7 Appendix

7.1 Polynomial regression examples

7.1.1 Example in Section 3.3

The true underlying function is chosen as $f(x) = 0.5x^3 + 0.3x^2 - 5x + 4$. There are three agents in total, each of whom has 50 data points. The local data points are generated using normal distributions: $x_1 \sim \mathcal{N}(-2, 1)$, $x_2 \sim \mathcal{N}(0, 1)$ and $x_3 \sim \mathcal{N}(2, 1)$. To introduce noise in the labels, each agent adds a normally distributed error term with zero mean and unit variance, i.e. $y_i = f(x_i) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 1)$.

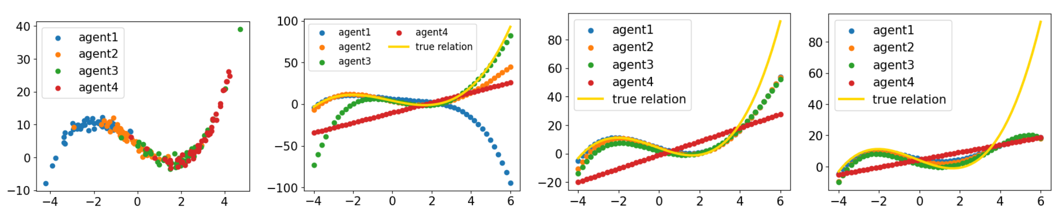
A set of 50 equally spaced data points in the range of -4 to 4, denoted as $\boldsymbol{X_s}$, is used in the analysis. The algorithm is applied using fixed trust weights with 1/3 in each entry and λ is chosen as 1.

7.1.2 Example with strong and weak architectures

The true underlying function is chosen as $f(x) = 0.5x^3 + 0.3x^2 - 5x + 4$. There are four agents in total, each of whom has 50 data points. The local data points are generated using normal distributions: $x_1 \sim \mathcal{N}(-2, 1)$, $x_2 \sim \mathcal{N}(0, 1)$, $x_3 \sim \mathcal{N}(2, 1)$ and $x_4 \sim \mathcal{N}(3, 1)$. To introduce noise in the labels, each agent adds a normally distributed error term with zero mean and unit variance, i.e. $y_i = f(x_i) + \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 1)$.

A set of 50 equally spaced data points in the range of -4 to 6, denoted as $\boldsymbol{X_s}$, is used in the analysis. The algorithm is applied using dynamic trust weights and λ is chosen as 1. For the first three agents, a polynomial model with a maximum degree of four is fit, while for the fourth agent, a polynomial model with a maximum degree of one is fit, signifying a weak node.

We see that after 50 rounds of model training using our proposed algorithm with dynamic trust, agent 4's model is still underfitting due to its limited expressiveness. Agents 1-3 end up agreeing with each other and giving good predictions in the union of their local regions. While with naive trust weights, we see that the strong agents also get influenced in the region where they could perform well, as the underfitted model has stronger impact through the collective pseudo-labeling.



(a) (b) (c) (d)

Figure 7: (a) local data distribution in each agent; (b) local model fit without collaboration; (c) model fits after 50 rounds of our algorithm with dynamic trust update; (d) model fits after 50 rounds with naive trust update

7.2 Proof of Theorem 1

The proof is rooted in the results from the work of Wolfowitz [35], we recommend readers to check the original paper for more detailed references. Note, for the following texts, when we say a matrix $\boldsymbol{W}$ has certain properties, it is equivalent to say a Markov chain induced by transition matrix $\boldsymbol{W}$ has certain peoperties.

Definition B (Irreducible Markov chains)**.** *A Markov chain induced by transition matrix $\boldsymbol{W}$ is irreducible if for all i, j, there exists some t such that $\boldsymbol{W}_{ij}^t > 0$. Equivalently, the graph corresponding to $\boldsymbol{W}$ is strongly connected.*

Definition C (Strongly connected graph)**.** *A graph is said to be strongly connected if every vertex is reachable from every other vertex.*

Definition D (Aperiodic Markov chains)**.** *A Markov chain induced by transition matrix $\boldsymbol{W}$ is aperiodic if every state has a self-loop. By self-loop, we mean that there is a nonzero probability of remaining in that state, i.e. $w_{ii} > 0$ for every i.*

Claim 5. *Given Assumption 2, matrix product of any n elements of $\{\boldsymbol{W}^{(t)}\}$ are SIA (SIA stands for stochastic, irreducible and aperiodic) for $n \geq 1$.*

Proof. According to Assumption (2), all $\boldsymbol{W}^{(t)}$'s are positive, and thus we have any product of $\boldsymbol{W}^{(t)}$'s being positive in each entry, which is equivalent to the graph introduced by the product being fully connected. Being fully connected implies being strongly connected. According to Definitions B C, irreducibility follows.

By the product being positive, we also have its diagonal entries being all positive. According to Definition D, aperiodicity follows.

The product of row-stochastic matrices remains row-stochastic: for $\boldsymbol{A}$ and $\boldsymbol{B}$ row stochastic, we have the product $\boldsymbol{AB}$ remains row-stochastic.

$$\sum_j (\sum_k a_{ik} b_{kj}) = \sum_k a_{ik} (\sum_j b_{kj}) = 1, \ \forall i$$

Thus, we have any product of $\boldsymbol{W}^{(t)}$'s being irreducible, aperiodic and stochastic (SIA). $\square$

Theorem 6 (Rewrite of Wolfowitz [35])**.** *Let $\boldsymbol{A}_1$, ..., $\boldsymbol{A}_k$ be square row stochastic matrices of the same order and any product of the $\boldsymbol{A}$'s (of whatever length) is SIA. When $k \to \infty$, the product of $\boldsymbol{A}_1$, ..., $\boldsymbol{A}_k$ gets reduced to a matrix with identity rows.*

Following Assumptions (1) (2), we have $\psi^{(t)} = \boldsymbol{W}^{(t)} \psi^{(t-1)}$ holds for all $t \geq 1$. From Claim 5, we have any products of $\boldsymbol{W}^{(t)}$'s being SIA. From Theorem 6, we have the product $\boldsymbol{W}^{(t)}\boldsymbol{W}^{(t-1)} \dots \boldsymbol{W}^{(1)}$ gets reduced to a matrix with identical rows when t goes to infinity. That implies, ψ^{∞} has identical rows. The statement is thus proved.

7.3 Proof of Claim 2

Definition E (Row differences)**.** *Define how different the rows of $\boldsymbol{W}$ are by*

$$\delta(\boldsymbol{W}) = \max_j \max_{i_1, i_2} |w_{i_1,j} - w_{i_2,j}| \tag{9}$$

For identical rows, $\delta(\boldsymbol{W}) = 0$

Definition F (Scrambling matrix)**.** *$\boldsymbol{W}$ is a scrambling matrix if*

$$\lambda(\boldsymbol{W}) \coloneqq 1 - \min_{i_1, i_2} \sum_j \min(w_{i_1 j}, w_{i_2 j}) < 1 \tag{10}$$

In plain words, Definition F says that if for every pair of rows i_1 and i_2 in a matrix $\boldsymbol{W}$, there exists a column j (which may depend on i_1 and i_2) such that $w_{i_1 j} > 0$ and $w_{i_2 j} > 0$, then $\boldsymbol{W}$ is a scrambling matrix. It is easy to verify that a positive matrix is always a scrambling matrix.

Lemma 1 (Adaptation of Lemma 2 from Wolfowitz [35])**.** *For any t,*

$$\delta(\boldsymbol{W}^{(t)}\boldsymbol{W}^{(t-1)} \dots \boldsymbol{W}^{(1)}) \leq \prod_{i=1}^{t} \lambda(\boldsymbol{W}^{(i)}) \tag{11}$$

Lemma 1 states that multiplying with scrambling matrices will make the row differences smaller. $tr(\boldsymbol{W}^{(t)}) = \sum_i w_{ii}^{(t)}$ represents the sum of self-confidences of all nodes. As every $\boldsymbol{W}^{(t)}$ is positive from Assumption (2), we have all $\boldsymbol{W}^{(t)}$'s scrambling. Thus, the differences between rows of $\boldsymbol{W}^{(t)}\boldsymbol{W}^{(t-1)}..\boldsymbol{W}^{(1)}$ get smaller when t gets bigger.

As $\boldsymbol{\psi}_i^{(t)} = \sum_j [\boldsymbol{W}^{(t)}\boldsymbol{W}^{(t-1)}..\boldsymbol{W}^{(1)}]_{ij} \boldsymbol{\psi}_j^{(t-1)}$, we have the predictions on $\boldsymbol{X_s}$ given by all nodes get similar over time. According to our calculation of $\boldsymbol{W}^{(t)}$ in Equation (7), which is based on cosine similarity between predictions, it follows that an agent's trust towards the others gets larger over time. That is, $\sum_j w_{ij}^{(t+1)} \geq \sum_j w_{ij}^{(t)}$. Since each row sums up to 1, we have $w_{ii}^{(t+1)} \leq w_{ii}^{(t)}$, for all i.

According to Theorem 1, we have $\boldsymbol{\psi}_i^{(t)} = \boldsymbol{\psi}_j^{(t)}$ as $t \to \infty$, for any i and j. According to the calculation of $\boldsymbol{W}$, we have $\boldsymbol{W}^{(t)}$ with equal entries when t reaches infinity.

7.4 Proof of Proposition 3

Recall stationary distribution ($\boldsymbol{\pi} \in \mathbb{R}^{1\times N}$) of a Markov chain being

$$\lim_{t\to\infty} \boldsymbol{W}^{(t)} \dots \boldsymbol{W}^{(1)} \to [\boldsymbol{\pi}^\top \dots \boldsymbol{\pi}^\top]^\top \tag{12}$$

The proof follows from the construction of Metropolis chains given a stationary distribution. We will first give an example of how Metropolis chains work.

Example 2 (Metropolis chains [27])**.** *Given stationary distribution $\boldsymbol{\pi} = [0.3, 0.3, 0.3, 0.1]$, how could we construct a transition matrix that leads to the stationary distribution?*

Suppose $\boldsymbol{\Phi}$ is a symmetric matrix, one can construct a M5 etropolis chain $\boldsymbol{P}$ as follows:

$$p(x,y) = \begin{cases} \phi(x,y)\min\left(1, \frac{\pi(y)}{\pi(x)}\right) & y \neq x \\ 1 - \sum_{z\neq x} \phi(x,z)\min\left(1, \frac{\pi(z)}{\pi(x)}\right) & y = x \end{cases} \tag{13}$$

Choose $\boldsymbol{\Phi} = \begin{bmatrix} 1/3 & 1/4 & 1/4 & 1/6 \\ 1/4 & 1/3 & 1/4 & 1/6 \\ 1/4 & 1/4 & 1/3 & 1/6 \\ 1/6 & 1/6 & 1/6 & 1/2 \end{bmatrix}$, *we could get* $\boldsymbol{P} = \begin{bmatrix} 4/9 & 1/4 & 1/4 & 1/18 \\ 1/4 & 4/9 & 1/4 & 1/18 \\ 1/4 & 1/4 & 4/9 & 1/18 \\ 1/6 & 1/6 & 1/6 & 1/2 \end{bmatrix}$. *It can be verified that $\boldsymbol{\pi}$ is the stationary distribution of Markov chain with transition matrix $\boldsymbol{P}$. If $\boldsymbol{\Phi}$ is not symmetric, we modify $\frac{\pi(y)}{\pi(x)}$ to $\frac{\pi(y)}{\pi(x)}\frac{\phi(y,x)}{\phi(x,y)}$, and the results remain unchanged.*

Following Example 2, choose $\boldsymbol{\Phi}$ to be any self-confident doubly stochastic matrix. For all x, choose $\boldsymbol{P}$ as calculated from (13), we have

$$p(x,x) = 1 - \sum_{z\neq x} \phi(x,z)\min\left(1, \frac{\pi(z)}{\pi(x)}\right) \geq 1 - \sum_{z\neq x}\phi(x,z) = \phi(x,x) \tag{14}$$

we see that probability distribution among each row gets more concentrated on the diagonal entries in $\boldsymbol{P}$ than $\boldsymbol{\Phi}$. As $\boldsymbol{\Phi}$ already has high diagonal values, the claim follows.

7.5 Proof of Proposition 4

Proposition 4 states sufficient conditions for $\boldsymbol{W}^{(t)}$'s to have such that a low quality node b is assigned lowest importance in $\boldsymbol{\pi}$, i.e. $\pi_b = \min_i \pi_i$.

From Equation (12), $\boldsymbol{\pi}$ comes from the product of trust matrices. We start from a product of two such matrices.

Proposition 7. *For row-stochastic and positive matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, and $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}$, if in both $\boldsymbol{A}$ and $\boldsymbol{B}$,*

(1) j-th column has the lowest column sum,

(2) (i,j)-th entry being the lowest value in i-th row for $i \neq j$,

then we have j-th column remains the the lowest column sum in matrix $\boldsymbol{C}$ and (i,j)-th entry being the lowest value in i-th row of $\boldsymbol{C}$ for $i \neq j$,

Proof. Let $\boldsymbol{C} = \boldsymbol{A}\boldsymbol{B}$, the column sum of column j of $\boldsymbol{C}$ can be expressed as:

$$\begin{aligned} \sum_i c_{ij} &= \sum_i \sum_k a_{ik} b_{kj} \\ &= \sum_k (\sum_i a_{ik}) b_{kj} \end{aligned} \tag{15}$$

for $t \neq j$, the column sum of $\boldsymbol{C}$ is

$$\begin{aligned} \sum_i c_{it} &= \sum_i \sum_k a_{ik} b_{kt} \\ &= \sum_k (\sum_i a_{ik}) b_{kt} \end{aligned} \tag{16}$$

We first show that j-th column remains the lowest column sum in $\boldsymbol{C}$. For $t \neq j$:

$$\begin{aligned}
\sum_i c_{it} - \sum_i c_{ij} &= \sum_k (\sum_i a_{ik})(b_{kt} - b_{kj}) \\
&= \sum_{k \neq j} (\sum_i a_{ik})(b_{kt} - b_{kj}) + (\sum_i a_{ij})(b_{jt} - b_{jj}) \\
&\overset{(i)}{>} \sum_{k \neq j} (\sum_i a_{ij})(b_{kt} - b_{kj}) + (\sum_i a_{ij})(b_{jt} - b_{jj}) \\
&= (\sum_i a_{ij}) \left(\sum_{k \neq j} (b_{kt} - b_{kj}) + (b_{jt} - b_{jj}) \right) \\
&= \sum_i a_{ij} \left(\sum_k b_{kt} - \sum_k b_{kj} \right) \\
&\overset{(ii)}{>} 0
\end{aligned}$$

(i) holds because for $k \neq j$, $b_{kt} - b_{kj} > 0$ and $\sum_i a_{ij} < \sum_i a_{ik}$
(ii) holds because the j-th column has the lowest column sum in B

We then show that (i, j)-th entry remains the lowest value in i-th row of $\boldsymbol{C}$ for $i \neq j$. For $t \neq j$, we have

$$\begin{aligned}
c_{it} - c_{ij} =& \sum_k a_{ik} b_{kt} - \sum_k a_{ik} b_{kj} \\
=& \sum_{k \neq j} a_{ik}(b_{kt} - b_{kj}) + a_{ij}(b_{jt} - b_{jj}) \\
\overset{(iii)}{>}& \sum_{k \neq j} a_{ij}(b_{kt} - b_{kj}) + a_{ij}(b_{jt} - b_{jj}) \\
=& a_{ij} \left(\sum_{k \neq j} (b_{kt} - b_{kj}) + (b_{jt} - b_{jj}) \right) \\
=& a_{ij} \left(\sum_k b_{kt} - \sum_k b_{kj} \right) \\
\overset{(iv)}{>}& 0
\end{aligned} \tag{17}$$

(iii) holds since $b_{kt} - b_{kj} > 0$ and $a_{ik} > a_{ij}$ for $i, k \neq j$.

(iv) holds because $\sum_k b_{kt} > \sum_k b_{kj}$ □

For time-inhomogenous trust matrix, Assumptions 1 2 ensure the Markov chain update: $\boldsymbol{\psi}_i^{(t)} = \sum_j w_{ij}^{(t)} \boldsymbol{\psi}_j^{(t-1)}$, which is followed by consensus as proven in Theorem 1. We see that b-th column remains the lowest column sum in the product $\boldsymbol{W}^{(\tau)} \boldsymbol{W}^{(\tau-1)} ... \boldsymbol{W}^{(1)}$, by iteratively applying Proposition 7. For $t \geq \tau$, $\boldsymbol{W}^{(t)} = \mathbf{1}\mathbf{1}^\top \frac{1}{N}$, the multiplication does not change the order of the column sum. Thus, the b-th column will remain to be the smallest column in the consensus. For the time-homogenous case, we can simply treat τ as ∞, as long as $\boldsymbol{W}$ remains to have the above-mentioned properties, the results will still hold. Thus, Proposition 4 is proved.

Extend to more than one node with low-quality data. For more than one low-quality node, what are the desired properties (sufficient conditions) for the transition (trust) matrices to have? It turns out that apart from the two conditions in a single low-quality node case, we need an extra assumption.

Proposition 8. *Given Assumptions 1 2 and that all agents are over-parameterized, let $\mathcal{R}$ be the set of indices of regular nodes, and $\mathcal{B}$ be the set of indices of low-quality nodes, if for $t \leq \tau$, $\boldsymbol{W}^{(t)}$ satisfies the following conditions:*

(1) any regular node's column sum is larger than any low-quality node's: $\min_{r\in\mathcal{R}}\sum_i w_{ir}^{(t)} > \max_{b\in\mathcal{B}}\sum_i w_{ib}^{(t)}$*;*

(2) the gap between the sum of trust from regular nodes towards any regular node r and low-quality node b is larger than the gap between low-quality node b's self-confidence and its trust towards the regular node: $\sum_{n\in\mathcal{R}}(w_{nr}^{(t)} - w_{nb}^{(t)}) > (w_{bb}^{(t)} - w_{br}^{(t)})$*,*

(3) any node's trust towards a regular node is bigger or equal than its trust towards a low-quality node other than itself: for any $r \in \mathcal{R}$ and any $b \in \mathcal{B}$, we have $w_{nr}^{(t)} \geq w_{nb}^{(t)}$ holds as long as $n \neq b$.

And after $t > \tau$, $\boldsymbol{W}^{(t)} = \mathbf{1}\mathbf{1}^\top \frac{1}{N}$. then we have nodes in $\mathcal{B}$ has the lower importance in the consensus than nodes in $\mathcal{R}$.

Proof. First, let us look at the multiplication of two such matrices when $1 < t < \tau$, for any $r \in \mathcal{R}$ and $b \in \mathcal{B}$, we have conditions (1)(2)(3) remain to be true for the product $\boldsymbol{W}^{(t)}\boldsymbol{W}^{(t-1)}$. We will verify them one by one in the following part:

Verification of condition (1): any regular node's column sum is larger than any low-quality node's in $\boldsymbol{W}^{(t)}\boldsymbol{W}^{(t-1)}$. For any $r \in \mathcal{R}$ and any $b \in \mathcal{B}$, we have

$$
\begin{aligned}
&\sum_i\sum_n w_{in}^{(t)} w_{nr}^{(t-1)} - \sum_i\sum_n w_{in}^{(t)} w_{nb}^{(t-1)} \\
=&\sum_n (\sum_i w_{in}^{(t)}) \left(w_{nr}^{(t-1)} - w_{nb}^{(t-1)}\right) \\
=&\sum_{n\in\mathcal{R}} (\sum_i w_{in}^{(t)}) \left(w_{nr}^{(t-1)} - w_{nb}^{(t-1)}\right) + \sum_{n\in\mathcal{B}\setminus\{b\}} (\sum_i w_{in}^{(t)}) \left(w_{nr}^{(t-1)} - w_{nb}^{(t-1)}\right) \\
&+(\sum_i w_{ib}^{(t)}) \left(w_{br}^{(t-1)} - w_{bb}^{(t-1)}\right) \\
\overset{(i)}{>}&\sum_{n\in\mathcal{R}} (\sum_i w_{ib}^{(t)}) \left(w_{nr}^{(t-1)} - w_{nb}^{(t-1)}\right) + \sum_i w_{ib}^{(t)} \left(w_{br}^{(t-1)} - w_{bb}^{(t-1)}\right) \\
&+ \sum_{n\in\mathcal{B}\setminus\{b\}} (\sum_i w_{in}^{(t)}) \left(w_{nr}^{(t-1)} - w_{nb}^{(t-1)}\right) \\
=&(\sum_i w_{ib}^{(t)}) \left(\sum_{n\in\mathcal{R}} w_{nr}^{(t-1)} - \sum_{n\in\mathcal{R}} w_{nb}^{(t-1)} + w_{br}^{(t-1)} - w_{bb}^{(t-1)}\right) \\
&+ \sum_{n\in\mathcal{B}\setminus\{b\}} (\sum_i w_{in}^{(t)}) \left(w_{nr}^{(t-1)} - w_{nb}^{(t-1)}\right) \\
\overset{(ii)}{>}&0
\end{aligned}
$$

(i) holds because $\sum_i w_{in}^{(t)}$ for any $n \in \mathcal{R}$ is larger than $\sum_i w_{ib}^{(t)}$ for any $b \in \mathcal{B}$, which follows from condition (1), and $w_{nr}^{(t)} - w_{nb}^{(t)} > 0$, which follows from condition (3).

(ii) holds following the conditions (2) and (3). From (2), $\sum_{n\in\mathcal{R}} w_{nr}^{(t-1)} - \sum_{n\in\mathcal{R}} w_{nb}^{(t-1)} + w_{br}^{(t-1)} - w_{bb}^{(t-1)} > 0$, and from (3), $w_{nr}^{(t-1)} \geq w_{nb}^{(t-1)}$ for $n \neq b$

Verification of condition (2):

$$
\begin{aligned}
&\sum_{n\in\mathcal{R}} \left(\sum_p w_{np}^{(t)} w_{pr}^{(t-1)} - \sum_p w_{np}^{(t)} w_{pb}^{(t-1)}\right) - \left(\sum_p w_{bp}^{(t)} w_{pb}^{(t-1)} - \sum_p w_{bp}^{(t)} w_{pr}^{(t-1)}\right) \\
=&\sum_p \left(\sum_{n\in\mathcal{R}} w_{np}^{(t)} + w_{bp}^{(t)}\right) w_{pr}^{(t-1)} - \sum_p \left(\sum_{n\in\mathcal{R}} w_{np}^{(t)} + w_{bp}^{(t)}\right) w_{pb}^{(t-1)}
\end{aligned}
$$

$$
\begin{aligned}
&= \sum_{p} \left(\sum_{n \in \mathcal{R}} w_{np}^{(t)} + w_{bp}^{(t)} \right) \left(w_{pr}^{(t-1)} - w_{pb}^{(t-1)} \right) \\
&= \sum_{p \in \mathcal{R}} \left(\sum_{n \in \mathcal{R}} w_{np}^{(t)} + w_{bp}^{(t)} \right) \left(w_{pr}^{(t-1)} - w_{pb}^{(t-1)} \right) + \sum_{p \in \mathcal{B} \setminus \{b\}} \left(\sum_{n \in \mathcal{R}} w_{np}^{(t)} + w_{bp}^{(t)} \right) \left(w_{pr}^{(t-1)} - w_{pb}^{(t-1)} \right) \\
&+ \left(\sum_{n \in \mathcal{R}} w_{nb}^{(t)} + w_{bb}^{(t)} \right) \left(w_{br}^{(t-1)} - w_{bb}^{(t-1)} \right) \\
&\overset{(iii)}{\geq} \sum_{p \in \mathcal{R}} \left(\sum_{n \in \mathcal{R}} w_{nb}^{(t)} + w_{bb}^{(t)} \right) \left(w_{pr}^{(t-1)} - w_{pb}^{(t-1)} \right) + \left(\sum_{n \in \mathcal{R}} w_{nb}^{(t)} + w_{bb}^{(t)} \right) \left(w_{br}^{(t-1)} - w_{bb}^{(t-1)} \right) \\
&+ \sum_{p \in \mathcal{B} \setminus \{b\}} \left(\sum_{n \in \mathcal{R}} w_{np}^{(t)} + w_{bp}^{(t)} \right) \left(w_{pr}^{(t-1)} - w_{pb}^{(t-1)} \right) \\
&= \left(\sum_{n \in \mathcal{R}} w_{nb}^{(t)} + w_{bb}^{(t)} \right) \left(\sum_{p \in \mathcal{R}} w_{pr}^{(t-1)} - \sum_{p \in \mathcal{R}} w_{pb}^{(t-1)} + w_{br}^{(t-1)} - w_{bb}^{(t-1)} \right) \\
&+ \sum_{p \in \mathcal{B} \setminus \{b\}} \left(\sum_{n \in \mathcal{R}} w_{np}^{(t)} + w_{bp}^{(t)} \right) \left(w_{pr}^{(t-1)} - w_{pb}^{(t-1)} \right) \\
&\overset{(iv)}{\geq} 0
\end{aligned}
$$

(iii) holds because for p a regular node, we have $\sum_{n \in \mathcal{R}} w_{np}^{(t)} + w_{bp}^{(t)} > \sum_{n \in \mathcal{R}} w_{nb}^{(t)} + w_{bb}^{(t)}$, which follows from condition (2), and $w_{pr}^{(t-1)} - w_{pb}^{(t-1)} \geq 0$ for $p \neq b$, following from condition (3).
(iv) holds because of conditions (2) and (3).

Verification of (3): for $n \neq b$, we want to show the trust towards a regular node r is bigger than towards a low-quality node b, that is $\sum_p w_{np}^{(t)} w_{pr}^{(t)} > \sum_p w_{np}^{(t)} w_{pb}^{(t)}$

$$
\begin{aligned}
& \sum_{p} w_{np}^{(t)} w_{pr}^{(t)} - \sum_{p} w_{np}^{(t)} w_{pb}^{(t)} \\
&= \sum_{p \in \mathcal{R}} w_{np}^{(t)} \left(w_{pr}^{(t)} - w_{pb}^{(t)} \right) + \sum_{p \in \mathcal{B} \setminus \{b\}} w_{np}^{(t)} \left(w_{pr}^{(t)} - w_{pb}^{(t)} \right) + w_{nb}^{(t)} \left(w_{br}^{(t)} - w_{bb}^{(t)} \right) \\
&\overset{(v)}{\geq} \sum_{p \in \mathcal{R}} w_{nb}^{(t)} \left(w_{pr}^{(t)} - w_{pb}^{(t)} \right) + w_{nb}^{(t)} \left(w_{br}^{(t)} - w_{bb}^{(t)} \right) + \sum_{p \in \mathcal{B} \setminus \{b\}} w_{np}^{(t)} \left(w_{pr}^{(t)} - w_{pb}^{(t)} \right) \\
&= w_{nb}^{(t)} \left(\sum_{p \in \mathcal{R}} w_{pr}^{(t)} - \sum_{p \in \mathcal{R}} w_{pb}^{(t)} + w_{br}^{(t)} - w_{bb}^{(t)} \right) + + \sum_{p \in \mathcal{B} \setminus \{b\}} w_{np}^{(t)} \left(w_{pr}^{(t)} - w_{pb}^{(t)} \right) \\
&\overset{(vi)}{\geq} 0
\end{aligned}
\tag{18}
$$

(v) holds because for $n \neq b$, we have $w_{np}^{(t)} \geq w_{nb}^{(t)}$, following from condition (3), and $w_{pr}^{(t)} - w_{pb}^{(t)} \geq 0$ for $p \neq b$.

(vi) holds following from conditions (2) and (3).

It follows that in the product $\boldsymbol{W}^{(\tau)} \boldsymbol{W}^{(\tau-1)} \ldots \boldsymbol{W}^{(1)}$, a low-quality node will still have a lower column sum than any regular node. Because conditions (1)(2)(3) holds for any product of $\boldsymbol{W}^{(t)}$'s as long as each of the $\boldsymbol{W}^{(t)}$ share the conditions listed by (1)(2)(3).

After $t > \tau$, multiplying with a naive weight matrix does not change the column sum order, we will have all low-quality nodes have lower importance in the consensus than the regular nodes.

□

7.6 Reasoning for confidence upweighting block

In this section, we provide our intuition of adding such a confidence weighting block in Equation (7).

$\mathbf{\Phi}^{(t)}$ is a row-normalized pairwise cosine similarity matrix, with (i, j)-th entry before row normalization as

$$\frac{1}{n_S}\sum_{\boldsymbol{x}'\in\boldsymbol{X_s}}\frac{\left\langle \boldsymbol{f}_{\boldsymbol{\theta}_i^{(t-1)}}(\boldsymbol{x}'), \boldsymbol{f}_{\boldsymbol{\theta}_j^{(t-1)}}(\boldsymbol{x}')\right\rangle}{\|\boldsymbol{f}_{\boldsymbol{\theta}_i^{(t-1)}}(\boldsymbol{x}')\|_2\|\boldsymbol{f}_{\boldsymbol{\theta}_j^{(t-1)}}(\boldsymbol{x}')\|_2} \tag{19}$$

After adding a confidence weighting block, we have $\boldsymbol{W}^{(t)}$ with (i, j)-th entry before row normalization as

$$\frac{1}{n_S}\sum_{\boldsymbol{x}'\in\boldsymbol{X_s}}\frac{1}{\mathcal{H}(\boldsymbol{f}_{\boldsymbol{\theta}_i^{(t-1)}}(\boldsymbol{x}'))}\frac{\left\langle \boldsymbol{f}_{\boldsymbol{\theta}_i^{(t-1)}}(\boldsymbol{x}'), \boldsymbol{f}_{\boldsymbol{\theta}_j^{(t-1)}}(\boldsymbol{x}')\right\rangle}{\|\boldsymbol{f}_{\boldsymbol{\theta}_i^{(t-1)}}(\boldsymbol{x}')\|_2\|\boldsymbol{f}_{\boldsymbol{\theta}_j^{(t-1)}}(\boldsymbol{x}')\|_2} \tag{20}$$

We want to show that the weighting scheme down-weights the a regular node i's trust towards a low-quality node b, that is

$$\phi_{ib}^{(t)} > w_{ib}^{(t)}$$

As the comparison is made with respect to the same time step t, we drop the t notation from now on. Let $\{a_0, .., a_{N-1}\}$ be the cosine similarity between a regular agent i and others inside agent i's confident region, and $\{b_0, .., b_{N-1}\}$ be the cosine similarity between i and others outside agent i's confident region. By confident region, we mean region with low entropy in class probabilities, i.e. the model is more sure about the prediction. Further, we make the following assumptions:

(1) for $\boldsymbol{x}'$ in agent i's confident region, we have low entropy of predicted class probabilities: $\mathcal{H}(\boldsymbol{f}_{\boldsymbol{\theta}_i^{(t-1)}}(\boldsymbol{x}')) = 1/c_1$ with $c_1 > 1$, while for $\boldsymbol{x}'$ outside agent i's confident region, we have $\mathcal{H}(\boldsymbol{f}_{\boldsymbol{\theta}_i^{(t-1)}}(\boldsymbol{x}')) = 1/c_2$ with $c_2 < 1$

(2) inside a regular node i's confident region, i has a better judgment of the alignment score produced by cosine similarity, such that the cosine similarity with low quality b is weighted lower inside:

$$\frac{a_b}{\sum_i a_i} < \frac{b_b}{\sum_i b_i} \tag{21}$$

to claim $w_{ib} < \phi_{ib}$, we need to show

$$\frac{c_1 a_b + c_2 b_b}{\sum_i (c_1 a_i + c_2 b_i)} < \frac{a_b + b_b}{\sum_i (a_i + b_i)} \tag{22}$$

Proof. Re-arrange Equation 21, we get

$$b_b \sum_i a_i > a_b \sum_i b_i \tag{23}$$

Multiply with $c_2 - c_1$ on both sides, we have

$$(c_2 - c_1) b_b \sum_i a_i < (c_2 - c_1) a_b \sum_i b_i \tag{24}$$

$$c_2 b_b \sum_i a_i + c_1 a_b \sum_i b_i < c_1 b_b \sum_i a_i + c_2 a_b \sum_i b_i \tag{25}$$

Now add $c_1 a_b \sum_i b_i + c_2 b_b \sum_i b_i$ to both sides, we have

$$\begin{aligned} c_1 a_b \sum_i a_i + c_2 b_b \sum_i a_i + c_1 a_b \sum_i b_i + c_2 b_b \sum_i b_i < \\ c_1 a_b \sum_i a_i + c_1 b_b \sum_i a_i + c_2 a_b \sum_i b_i + c_2 b_b \sum_i b_i \end{aligned} \tag{26}$$

Combining the terms we have

$$(c_1 a_b + c_2 b_b)\left(\sum_i (a_i + b_i)\right) < \left(\sum_i (c_1 a_i + c_2 b_i)\right)(a_b + b_b) \tag{27}$$

following which, we directly have

$$\frac{c_1 a_b + c_2 b_b}{\sum_i (c_1 a_i + c_2 b_i)} < \frac{a_b + b_b}{\sum_i (a_i + b_i)} \tag{28}$$

$\square$

7.7 Complementary details

7.7.1 Details regarding model training

All the model training was done using a single GPU (NVIDIA Tesla V100). For each local iteration, we load local data and shared unlabeled data with batch size 64 and 256 separately. We empirically observed that a larger batch size for unlabeled data is necessary for the training to work well. The optimizer used is Adam with a learning rate 5e-3. For Cifar10 and Cifar100, as the base model is not pretrained, we do 50 global rounds with 5 local training epochs for each agent per global round. For Fed-ISIC-2019 dataset, as the base model is pretrained EfficientNet, we do 20 global rounds. For the first 5 global rounds, we set $\lambda = 0$ to arrive at good local models, such that every agent can evaluate trust more fairly. After that, λ is fixed as 0.5. *Dynamic* trust is computed after each global round, while *static* trust denotes the utilization of the initially calculated trust value throughout the whole experiment.

For Cifar10 and Cifar100, we use 5% of the whole dataset to constitute $\boldsymbol{X_s}$, where each class has equal representation. For the rest, we spread them into 10 clients using Dirichlet distribution with $\alpha = 1$. For Fed-ISIC-2019 dataset, we follow the original splits as in du Terrail et al. [33], and we let each client contribute 50 data samples to constitute $\boldsymbol{X_s}$.

We employ a fixed λ for all our experiments. To select λ, we randomly sample 10% of the full Cifar10 dataset, which we then split into local training data (95%) and $\boldsymbol{X_s}$ (5%). The local training data is then spread into 10 clients using Dirichlet distribution with $\alpha = 1$. The test global accuracy and value of λ is plotted out in Figure 8. We thus choose $\lambda = 0.5$ for all our experiments, and it is always able to give stable performances according to our experiments.

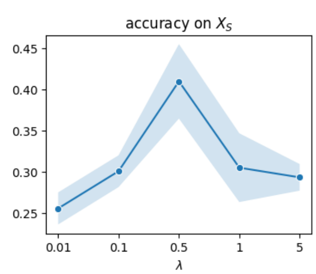


Figure 8: λ versus algorithm performance

7.7.2 Limitations of the work

The main limitation of this work is the requirement of an extra shared unlabelled dataset, like in other knowledge distillation-based decentralized learning works. Moreover, each agent needs to calculate their trust towards all other nodes locally. The extra computational complexity is $\mathcal{O}(N \times n_S \times C)$, where N stands for the number of agents, n_S stands for the size of the shared dataset and C denotes the number of classes. The computation can be heavy if the number of clients gets large. But as we focus on cross-silo setting, N usually does not tend to be a big number.