$ of $f(\boldsymbol{x})$ element-wise for each class, and the objective becomes:

$$\boldsymbol{r}_f := \arg\max_{\boldsymbol{r}\in\mathbb{R}} \ell(\boldsymbol{r}, f) := \arg\max_{\boldsymbol{r}\in\mathbb{R}} \mathbb{E}_{\text{te}} \left[\log(g \circ \hat{f}(\boldsymbol{x}))^T \boldsymbol{r})\right], \tag{15}$$

where g is a calibration function.

Scenario	#Nodes	Assumptions on Distributions	Ratio Node i Needs
`No-LS` in equation 16	2	$p_1^{\text{tr}}(\boldsymbol{y}) = p_1^{\text{te}}(\boldsymbol{y})$ and $p_1^{\text{tr}}(\boldsymbol{y}) \neq p_2^{\text{tr}}(\boldsymbol{y})$	$p_1^{\text{tr}}(\boldsymbol{y})/p_2^{\text{tr}}(\boldsymbol{y})$
`LS on single` in equation 17	2	$p_1^{\text{tr}}(\boldsymbol{y}) \neq p_1^{\text{te}}(\boldsymbol{y})$ and $p_2^{\text{tr}}(\boldsymbol{y}) = p_2^{\text{te}}(\boldsymbol{y})$	$p_1^{\text{te}}(\boldsymbol{y})/p_1^{\text{tr}}(\boldsymbol{y})$ and $p_1^{\text{te}}(\boldsymbol{y})/p_2^{\text{tr}}(\boldsymbol{y})$
`LS on both` in equation 17	2	$p_1^{\text{tr}}(\boldsymbol{y}) \neq p_1^{\text{te}}(\boldsymbol{y})$ and $p_2^{\text{tr}}(\boldsymbol{y}) \neq p_2^{\text{te}}(\boldsymbol{y})$	$p_1^{\text{te}}(\boldsymbol{y})/p_1^{\text{tr}}(\boldsymbol{y})$ and $p_1^{\text{te}}(\boldsymbol{y})/p_2^{\text{tr}}(\boldsymbol{y})$
`LS on multi` in equation 18	K	$p_k^{\text{tr}}(\boldsymbol{y}) \neq p_1^{\text{te}}(\boldsymbol{y})$ for all k	$p_1^{\text{te}}(\boldsymbol{y})/p_k^{\text{tr}}(\boldsymbol{y})$ for all k

Table 4: Details of scenarios described in Section 2

C Proof of Proposition 2.1

In the following, we consider four typical scenarios under various distribution shifts and formulate their IW-ERM with a focus on minimizing R_1.

C.1 No Intra-node Label Shift

For simplicity, we assume that there are only 2 nodes, but our results can be extended to multiple nodes. This scenario assumes $p_k^{\text{tr}}(\boldsymbol{y}) = p_k^{\text{te}}(\boldsymbol{y})$ for $k = 1, 2$, but $p_1^{\text{tr}}(\boldsymbol{y}) \neq p_2^{\text{tr}}(\boldsymbol{y})$. Node 1 aims to learn $h_{\boldsymbol{w}}$ assuming $\frac{p_1^{\text{tr}}(\boldsymbol{y})}{p_2^{\text{tr}}(\boldsymbol{y})}$ is given. We consider the following IW-ERM that is consistent in minimizing R_1:

$$\begin{aligned}\min_{h_{\boldsymbol{w}} \in \mathcal{H}} & \frac{1}{n_1^{\text{tr}}} \sum_{i=1}^{n_1^{\text{tr}}} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}_{1,i}^{\text{tr}}), \boldsymbol{y}_{1,i}^{\text{tr}}) \\ & + \frac{1}{n_2^{\text{tr}}} \sum_{i=1}^{n_2^{\text{tr}}} \frac{p_1^{\text{tr}}(\boldsymbol{y}_{2,i}^{\text{tr}})}{p_2^{\text{tr}}(\boldsymbol{y}_{2,i}^{\text{tr}})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}_{2,i}^{\text{tr}}), \boldsymbol{y}_{2,i}^{\text{tr}}).\end{aligned} \tag{16}$$

Here $\mathcal{H}$ is the hypothesis class of $h_{\boldsymbol{w}}$. This scenario is referred to as `No-LS`.

C.2 Label Shift only for Node 1

Here we consider label shift only for node 1, i.e., $p_1^{\text{tr}}(\boldsymbol{y}) \neq p_1^{\text{te}}(\boldsymbol{y})$ and $p_2^{\text{tr}}(\boldsymbol{y}) = p_2^{\text{te}}(\boldsymbol{y})$. We consider the following IW-ERM:

$$\begin{aligned}\min_{h_{\boldsymbol{w}} \in \mathcal{H}} & \frac{1}{n_1^{\text{tr}}} \sum_{i=1}^{n_1^{\text{tr}}} \frac{p_1^{\text{te}}(\boldsymbol{y}_{1,i}^{\text{tr}})}{p_1^{\text{tr}}(\boldsymbol{y}_{1,i}^{\text{tr}})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}_{1,i}^{\text{tr}}), \boldsymbol{y}_{1,i}^{\text{tr}}) \\ & + \frac{1}{n_2^{\text{tr}}} \sum_{i=1}^{n_2^{\text{tr}}} \frac{p_1^{\text{te}}(\boldsymbol{y}_{2,i}^{\text{tr}})}{p_2^{\text{tr}}(\boldsymbol{y}_{2,i}^{\text{tr}})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}_{2,i}^{\text{tr}}), \boldsymbol{y}_{2,i}^{\text{tr}}).\end{aligned} \tag{17}$$

This scenario is referred to as `LS on single`.

C.3 Label shift for both nodes

Here we assume $p_1^{\text{tr}}(\boldsymbol{y}) \neq p_1^{\text{te}}(\boldsymbol{y})$ and $p_2^{\text{tr}}(\boldsymbol{y}) \neq p_2^{\text{te}}(\boldsymbol{y})$, i.e., label shift for both nodes. The corresponding IW-ERM is the same as Eq. equation 17. This scenario is referred to as `LS on both`.

Without loss of generality and for simplicity, we set $l = 1$. We consider four typical scenarios under various distribution shifts and formulate their IW-ERM with a focus on minimizing R_1. The details of these scenarios are summarized in Table 4.

C.4 MULTIPLE NODES

Here we consider a general scenario with K nodes. We assume both intra-node and inter-node label shifts by the following IW-ERM:

$$\min_{h_{\boldsymbol{w}} \in \mathcal{H}} \sum_{k=1}^{K} \frac{\lambda_k}{n_k^{\text{tr}}} \sum_{i=1}^{n_k^{\text{tr}}} \frac{p_1^{\text{te}}(\boldsymbol{y}_{k,i}^{\text{tr}})}{p_k^{\text{tr}}(\boldsymbol{y}_{k,i}^{\text{tr}})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}_{k,i}^{\text{tr}}), \boldsymbol{y}_{k,i}^{\text{tr}}), \tag{18}$$

where $\sum_{k=1}^{K} \lambda_k = 1$ and $\lambda_k \geq 0$. This scenario is referred to as `LS on multi`.

For the scenario without intra-node label shift, the IW-ERM in Equation (16) can be expressed as

$$\begin{aligned}
&\frac{1}{n_2^{\text{tr}}} \sum_{i=1}^{n_2^{\text{tr}}} \frac{p_1^{\text{tr}}(\boldsymbol{y}_{2,i}^{\text{tr}})}{p_2^{\text{tr}}(\boldsymbol{y}_{2,i}^{\text{tr}})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}_{2,i}^{\text{tr}}), \boldsymbol{y}_{2,i}^{\text{tr}}) \\
&\qquad \xrightarrow{n_2^{\text{tr}} \to \infty} \mathbb{E}_{p_2^{\text{tr}}(\boldsymbol{x},\boldsymbol{y})} \left[\frac{p_1^{\text{tr}}(\boldsymbol{y})}{p_2^{\text{tr}}(\boldsymbol{y})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{y}) \right] \\
&\qquad = \int_{\mathcal{Y}} \frac{p_1^{\text{tr}}(\boldsymbol{y})}{p_2^{\text{tr}}(y)} \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{y})}[\ell(h_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{y})] p_2^{\text{tr}}(\boldsymbol{y}) d\boldsymbol{y}) \\
&\qquad = \int_{\mathcal{Y}} p_1^{\text{tr}}(\boldsymbol{y}) \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{y})}[\ell(h_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{y})] d\boldsymbol{y} \\
&\qquad = \int_{\mathcal{Y}} p_1^{\text{te}}(\boldsymbol{y}) \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{y})}[\ell(h_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{y})] d\boldsymbol{y} \\
&\qquad = \mathbb{E}_{p_1^{\text{te}}(\boldsymbol{x},\boldsymbol{y})} \left[\ell(h_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{y}) \right] \\
&\qquad = R_1(h_{\boldsymbol{w}}).
\end{aligned} \tag{19}$$

where the second equality holds due to the assumption of the label shift setting and Bayes' theorem: $p(\boldsymbol{x}, \boldsymbol{y}) = p(\boldsymbol{x}|\boldsymbol{y}) \cdot p(\boldsymbol{y})$, and the fourth equality holds by the assumption that $p_1^{\text{tr}}(\boldsymbol{y}) = p_1^{\text{te}}(\boldsymbol{y})$ in the No-LS setting.

For the scenario with label shift only for Node 1 or for both nodes, the IW-ERM in Equation (17) admits

$$\frac{1}{n_2^{\text{tr}}} \sum_{i=1}^{n_2^{\text{tr}}} \frac{p_1^{\text{te}}(\boldsymbol{y}_{2,i}^{\text{tr}})}{p_2^{\text{tr}}(\boldsymbol{y}_{2,i}^{\text{tr}})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}_{2,i}^{\text{tr}}), \boldsymbol{y}_{2,i}^{\text{tr}}) \tag{20}$$

$$\xrightarrow{n_2^{\text{tr}} \to \infty} \mathbb{E}_{p_2^{\text{tr}}(\boldsymbol{x},\boldsymbol{y})} \left[\frac{p_1^{\text{te}}(\boldsymbol{y})}{p_2^{\text{tr}}(\boldsymbol{y})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{y}) \right] \tag{21}$$

$$= \int_{\mathcal{Y}} \frac{p_1^{\text{te}}(y)}{p_2^{\text{tr}}(y)} \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{y})}[\ell(h_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{y})] p_2^{\text{tr}}(\boldsymbol{y}) d\boldsymbol{y} \tag{22}$$

$$= \int_{\mathcal{Y}} p_1^{\text{te}}(y = \boldsymbol{y}) \mathbb{E}_{p(\boldsymbol{x}|\boldsymbol{y})}[\ell(h_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{y})] d\boldsymbol{y} \tag{23}$$

$$= \mathbb{E}_{p_1^{\text{te}}(\boldsymbol{x},\boldsymbol{y})} \left[\ell(h_{\boldsymbol{w}}(\boldsymbol{x}), \boldsymbol{y}) \right] \tag{24}$$

$$= R_1(h_{\boldsymbol{w}}). \tag{25}$$

For multiple nodes, let $k \in [K]$. Similarly, we have

$$\frac{1}{n_k^{\text{tr}}} \sum_{i=1}^{n_k^{\text{tr}}} \frac{p_1^{\text{te}}(\boldsymbol{y}_{k,i}^{\text{tr}})}{p_k^{\text{tr}}(\boldsymbol{y}_{k,i}^{\text{tr}})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}_{k,i}^{\text{tr}}), \boldsymbol{y}_{k,i}^{\text{tr}}) \xrightarrow{n_k^{\text{tr}} \to \infty} R_1(h_{\boldsymbol{w}}). \tag{26}$$

Then we have

$$\sum_{k=1}^{K} \frac{\lambda_k}{n_k^{\text{tr}}} \sum_{i=1}^{n_k^{\text{tr}}} \frac{p_1^{\text{te}}(\boldsymbol{y}_{k,i}^{\text{tr}})}{p_k^{\text{tr}}(\boldsymbol{y}_{k,i}^{\text{tr}})} \ell(h_{\boldsymbol{w}}(\boldsymbol{x}_{k,i}^{\text{tr}}), \boldsymbol{y}_{k,i}^{\text{tr}}) \xrightarrow{n_1^{\text{tr}}, \ldots, n_K^{\text{tr}} \to \infty} R_1(h_{\boldsymbol{w}}). \tag{27}$$

Note that to solve Equation (18), node 1 needs to estimate $\frac{p_1^{\text{te}}(\boldsymbol{y})}{p_k^{\text{tr}}(\boldsymbol{y})}$ for all nodes k with $\lambda_k > 0$ in equation 18.

The consistency of Equation (IW-ERM), i.e., convergence in probability, is followed the standard arguments in e.g., (Shimodaira, 2000)[Section 3] and (Sugiyama et al., 2007)[Section 2.2] using the law of large numbers.

D ALGORITHMIC DESCRIPTION

Algorithm 3 IW-ERM with VRLS in Distributed Learning

Require: Labeled training data $\{(\boldsymbol{x}^{\text{tr}}_{k,i}, \boldsymbol{y}^{\text{tr}}_{k,i})\}_{i=1}^{n^{\text{tr}}_k}$ at each node k, for $k = [K]$.
Require: Unlabeled test data $\{\boldsymbol{x}^{\text{te}}_{k,j}\}_{j=1}^{n^{\text{te}}_k}$ at each node k, for $k = [K]$.
Require: Initial global model $h_{\boldsymbol{w}}$.
Ensure: Trained global model $h_{\boldsymbol{w}}$ optimized with IW-ERM.

1: **Phase 1: Density Ratio Estimation with VRLS**
2: **for each node** $k = 1$ to K **in parallel do**
3: Train local predictor $f_{k,\hat{\boldsymbol{\theta}}_{n^{\text{tr}}_k}}$ on local training data $\{(\boldsymbol{x}^{\text{tr}}_{k,i}, \boldsymbol{y}^{\text{tr}}_{k,i})\}$.
4: Use $f_{k,\hat{\boldsymbol{\theta}}_{n^{\text{tr}}_k}}$ to estimate the density ratio $\hat{\boldsymbol{r}}_{n^{\text{te}}_k}$ on unlabelled test data $\{\boldsymbol{x}^{\text{te}}_k\}$ at node k.
5: **end for**
6: **Phase 2: Importance Weight Computation**
7: **for each node** $k = 1$ to K **do**
8: Compute importance weight:

$$\omega_k = \frac{\sum_{j=1}^{K} \hat{\boldsymbol{r}}_{n^{\text{te}}_j} \cdot p^{\text{tr}}_j(\boldsymbol{y})}{p^{\text{tr}}_k(\boldsymbol{y})}$$

9: **end for**
10: **Phase 3: Global Model Training with IW-ERM**
11: Train global model $h_{\boldsymbol{w}}$ by minimizing the weighted empirical risk:

$$\min_{h_{\boldsymbol{w}}} \sum_{k=1}^{K} \frac{\lambda_k}{n^{\text{tr}}_k} \sum_{i=1}^{n^{\text{tr}}_k} \omega_k \cdot \ell\left(h_{\boldsymbol{w}}(\boldsymbol{x}^{\text{tr}}_{k,i}), \boldsymbol{y}^{\text{tr}}_{k,i}\right)$$

```python

# Split the training dataset on each node
trainsets = target_shift.split_dataset(trainset.data, trainset.targets,
    node_label_dist_train, transform=transform_train)

# Split the test dataset on each node
testsets = target_shift.split_dataset(testset.data, testset.targets,
    node_label_dist_test, transform=transform_test)

# Initialize K local models (nets) for each node
nets = [initialize_model() for _ in range(node_num)]

# Initialize the estimator for each local model
estimators = [LS_RatioModel(nets[k]) for k in range(node_num)]

# Initialize tensors to store the estimated ratios, values, and marginal
    values for each pair of nodes.
estimated_ratios = torch.zeros(node_num, node_num, nclass)
estimated_values = torch.zeros(node_num, node_num, nclass)
marginal_values = torch.zeros(node_num, nclass)

# Phase 1: Compute the estimated ratios for each node pair (k, j)
for k in range(node_num):
    for j in range(node_num):
        # Perform test on node k using node j's testset
        estimated_ratios[k, j] = estimators[k](testsets[j].data.cpu().
    numpy())

# Phase 2: Compute the marginal values on each node's training set
for i, trainset in enumerate(trainsets):
    marginal_values[i] = marginal(trainset.targets)

# Phase 3: Compute the final estimated values for each node
for k in range(node_num):
    for j in range(node_num):
        estimated_values[k, j] = marginal_values[j] * estimated_ratios[k,
     j]

# Aggregate the estimated values across nodes
aggregated_values = torch.sum(estimated_values, dim=1)

# Compute the final ratios for each node
ratios = (aggregated_values / marginal_values).to(args.device)
```

Listing 1: Our VRLS in distributed learning. It is the implementation of Algorithm 3

E PROOF OF THEOREM 5.1

Proof. Let $H(\boldsymbol{r},\boldsymbol{\theta},\boldsymbol{x}) = -\log(f(\boldsymbol{x},\boldsymbol{\theta})^\top \boldsymbol{r})$. From the strong convexity in Lemma E.7, we have that

$$\|\hat{\boldsymbol{r}}_{n^{\text{te}}} - \boldsymbol{r}_{f^\star}\|_2^2 \leq \frac{2}{\mu p_{\min}} \left(\mathcal{L}_{\boldsymbol{\theta}^\star}(\hat{\boldsymbol{r}}_{n^{\text{te}}}) - \mathcal{L}_{\boldsymbol{\theta}^\star}(\boldsymbol{r}_{f^\star})\right) \tag{28}$$

Now focusing on the term on the right-hand side, we find by invoking Lemma E.4 that

$$\begin{aligned}
&\mathcal{L}_{\boldsymbol{\theta}^\star}(\hat{\boldsymbol{r}}_{n^{\text{te}}}) - \mathcal{L}_{\boldsymbol{\theta}^\star}(\boldsymbol{r}_{f^\star}) \\
&\leq \mathbb{E}\left[H(\hat{\boldsymbol{r}}_{n^{\text{te}}}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, \boldsymbol{x})\right] - \mathbb{E}\left[H(\boldsymbol{r}_{f^\star}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, \boldsymbol{x})\right] + 2L\mathbb{E}\left[\|\hat{\boldsymbol{\theta}}_{n^{\text{tr}}} - \boldsymbol{\theta}^\star\|_2\right] \\
&= \mathbb{E}\left[H(\hat{\boldsymbol{r}}_{n^{\text{te}}}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, x)\right] - \frac{1}{n^{\text{te}}}\sum_{j=1}^{n^{\text{te}}} H(\hat{\boldsymbol{r}}_{n^{\text{te}}}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, \boldsymbol{x}_j) + \frac{1}{n^{\text{te}}}\sum_{j=1}^{n^{\text{te}}} H(\hat{\boldsymbol{r}}_{n}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, \boldsymbol{x}_j) \\
&\qquad - \mathbb{E}\left[H(\boldsymbol{r}_{f^\star}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, \boldsymbol{x})\right] + 2L\mathbb{E}\left[\|\hat{\boldsymbol{\theta}}_{n^{\text{tr}}} - \boldsymbol{\theta}^\star\|_2\right] \\
&\leq \mathbb{E}\left[H(\hat{\boldsymbol{r}}_{n^{\text{te}}}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, \boldsymbol{x})\right] - \frac{1}{n^{\text{te}}}\sum_{j=1}^{n^{\text{te}}} H(\hat{\boldsymbol{r}}_{n^{\text{te}}}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, \boldsymbol{x}_j) + \frac{1}{n^{\text{te}}}\sum_{j=1}^{n^{\text{te}}} H(\boldsymbol{r}_{f^\star}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, \boldsymbol{x}_j) \\
&\qquad - \mathbb{E}\left[H(\boldsymbol{r}_{f^\star}, \hat{\boldsymbol{\theta}}_{n^{\text{tr}}}, \boldsymbol{x})\right] + 2L\mathbb{E}\left[\|\hat{\boldsymbol{\theta}}_{n^{\text{tr}}} - \boldsymbol{\theta}^\star\|_2\right],
\end{aligned} \tag{29}$$

where in the last inequality we used the fact that $\hat{\boldsymbol{r}}_n$ is a minimizer of $\boldsymbol{r} \mapsto \frac{1}{n}\sum_{j=1}^n H(\boldsymbol{r}, \hat{\boldsymbol{\theta}}_t, \boldsymbol{x}_j)$. Finally by using Lemma E.5 and Lemma E.6 with $\delta/2$ each, we have that with probability $1-\delta$,

$$\mathcal{L}_{\boldsymbol{\theta}^\star}(\hat{\boldsymbol{r}}_{n^{\text{te}}}) - \mathcal{L}_{\boldsymbol{\theta}^\star}(\boldsymbol{r}_{f^\star}) \leq \frac{4}{\sqrt{n^{\text{te}}}}\text{Rad}(\mathcal{F}) + 2L\mathbb{E}\left[\|\hat{\boldsymbol{\theta}}_{n^{\text{tr}}} - \boldsymbol{\theta}^\star\|_2\right] + 4B\sqrt{\frac{\log(4/\delta)}{n^{\text{te}}}} \tag{30}$$

Plugging this back into Equation (28), we have that

$$\|\hat{\boldsymbol{r}}_{n^{\text{te}}} - \boldsymbol{r}_{f^\star}\|_2^2 \leq \frac{2}{\mu p_{\min}}\left(\frac{4}{\sqrt{n^{\text{te}}}}\text{Rad}(\mathcal{F}) + 4B\sqrt{\frac{\log(4/\delta)}{n^{\text{te}}}}\right) + \frac{4L}{\mu p_{\min}}\mathbb{E}\left[\|\hat{\boldsymbol{\theta}}_{n^{\text{tr}}} - \boldsymbol{\theta}^\star\|_2\right]. \tag{31}$$

□

Lemma E.1. *For any $\boldsymbol{r} \in \mathbb{R}_+^m,\ \boldsymbol{\theta} \in \Theta,\ \boldsymbol{x} \in \mathcal{X}$, we have that*

$$\boldsymbol{r}^\top f(\boldsymbol{x},\boldsymbol{\theta}) \leq \frac{1}{p_{min}}.$$

Proof. Applying Hölder's inequality we have that

$$\boldsymbol{r}^\top f(\boldsymbol{x},\boldsymbol{\theta}) \leq \|\boldsymbol{r}\|_\infty \|f(\boldsymbol{x},\boldsymbol{\theta})\|_1 = \|\boldsymbol{r}\|_\infty.$$

Moreover, since $\boldsymbol{r} \in \mathbb{R}_+^m$, we have that $\sum_y r_y p_{tr}(y) = 1$ This implies that $\|\boldsymbol{r}\|_\infty \leq \frac{1}{p_{\min}}$, which yields the result. □

Lemma E.2 (Implication of Assumption Assumption 5.1)**.** *Under Assumption 5.1, there exists $B > 0$ such that for any $\boldsymbol{r} \in \mathbb{R}_+^m,\ \boldsymbol{\theta} \in \Theta,\ \boldsymbol{x} \in \mathcal{X}$,*

$$|\log(\boldsymbol{r}^\top f(\boldsymbol{x},\boldsymbol{\theta}))| \leq B.$$

Proof. Since $\boldsymbol{r} \in \mathbb{R}_+^m$, it has at least one non-zero coordinate and $f(\boldsymbol{x},\boldsymbol{\theta})$ is the output of a softmax layer so all of its coordinates are non-zero. Consequently,

$$\boldsymbol{r}^\top f(\boldsymbol{x},\boldsymbol{\theta}) > 0$$

So by Assumption 5.1, the function $(\boldsymbol{r},\boldsymbol{\theta},\boldsymbol{x}) \mapsto \log(\boldsymbol{r}^\top f(\boldsymbol{x},\boldsymbol{\theta}))$ is defined and continuous over a compact set, so there exists a constant B giving us the result. □

Lemma E.3 (Population Strong Convexity). *Let $H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x}) = -\log(\boldsymbol{r}^\top f(\boldsymbol{x}, \boldsymbol{\theta}))$. Under Assumption Assumption 5.2, the function*

$$\mathcal{L}_{\boldsymbol{\theta}^\star} : \boldsymbol{r} \mapsto \mathbb{E}\Big[H(\boldsymbol{r}, \boldsymbol{\theta}^\star, \boldsymbol{x})\Big]$$

is $\mu p_{\min}$-strongly convex.

Proof. We first compute the Hessian of $\mathcal{L}$ to find that

$$\nabla^2 \mathcal{L}(\boldsymbol{r}) = \mathbb{E}\left[\frac{1}{(\boldsymbol{r}^\top f(\boldsymbol{x}, \boldsymbol{\theta}^\star))^2} f(\boldsymbol{x}, \boldsymbol{\theta}^\star) f(\boldsymbol{x}, \boldsymbol{\theta}^\star)^\top\right].$$

Since by Lemma E.1, we have that $\boldsymbol{r}^\top f(\boldsymbol{x}, \boldsymbol{\theta}^\star) \leq p_{\min}^{-1}$, we conclude that

$$\nabla^2 \mathcal{L}(\boldsymbol{r}) \succeq p_{\min} \mathbb{E}\Big[f(\boldsymbol{x}, \boldsymbol{\theta}^\star) f(\boldsymbol{x}, \boldsymbol{\theta}^\star)^\top\Big] \succeq \mu p_{\min} \mathbf{I}_m.$$

□

Lemma E.4 (Lipschitz Parametrization). *Let $H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x}) = -\log(f(\boldsymbol{x}, \boldsymbol{\theta})^\top \boldsymbol{r})$. There exists $L > 0$ such that for any $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2 \in \Theta$, and $\boldsymbol{r} \in \mathbb{R}_+^m$, we have that*

$$|H(\boldsymbol{r}, \boldsymbol{\theta}_1, \boldsymbol{x}) - H(\boldsymbol{r}, \boldsymbol{\theta}_2, \boldsymbol{x})| \leq L \|\boldsymbol{\theta}_1 - \boldsymbol{\theta}_2\|_2.$$

Proof. The gradient of H with respect to $\boldsymbol{\theta}$ is given by

$$\nabla_{\boldsymbol{\theta}} H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x}) = -\frac{1}{f(\boldsymbol{x}, \boldsymbol{\theta})^\top \boldsymbol{r}} \nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta})$$

Reasoning like in Lemma E.1, we know that $\frac{1}{f(\boldsymbol{x}, \boldsymbol{\theta})^\top \boldsymbol{r}}$ is defined and continuous over the compact set of its parameters, we also know that f is a neural network parametrized by $\boldsymbol{\theta}$, hence $\nabla_{\boldsymbol{\theta}} f(\boldsymbol{x}, \boldsymbol{\theta})$ is bounded when $\boldsymbol{\theta}$ and $\boldsymbol{x}$ are bounded. Consequently, under Assumption 5.1, there exists a constant $L > 0$ such that

$$\|\nabla_{\boldsymbol{\theta}} H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x})\|_2 \leq L.$$

□

Lemma E.5 (Uniform Bound 1). *Let $\delta \in (0, 1)$, with probability $1 - \delta$, we have that*

$$\begin{aligned} &\mathbb{E}\Big[H(\hat{\boldsymbol{r}}_n, \hat{\boldsymbol{\theta}}_t, \boldsymbol{x})\Big] - \frac{1}{n} \sum_{j=1}^n H(\hat{\boldsymbol{r}}_n, \hat{\boldsymbol{\theta}}_t, \boldsymbol{x}_j) \\ &\leq \frac{2}{\sqrt{n}} \mathit{Rad}(\mathcal{F}) + 2B \sqrt{\frac{\log(4/\delta)}{n}}. \end{aligned} \tag{32}$$

Proof. Let $\delta \in (0, 1)$. Since $\hat{\boldsymbol{r}}_n$ is learned from the samples $\boldsymbol{x}_j$, we do not have independence, which would have allowed us to apply a concentration inequality. Hence, we derive a uniform bound as follows. We begin by observing that:

$$\begin{aligned} &\mathbb{E}\Big[H(\hat{\boldsymbol{r}}_n, \hat{\boldsymbol{\theta}}_t, \boldsymbol{x})\Big] - \frac{1}{n} \sum_{j=1}^n H(\hat{\boldsymbol{r}}_n, \hat{\boldsymbol{\theta}}_t, \boldsymbol{x}_j) \\ &\leq \sup_{\boldsymbol{r}, \boldsymbol{\theta}} \left(\mathbb{E}\Big[H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x})\Big] - \frac{1}{n} \sum_{j=1}^n H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x}_j)\right) \end{aligned}$$

Now since Lemma E.2 holds, we can apply McDiarmid's Inequality to get that with probability $1 - \delta$, we have:

$$\begin{aligned} &\sup_{\boldsymbol{r}, \boldsymbol{\theta}} \left(\mathbb{E}\Big[H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x})\Big] - \frac{1}{n} \sum_{j=1}^n H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x}_j)\right) \\ &\leq \mathbb{E}\left[\sup_{\boldsymbol{r}, \boldsymbol{\theta}} \left(\mathbb{E}[H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x})] - \frac{1}{n} \sum_{j=1}^n H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x}_j)\right)\right] + 2B \sqrt{\frac{\log(2/\delta)}{n}} \end{aligned}$$

The expectation of the supremum on the right-hand side can be bounded by the Rademacher complexity of $\mathcal{F} := \{\boldsymbol{x} \mapsto \boldsymbol{r}^\top f(\boldsymbol{x}, \boldsymbol{\theta}),\ (\boldsymbol{r}, \boldsymbol{\theta}) \in \mathbb{R}_+^m \times \Theta\}$, and we obtain:

$$\begin{aligned} &\sup_{\boldsymbol{r},\boldsymbol{\theta}} \left(\mathbb{E}\big[H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x})\big] - \frac{1}{n}\sum_{j=1}^{n} H(\boldsymbol{r}, \boldsymbol{\theta}, \boldsymbol{x}_j) \right) \\ &\leq \frac{2}{\sqrt{n}}\mathrm{Rad}(\mathcal{F}) + 2B\sqrt{\frac{\log(2/\delta)}{n}}. \end{aligned} \tag{33}$$

□

Lemma E.6 (Uniform Bound 2)**.** *Let $\delta \in (0, 1)$, with probability $1 - \delta$, we have that*

$$\begin{aligned} &\mathbb{E}\Big[H(\boldsymbol{r}_{f^\star}, \hat{\boldsymbol{\theta}}_t, \boldsymbol{x})\Big] - \frac{1}{n}\sum_{j=1}^{n} H(\boldsymbol{r}_{f^\star}, \hat{\boldsymbol{\theta}}_t, \boldsymbol{x}_j) \\ &\leq \frac{2}{\sqrt{n}}Rad(\mathcal{F}) + 2B\sqrt{\frac{\log(2/\delta)}{n}}. \end{aligned} \tag{34}$$

Proof. The proof is identical to that of Lemma E.5. □

Lemma E.7 (Strong Convexity of Population Loss)**.** *Let $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\theta})$ be the population loss as defined in Lemma E.7. We establish that $\mathcal{L}(\boldsymbol{r}, \boldsymbol{\theta})$ is $\mu p_{\min}$-strongly convex under the assumptions of calibration (Assumption 5.2).*

Proof. We compute the Hessian of the population loss $\mathcal{L}$ as in Lemma E.7, obtaining that:

$$\nabla^2 \mathcal{L}(\boldsymbol{r}) = \mathbb{E}\left[\frac{1}{(\boldsymbol{r}^\top f(\boldsymbol{x}, \boldsymbol{\theta}))^2} f(\boldsymbol{x}, \boldsymbol{\theta}) f(\boldsymbol{x}, \boldsymbol{\theta})^\top\right].$$

From Lemma E.1, we have that $\boldsymbol{r}^\top f(\boldsymbol{x}, \boldsymbol{\theta}) \leq p_{\min}^{-1}$. Therefore, we conclude:

$$\nabla^2 \mathcal{L}(\boldsymbol{r}) \succeq p_{\min} \mathbb{E}\Big[f(\boldsymbol{x}, \boldsymbol{\theta}) f(\boldsymbol{x}, \boldsymbol{\theta})^\top\Big] \succeq \mu p_{\min} \mathbf{I}_m.$$

□

Lemma E.8 (Bound on Empirical Loss)**.** *Under Assumption 5.1, the empirical loss $\mathcal{L}_{n^{te}}(\boldsymbol{r}, \hat{\boldsymbol{\theta}}_{n^{tr}})$ satisfies the following concentration bound:*

$$\mathbb{P}\left(\sup_{\boldsymbol{r} \in \mathbb{R}_+^m} \left| \mathcal{L}_{n^{te}}(\boldsymbol{r}, \hat{\boldsymbol{\theta}}_{n^{tr}}) - \mathcal{L}(\boldsymbol{r}, \hat{\boldsymbol{\theta}}_{n^{tr}}) \right| > \epsilon \right) \leq 2\exp\left(-c n^{te} \epsilon^2\right).$$

Proof. This result follows from standard concentration inequalities, such as McDiarmid's inequality, together with the Lipschitz continuity of the loss function $\mathcal{L}$ with respect to the samples. □

F PROOF OF THEOREM 5.2 AND CONVERGENCE-COMMUNICATION GUARANTEES FOR IW-ERM WITH VRLS

We now establish convergence rates for IW-ERM with VRLS and show our proposed importance weighting achieves *the same rates* with the data-dependent *constant terms* increase linearly with $\max_{y \in \mathcal{Y}} \sup_f r_f(y) = r_{\max}$ under negligible communication overhead over the baseline ERM-solvers without importance weighting. In Appendix F, we establish tight convergence rates and communication guarantees for IW-ERM with VRLS in a broad range of importance optimization settings including convex optimization, second-order differentiability, composite optimization with proximal operator, optimization with adaptive step-sizes, and nonconvex optimization, along the lines of e.g., (Woodworth et al., 2020; Haddadpour et al., 2021; Glasgow et al., 2022; Liu et al., 2023; Hu & Huang, 2023; Wu et al., 2023; Liu et al., 2023).

By estimating the ratios locally and absorbing into local losses, we note that the properties of the modified local loss w.r.t. the neural network parameters $\boldsymbol{w}$, e.g., convexity and smoothness, do not change. The data-dependent parameters such as Lipschitz and smoothness constants for $\ell \circ h_{\boldsymbol{w}}$ w.r.t. $\boldsymbol{w}$ are scaled linearly by $r_{\max}$. Our method of density ratio estimation trains the pre-defined predictor *exclusively using local training data*, which implies IW-ERM with VRLS achieves the same privacy guarantees as the baseline ERM-solvers without importance weighting. For ratio estimation, the communication between clients involves only the estimated marginal label distribution, instead of data, ensuring negligible communication overhead. Given the size of variables to represent marginal distributions, which is by orders of magnitude smaller than the number of parameters of the underlying neural networks for training and the fact that ratio estimation involves only one round of communication, the overall communication overhead for ratio estimation is masked by the communication costs of model training. The communication costs for IW-ERM with VRLS over the course of optimization are exactly the same as those of the baseline ERM-solvers without importance weighting. All in all, importance weighting does not negatively impact communication guarantees throughout the course of optimization, which proves Theorem 5.2.

In the following, we establish tight convergence rates and communication guarantees for IW-ERM with VRLS in a broad range of importance optimization settings including convex optimization, second-order differentiability, composite optimization with proximal operator, optimization with adaptive step-sizes, and nonconvex optimization.

For convex and second-order Differentiable optimization, we establish a lower bound on the convergence rates for IW-ERM in with VRLS and local updating along the lines of e.g., (Glasgow et al., 2022, Theorem 3.1).

Assumption F.1 (PL with Compression)**.** *1) The $\ell(h_{\boldsymbol{w}}(\boldsymbol{x}), y)$ is β-smoothness and convex w.r.t. $\boldsymbol{w}$ for any $(\boldsymbol{x}, y)$ and satisfies Polyak-Łojasiewicz (PL) condition (there exists $\alpha_\ell > 0$ such that, for all $\boldsymbol{w} \in \mathcal{W}$, we have $\ell(h_{\boldsymbol{w}}) \leq \|\nabla_{\boldsymbol{w}} \ell(h_{\boldsymbol{w}})\|_2^2/(2\alpha_\ell)$; 2) The compression scheme $\mathcal{Q}$ is unbiased with bounded variance, i.e., $\mathbb{E}[\mathcal{Q}(\boldsymbol{x})] = \boldsymbol{x}$ and $\mathbb{E}[\|\mathcal{Q}(\boldsymbol{x}) - \boldsymbol{x}\|_2^2 \leq q\|\boldsymbol{x}\|_2^2]$; 3) The stochastic gradient $\boldsymbol{g}(\boldsymbol{w}) = \widetilde{\nabla}_{\boldsymbol{w}} \ell(h_{\boldsymbol{w}})$ is unbiased, i.e., $\mathbb{E}[\boldsymbol{g}(\boldsymbol{w})] = \nabla_{\boldsymbol{w}} \ell(h_{\boldsymbol{w}})$ for any $\boldsymbol{w} \in \mathcal{W}$ with bounded variance $\mathbb{E}[\|\boldsymbol{g}(\boldsymbol{w}) - \nabla_{\boldsymbol{w}} \ell(h_{\boldsymbol{w}})\|_2^2]$.*

For nonconvex optimization with PL condition and communication compression, we establish convergence and communication guarantees for IW-ERM with VRLS, compression, and local updating along the lines of e.g., (Haddadpour et al., 2021, Theorem 5.1).

Theorem F.1 (Convergence and Communication Bounds for Nonconvex Optimization with PL)**.** *Let κ denote the condition number, τ denote the number of local steps, R denote the number of communication rounds, and $\max_{y \in \mathcal{Y}} \sup_f r_f(y) = r_{\max}$. Under Assumption F.1, suppose Algorithm 2 with τ local updates and communication compression (Haddadpour et al., 2021, Algorithm 1) is run for $T = \tau R$ total stochastic gradients per node with fixed step-sizes $\eta = 1/(2r_{\max}\beta\gamma\tau(q/K + 1))$ and $\gamma \geq K$. Then we have $\mathbb{E}[\ell(h_{\boldsymbol{w}_T}) - \ell(h_{\boldsymbol{w}^\star})] \leq \epsilon$ by setting*

$$R \lesssim \left(\frac{q}{K} + 1\right)\kappa \log\left(\frac{1}{\epsilon}\right) \quad \text{and} \quad \tau \lesssim \left(\frac{q+1}{K(q/K+1)\epsilon}\right). \tag{35}$$

Assumption F.2 (Nonconvex Optimization with Adaptive Step-sizes)**.** *1) The $\ell \circ h_{\boldsymbol{w}}$ is β-smoothness with bounded gradients; 2) The stochastic gradients $\boldsymbol{g}(\boldsymbol{w}) = \widetilde{\nabla}_{\boldsymbol{w}} \ell(h_{\boldsymbol{w}})$ is unbiased with bounded variance $\mathbb{E}[\|\boldsymbol{g}(\boldsymbol{w}) - \nabla_{\boldsymbol{w}} \ell(h_{\boldsymbol{w}})\|_2^2]$; 3) Adaptive matrices A_t constructed as in (Wu et al., 2023, Algorithm 2) are diagonal and the minimum eigenvalues satisfy $\lambda_{\min}(A_t) \geq \rho > 0$ for some $\rho \in \mathbb{R}_+$.*

For nonconvex optimization with adaptive step-sizes, we establish convergence and communication guarantees for IW-ERM with VRLS and local updating along the lines of e.g., (Wu et al., 2023, Theorem 2).

Theorem F.2 (Convergence and Communication Guarantees for Nonconvex Optimization with Adaptive Step-sizes)**.** *Let τ denote the number of local steps, R denote the number of communication rounds, and $\max_{y\in\mathcal{Y}}\sup_f r_f(y) = r_{\max}$. Under Assumption F.2, suppose Algorithm 2 with τ local updates is run for $T = \tau R$ total stochastic gradients per node with an adaptive step-size similar to (Wu et al., 2023, Algorithm 2). Then we $\mathbb{E}[\|\nabla_{\boldsymbol{w}}\ell(h_{\boldsymbol{w}_T})\|_2] \leq \epsilon$ by setting:*

$$T \lesssim \frac{r_{\max}}{K\epsilon^3} \quad \textit{and} \quad R \lesssim \frac{r_{\max}}{\epsilon^2}. \tag{36}$$

Assumption F.3 (Composite Optimization with Proximal Operator)**.** *1) The $\ell \circ h_{\boldsymbol{w}}$ is smooth and strongly convex with condition number κ; 2) The stochastic gradients $\boldsymbol{g}(\boldsymbol{w}) = \widetilde{\nabla}_{\boldsymbol{w}}\ell(h_{\boldsymbol{w}})$ is unbiased.*

For composite optimization with strongly convex and smooth functions and proximal operator, we establish an upper bound on oracle complexity to achieve ϵ error on the Lyapunov function defined as in (Hu & Huang, 2023, Section 4) for Gradient Flow-type transformation of IW-ERM with VRLS in the limit of infinitesimal step-size.

Theorem F.3 (Oracle Complexity of Proximal Operator for Composite Optimization)**.** *Let κ denote the condition number. Under Assumption F.3, suppose Gradient Flow-type transformation of Algorithm 2 with VRLS and Proximal Operator evolves in the limit of infinitesimal step-size (Hu & Huang, 2023, Algorithm 3). Then it achieves $\mathcal{O}\big(r_{\max}\sqrt{\kappa}\log(1/\epsilon)\big)$ Proximal Operator Complexity.*

G Complexity Analysis

In our algorithm, the ratio estimation is performed once in parallel before the IW-ERM step.

In the experiments, we used a simple network to estimate the ratios in advance, which required significantly less computational effort compared to training the global model. Although IW-ERM with VRLS introduces additional computational complexity compared to the baseline FedAvg, it results in substantial improvements in overall generalization, particularly under challenging label shift conditions.

H MATHEMATICAL NOTATIONS

In this appendix, we provide a summary of mathematical notations used in this paper in Table 5:

Table 5: Math Symbols

Math Symbol	Definition
$\mathcal{X}$	Compact metric space for features
$\mathcal{Y}$	Discrete label space with $\lvert\mathcal{Y}\rvert = m$
K	Number of clients in an FL setting
$\mathcal{S}_k$	All samples in the training set of client k
$h_{\boldsymbol{w}}$	Hypothesis function $h_{\boldsymbol{w}} : \mathcal{X} \to \mathcal{Y}$
$\mathcal{H}$	Hypothesis class for $h_{\boldsymbol{w}}$
$\mathcal{Z}$	Mapping space from $\mathcal{X}$, which can be discrete or continuous

I LIMITATIONS

The distribution shifts observed in real-world data are often not fully captured by the label shift or relaxed distribution shift assumptions. In our experiments, we applied mild test data augmentation to approximate the relaxed label shift and manage ratio estimation errors for both the baselines and our method. However, the label shift assumption remains overly restrictive, and the relaxed label shift lacks robust empirical validation in practical scenarios.

Additionally, IW-ERM's parameter estimation relies on local predictors at each client, which limits its scalability. In practice, a simpler global predictor could be sufficient for parameter estimation and IW-ERM training. Future research could explore VRLS variants capable of effectively handling more complex distribution shifts in challenging datasets, such as CIFAR-10.1 (Recht et al., 2018; Torralba et al., 2008), as suggested in (Garg et al., 2023).

J EXPERIMENTAL DETAILS AND ADDITIONAL EXPERIMENTS

In this section, we provide experimental details and additional experiments. In particular, we validate our theory on multiple clients in a federated setting and show that our IW-ERM outperforms FedAvg and FedBN baselines *under drastic and challenging label shifts*.

J.1 EXPERIMENTAL DETAILS

In single-client experiments, a simple MLP without dropout is used as the predictor for MNIST, and ResNet-18 for CIFAR-10.

For experiments in a federated learning setting, both MNIST (LeCun et al., 1998) and Fashion MNIST (Xiao et al., 2017) datasets are employed, each containing 60,000 training samples and 10,000 test samples, with each sample being a 28 by 28 pixel grayscale image. The CIFAR-10 dataset (Krizhevsky) comprises 60,000 colored images, sized 32 by 32 pixels, spread across 10 classes with 6,000 images per class; it is divided into 50,000 training images and 10,000 test images. In this setting, the objective is to minimize the cross-entropy loss. Stochastic gradients for each client are calculated with a batch size of 64 and aggregated on the server using the Adam optimizer. LeNet is used for experiments on MNIST and Fashion MNIST with a learning rate of 0.001 and a weight decay of 1×10^{-6}. For CIFAR-10, ResNet-18 is employed with a learning rate of 0.0001 and a weight decay of 0.0001. Three independent runs are implemented for 5-client experiments on Fashion MNIST and CIFAR-10, while for 10 clients, one run is conducted on CIFAR-10. The regularization coefficient ζ in Equation (2) is set to 1 for all experiments. All experiments are performed using a single GPU on an internal cluster and Colab.

Importantly, the training of the predictor for ratio estimation on both the baseline MLLS and our VRLS is executed with identical hyperparameters and epochs for CIFAR-10 and Fashion MNIST. The training is halted once the classification loss reaches a predefined threshold on MNIST.

J.2 RELAXED LABEL SHIFT EXPERIMENTS

In conventional label shift, it is assumed that $p(\boldsymbol{x} \mid y)$ remains unchanged across training and test data. However, this assumption is often too strong for real-world applications, such as in healthcare, where different hospitals may use varying equipment, leading to shifts in $p(\boldsymbol{x} \mid y)$ even with the same labels (Rajendran et al., 2023). Relaxed label shift loosens this assumption by allowing small changes in the conditional distribution (Garg et al., 2023; Luo & Ren, 2022).

To formalize this, we use the distributional distance $\mathcal{D}$ and a relaxation parameter $\epsilon > 0$, as defined by Garg et al. (2023): $\max_y \mathcal{D}\left(p_{\text{tr}}(\boldsymbol{x} \mid y), p_{\text{te}}(\boldsymbol{x} \mid y)\right) \leq \epsilon$. This allows for slight differences in feature distributions between training and testing, capturing a more realistic scenario where the conditional distribution is not strictly invariant.

In our case, visual inspection suggests that the differences between temporally distinct datasets, such as CIFAR-10 and CIFAR-10.1_v6 (Torralba et al., 2008; Recht et al., 2018), may not meet the assumption of a small ϵ. To address this, we instead simulate controlled shifts using test data augmentation, allowing us to regulate the degree of relaxation, following the approach outlined in Garg et al. (2023).

J.3 ADDITIONAL EXPERIMENTS

In this section, we provide supplementary results, visualizations of accuracy across clients and tables showing dataset distribution in FL setting and relaxed label shift.

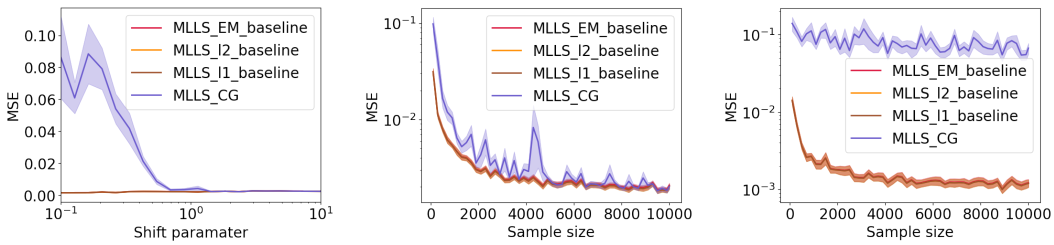

Figure 3: MSE analysis on MNIST for MLLS baselines. **Left:** Performance evaluation across various alpha values, comparing different methods: MLLS_EM, MLLS_L1, MLLS_L2, and MLLS_CG. MLLS_L1 and MLLS_L2 utilize convex optimization with L_1 and L_2 regularization for estimating our limited test sample problem, respectively, and are solved directly with a convex solver. In contrast, MLLS_CG uses conjugate gradient descent and MLLS_EM solves this convex optimization problem with EM algorithm. Both the EM and convex optimization methods (MLLS_L1, MLLS_L2) demonstrate superior and more consistent performance, especially under severe label shift conditions, when compared to MLLS_CG. **Middle:** At an alpha value of 1.0, the MSE analysis shows comparable performance across most methods, with the exception of MLLS_CG, which lags behind. **Right:** For alpha=0.1, MLLS_CG performs significantly worse than the EM and convex optimization methods, consistent with the trends observed in the left plot.

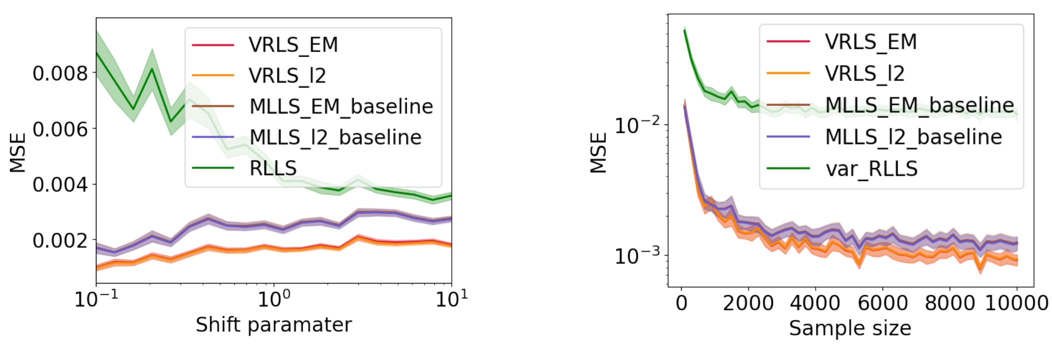

Figure 4: In our detailed analysis with the MNIST dataset, we conduct a thorough comparison of VRLS alongside MLLS (Garg et al., 2020), EM (Saerens et al., 2002), and also RLLS (Azizzadenesheli et al., 2019).

Table 6: LeNet on Fashion MNIST with label shift across 5 clients. 15,000 iterations for FedAvg and FedBN; 5,000 for Upper Bound (FTW-ERM) using true ratios and our IW-ERM. To mention, to train our predictor, we use a simpliest MLP and employ linear kernel.

FMNIST	**Our IW-ERM**	FedAvg	FedBN	Upper Bound
Avg. accuracy	$\mathbf{0.7520 \pm 0.0209}$	0.5472 ± 0.0297	0.5359 ± 0.0306	0.8273 ± 0.0041
Client 1 accuracy	$\mathbf{0.7162 \pm 0.0059}$	0.3616 ± 0.0527	0.3261 ± 0.0296	0.8590 ± 0.0062
Client 2 accuracy	$\mathbf{0.9266 \pm 0.0125}$	0.9060 ± 0.0157	0.9035 ± 0.0162	0.9357 ± 0.0037
Client 3 accuracy	$\mathbf{0.6724 \pm 0.0467}$	0.3279 ± 0.0353	0.3612 ± 0.0814	0.7896 ± 0.0109
Client 4 accuracy	$\mathbf{0.7979 \pm 0.0448}$	0.6858 ± 0.0105	0.6654 ± 0.0121	0.8098 ± 0.0112
Client 5 accuracy	$\mathbf{0.6468 \pm 0.0248}$	0.4548 ± 0.0655	0.4234 ± 0.0387	0.7426 ± 0.0257

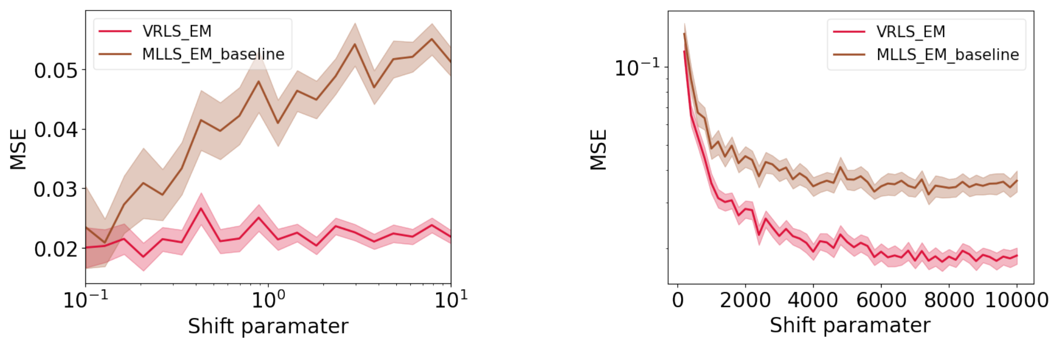

Figure 5: In this experiment with Fashion MNIST, a simple MLP with dropout were employed.

Table 7: ResNet-18 on CIFAR-10 with label shift across 5 clients. For fair comparison, we run 5,000 iterations for our method and Upper Bound, while 10000 for FedAvg and FedBN.

CIFAR-10	Our IW-ERM	FedAvg	FedBN	Upper Bound
Avg. accuracy	$\mathbf{0.5640 \pm 0.0241}$	0.4515 ± 0.0148	0.4263 ± 0.0975	0.5790 ± 0.0103
Client 1 accuracy	$\mathbf{0.6410 \pm 0.0924}$	0.5405 ± 0.1845	0.5321 ± 0.0620	0.7462 ± 0.0339
Client 2 accuracy	$\mathbf{0.8434 \pm 0.0359}$	0.3753 ± 0.0828	0.4656 ± 0.2158	0.7509 ± 0.0534
Client 3 accuracy	$\mathbf{0.4591 \pm 0.1131}$	0.3973 ± 0.1333	0.2838 ± 0.1055	0.5845 ± 0.0854
Client 4 accuracy	$\mathbf{0.4751 \pm 0.1241}$	0.5007 ± 0.1303	0.5256 ± 0.1932	0.3507 ± 0.0578
Client 5 accuracy	$\mathbf{0.4013 \pm 0.0430}$	0.4429 ± 0.1195	0.5603 ± 0.1581	0.4627 ± 0.0456

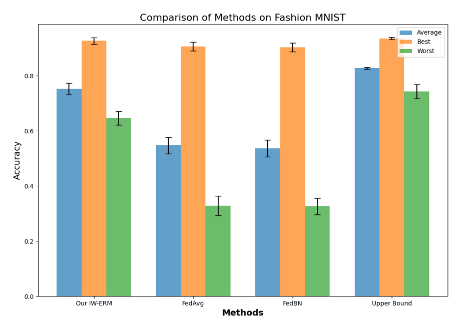

Figure 6: The average, best-client, and worst-client accuracy, along with their standard deviations, are derived from Table 6. Our method exhibits the lowest standard deviation, showcasing the most robust accuracy amongst the compared methods.

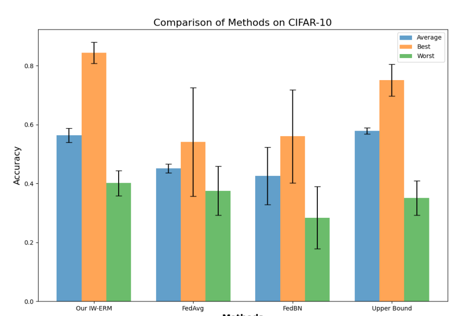

Figure 7: The average, best-client, and worst-client accuracy, along with their standard deviations, are derived from Table 7.

Table 8: Label distribution on Fasion MNIST with 5 clients, with the majority of classes possessing a limited number of training and test images across each client.

		Class									
		0	1	2	3	4	5	6	7	8	9
Client 1	Train	34	34	34	34	34	5862	34	34	34	34
	Test	977	5	5	5	5	5	5	5	5	5
Client 2	Train	34	34	34	34	34	34	5862	34	34	34
	Test	5	977	5	5	5	5	5	5	5	5
Client 3	Train	34	34	34	34	34	34	34	5862	34	34
	Test	5	5	977	5	5	5	5	5	5	5
Client 4	Train	34	34	34	34	34	34	34	34	5862	34
	Test	5	5	5	977	5	5	5	5	5	5
Client 5	Train	34	34	34	34	34	34	34	34	34	5862
	Test	5	5	5	5	977	5	5	5	5	5

Table 9: Label distribution on CIFAR-10 with 5 clients, with the majority of classes possessing a limited number of training and test images across each client.

		Class									
		0	1	2	3	4	5	6	7	8	9
Client 1	Train	34	34	34	34	34	5862	34	34	34	34
	Test	977	5	5	5	5	5	5	5	5	5
Client 2	Train	34	34	34	34	34	34	5862	34	34	34
	Test	5	977	5	5	5	5	5	5	5	5
Client 3	Train	34	34	34	34	34	34	34	5862	34	34
	Test	5	5	977	5	5	5	5	5	5	5
Client 4	Train	34	34	34	34	34	34	34	34	5862	34
	Test	5	5	5	977	5	5	5	5	5	5
Client 5	Train	34	34	34	34	34	34	34	34	34	5862
	Test	5	5	5	5	977	5	5	5	5	5

Table 10: Label distribution on CIFAR-10 with 100 clients, wherein groups of 10 clients share the same distribution and ratios. The majority of classes possess a limited quantity of training and test images on each client.

		Class 0	Class 1	Class 2	Class 3	Class 4
Client 1-10	Train	95/100	5/9	5/9	5/9	5/9
	Test	5/9	5/9	5/9	5/9	5/9
Client 11-20	Train	5/9	95/100	5/9	5/9	5/9
	Test	5/9	5/9	5/9	5/9	5/9
Client 21-30	Train	5/9	5/9	95/100	5/9	5/9
	Test	5/9	5/9	5/9	5/9	5/9
Client 31-40	Train	5/9	5/9	5/9	95/100	5/9
	Test	5/9	5/9	5/9	5/9	5/9
Client 41-50	Train	5/9	5/9	5/9	5/9	95/100
	Test	5/9	5/9	5/9	5/9	5/9
Client 51-60	Train	5/9	5/9	5/9	5/9	5/9
	Test	5/9	5/9	5/9	5/9	95/100
Client 61-70	Train	5/9	5/9	5/9	5/9	5/9
	Test	5/9	5/9	5/9	95/100	5/9
Client 71-80	Train	5/9	5/9	5/9	5/9	5/9
	Test	5/9	5/9	95/100	5/9	5/9
Client 81-90	Train	5/9	5/9	5/9	5/9	5/9
	Test	5/9	95/100	5/9	5/9	5/9
Client 91-100	Train	5/9	5/9	5/9	5/9	5/9
	Test	95/100	5/9	5/9	5/9	5/9

		Class 5	Class 6	Class 7	Class 8	Class 9
Client 1-10	Train	5/9	5/9	5/9	5/9	5/9
	Test	5/9	5/9	5/9	5/9	95/100
Client 11-20	Train	5/9	5/9	5/9	5/9	5/9
	Test	5/9	5/9	5/9	95/100	5/9
Client 21-30	Train	5/9	5/9	5/9	5/9	5/9
	Test	5/9	5/9	95/100	5/9	5/9
Client 31-40	Train	5/9	5/9	5/9	5/9	5/9
	Test	5/9	95/100	5/9	5/9	5/9
Client 41-50	Train	5/9	5/9	5/9	5/9	5/9
	Test	95/100	5/9	5/9	5/9	5/9
Client 51-60	Train	95/100	5/9	5/9	5/9	5/9
	Test	5/9	5/9	5/9	5/9	5/9
Client 61-70	Train	5/9	95/100	5/9	5/9	5/9
	Test	5/9	5/9	5/9	5/9	5/9
Client 71-80	Train	5/9	5/9	95/100	5/9	5/9
	Test	5/9	5/9	5/9	5/9	5/9
Client 81-90	Train	5/9	5/9	5/9	95/100	5/9
	Test	5/9	5/9	5/9	5/9	5/9
Client 91-100	Train	5/9	5/9	5/9	5/9	95/100
	Test	5/9	5/9	5/9	5/9	5/9