POLITECNICO DI TORINO
Repository ISTITUZIONALE

Scalable Decentralized Algorithms for Online Personalized Mean Estimation

Original
Scalable Decentralized Algorithms for Online Personalized Mean Estimation / Galante, F., Neglia, G., Leonardi, E.. - ELETTRONICO. - 39:(2025), pp. 16699-16707. (The 39th Annual AAAI Conference on Artificial Intelligence Philadelphia (USA) February 25 – March 4, 2025) [10.1609/aaai.v39i16.33835].

Availability:
This version is available at: 11583/2995225 since: 2025-01-16T11:07:36Z

Publisher:
AAAI

Published
DOI:10.1609/aaai.v39i16.33835

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

28 November 2026

Scalable Decentralized Algorithms for Online Personalized Mean Estimation

Franco Galante[1], **Giovanni Neglia**[2], **Emilio Leonardi**[1]

[1]Politecnico di Torino
[2]INRIA
franco.galante@polito.it, giovanni.neglia@inria.fr, emilio.leonardi@polito.it

Abstract

In numerous settings, agents lack sufficient data to learn a model directly. Collaborating with other agents may help, but introduces a bias-variance trade-off when local data distributions differ. A key challenge is for each agent to identify clients with similar distributions while learning the model, a problem that remains largely unresolved. This study focuses on a particular instance of the overarching problem, where each agent collects samples from a real-valued distribution over time to estimate its mean. Existing algorithms face impractical per-agent space and time complexities (linear in the number of agents $|\mathcal{A}|$). To address scalability challenges, we propose a framework where agents self-organize into a graph, allowing each agent to communicate with only a selected number of peers r. We propose two collaborative mean estimation algorithms: one employs a consensus-based approach, while the other uses a message-passing scheme, with complexity $\mathcal{O}(r)$ and $\mathcal{O}(r \cdot \log|\mathcal{A}|)$, respectively. We establish conditions for both algorithms to yield asymptotically optimal estimates and we provide a theoretical characterization of their performance.

Code — https://github.com/Franco-Galante/scalable-decentralized-algorithms-AAAI25

1 Introduction

Users' devices have become increasingly sophisticated and generate vast amounts of data. This wealth of data has enabled the development of accurate and complex models. However, it has also introduced challenges related to security, privacy, real-time processing, and resource management. In response, Federated Learning (FL) has emerged as a key privacy-preserving approach for collaborative model training (Kairouz et al. 2021; Li et al. 2020). While traditional FL methods aim to develop a single model for all clients, the statistical diversity of clients' datasets has led to the development of *personalized* models, designed to better align with the data distributions of individual clients (e.g., Ghosh et al. 2020; Fallah, Mokhtari, and Ozdaglar 2020; Li et al. 2021; Marfoq et al. 2021; Ding and Wang 2022).

Many personalized FL strategies group clients into clusters and then tailor a model for each cluster (e.g., Ghosh et al. 2020; Sattler, Müller, and Samek 2021; Ding and Wang 2022). Ideally, clustering would group clients with similar local optimal models. However, since the optimal models are unknown a priori, model learning and cluster identification become deeply interconnected tasks. Various studies have suggested empirical measures of similarity as a workaround (e.g., Ghosh et al. 2020; Sattler, Müller, and Samek 2021), while others rely on presumed knowledge of distances across data distributions (e.g. Ding and Wang 2022; Even, Massoulié, and Scaman 2022). Nonetheless, accurately estimating these distances, especially within an FL framework where clients may possess limited data, proves to be particularly challenging. These estimation difficulties are well documented in the literature—e.g., by Even, Massoulié, and Scaman (2022) [Sec. 6]—highlighting the problem of identifying similar clients for collaborative model learning as a significant yet unresolved issue.

In this paper, we focus on a fundamental aspect of the broader challenge: estimating the mean of an $\mathbb{R}^K$-valued distribution. This problem is often regarded as the archetypal federated learning problem (Dorner et al. 2024; Tsoy et al. 2024; Grimberg et al. 2021), but it also holds significant practical relevance across various fields, such as smart agriculture, grid management, and healthcare, where multiple sensors collect private, noisy data on identical or related variables (Adi et al. 2020).

We consider an online, decentralized scenario where, at each time slot, clients receive new samples and exchange information with a limited number of peers. To the best of our knowledge, the state-of-the-art method in this setting is the Collaborative Mean Estimation algorithm (ColME) by Asadi et al. (2022). Unfortunately, ColME faces scalability issues in large systems, as both its per-agent space and time complexities are linear in the number of clients $|\mathcal{A}|$. Moreover, in its current form, ColME is applicable only to scalar mean estimation problems ($K = 1$) and its convergence guarantees only hold for sub-Gaussian data distributions.

We extend the methodology proposed by Asadi et al. (2022) to accommodate multidimensional data drawn from the broader class of distributions with bounded fourth moment. To address ColME's scalability challenges, we propose that clients self-organize into a network where each client communicates with at most r neighbors. Over time, this set of neighbors is pruned as clients progressively ex-

Copyright © 2025, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

clude the less similar ones. In this framework, we introduce two collaborative mean estimation algorithms: one based on consensus and the other on a message-passing scheme. The complexities of these algorithms are $\mathcal{O}(r)$ and $\mathcal{O}(r \cdot \log |\mathcal{A}|)$, respectively. We demonstrate that, despite each client exchanging information with only $r \ll |\mathcal{A}|$ neighbors, it is possible to achieve a convergence speedup for mean estimates by a factor of $\Omega(|\mathcal{A}|^{1/2-\phi})$, where ϕ can be made arbitrarily close to 0.

Lastly, we conduct preliminary experiments demonstrating how our algorithms can be adapted to federatedly learn more general machine learning models.

2 Related Work

For an overview of personalized federated learning, see the recent survey by Tan et al. (2023). Here, we highlight only the most relevant approaches related to this paper.

Ghosh et al. (2020) and Sattler, Müller, and Samek (2021) were the first to propose clustered FL algorithms, which divide the clients based on the similarity of their data distributions. Similarity is empirically evaluated by the Euclidean distance between local models and by the cosine similarity of their updates. Ding and Wang (2022) study more sophisticated clustering algorithms assuming that clients can efficiently estimate some specific (pseudo)-distances across local distributions (i.e., the integral probability metrics).

Beaussart et al. (2021), Chayti et al. (2022), Grimberg et al. (2021), and Even, Massoulié, and Scaman (2022) consider decentralized approaches, which allow each client to learn a personal model relying on a specific convex combination of information (gradients) from other clients. In particular, Even, Massoulié, and Scaman (2022) prove that collaboration can at most speed up the convergence time linearly in the number of similar agents and provide algorithms, which, under a priori knowledge of pairwise client distributions' distances, achieve such speedup. The authors recognize the complexity of estimating these distances and provide practical estimation algorithms for linear regression problems, which asymptotically achieve the same speedup, scaling the number of clients but maintaining the number of clusters fixed. The generalization properties of personalized models obtained by convex combinations of clients' models are studied in Mansour et al. (2020), while Donahue and Kleinberg (2021) look at the problem through the lens of game theory. The work most similar to ours is Asadi et al. (2022), which we describe in detail in the next section.

3 Model and Background

Table 1 lists the most important symbols used throughout the paper. Superscripts are added to variables to indicate whether they pertain to C-ColME (C), B-ColME (B), or both approaches (D).

We consider a set $\mathcal{A}$ of agents (computational units). At each time instant t, an agent $a \in \mathcal{A} = \{1, 2, \cdots, |\mathcal{A}|\}$ generates a new sample $\boldsymbol{x}_a^t \in \mathbb{R}^K$, with $K \in \mathbb{N}$, drawn i.i.d. from a distribution D_a with expected value $\boldsymbol{\mu}_a = \mathbb{E}[\boldsymbol{x}_a^t]$. Expected values are not necessarily distinct across agents. Indeed, given two agents a and a', and a norm $||\cdot||$ in $\mathbb{R}^K$, we denote the gap between the agents' *true* means by $\Delta_{a,a'} := ||\boldsymbol{\mu}_a - \boldsymbol{\mu}_{a'}||$. Let $\mathcal{C}_a$ be the group of agents with the same *true* mean as a, (i.e., those for which $\Delta_{a,a'} = 0$). In the following, we will refer to $\mathcal{C}_a$ as the 'similarity class of a.'

The goal of each agent $a \in \mathcal{A}$ is to estimate its mean $\boldsymbol{\mu}_a$. To this end, at each time t the agent can compute its *local* mean estimate $\bar{\boldsymbol{x}}_{a,a}^t = \frac{1}{t}\sum_{t'=1}^{t} \boldsymbol{x}_a^{t'}$ over the t available samples. Additionally, the agent can obtain a more accurate estimate by leveraging information from other agents in $\mathcal{A}$, provided it can identify those who share the same true mean.

For the scalar case, (i.e., when $K = 1$) Asadi et al. (2022) proposed ColME as a collaborative algorithm for mean estimation. It relies on two key steps, executed concurrently: i) the identification of the similarity classes; ii) the collaboration with agents *believed to belong* to the same class to improve the local estimate. To ensure direct comparability with the results in (Asadi et al. 2022) and simplify notation, in what follows, we focus on the scalar case. Readers interested in the general case can find the analysis in Appendix C. All the appendices referred to in this work are accessible at (Galante, Neglia, and Leonardi 2024).

Symbol	Description
$\mathcal{A}$	Set of agents (computational units)
D_a	Distribution of agent $a \in \mathcal{A}$
$\boldsymbol{\mu}_a$	True mean of distribution D_a, i.e., $\boldsymbol{\mu}_a = \mathbb{E}[\boldsymbol{x}_a^t]$
$\bar{\boldsymbol{x}}_a^t$	Vectorial sample $\bar{\boldsymbol{x}}_a^t \in \mathbb{R}^K$ drawn from D_a
$\Delta_{a,a'}$	Gap between agents a and a' true means
$\mathcal{C}_a$	Similarity class of a, $\mathcal{C}_a = \{a' \in \mathcal{A} \vert \Delta_{a,a'} = 0\}$
$\mathcal{C}_a^t$	Estimated similarity class of a at time t
$\bar{x}_{a,a}^t$	Local mean estimate of agent a at t
$\bar{x}_{a,a'}^t$	Local mean estimate of agent a' by agent a at t
$n_{a,a'}^t$	Number of samples used to compute $\bar{x}_{a,a'}^t$ at t
$\hat{\mu}_a^t$	Collaborative mean estimate at t
$\beta_\gamma(n)$	Width of the confidence interval
$d_\gamma^t(a,a')$	Optimistic distance between agents a and a'
ζ_a	Time to identify same-class neighbors w.h.p.
τ_a	Time to obtain (ϵ, δ) convergence
$\mathcal{G}$	Collaborative graph $\mathcal{G}(\mathcal{A}, \mathcal{E})$
$\mathcal{G}^t$	Pruned collaborative graph $\mathcal{G}^t(\mathcal{A}, \mathcal{E}^t)$
$\mathcal{N}_a$	Neighborhood of agent a (or up to distance d: $\mathcal{N}_a^d$)
r	Upper bound on a's neighborhood size $\vert\mathcal{N}_a\vert$
$\mathcal{CC}_a$	Agents in the connected component of the subgraph induced by agents in $\mathcal{C}_a$ to which agent a belongs
$\mathcal{CC}_a^d$	Same as $\mathcal{CC}_a$ but with agents up to d-hops from a

Table 1: Notation Summary

1) Identifying Similarity Classes. We denote $\mathcal{C}_a^t$ as the set of agents that, at time t, agent a *estimates* to belong to its similarity class $\mathcal{C}_a$. Specifically, agent a includes agent a' in its *estimated* similarity class $\mathcal{C}_a^t$ if their local mean estimates are sufficiently close. In general, at time t, agent a does not have access to the most recent local mean estimate of agent a' (computed over t samples), but rather to a *stale* value $\bar{x}_{a,a'}^t$ computed over $n_{a,a'}^t$ samples, where $n_{a,a'}^t$ corresponds to the time when a

and a' last communicated. Agent a can then estimate its true mean and the true mean of agent a' to belong to the confidence intervals $I_{a,a} = \left[\bar{x}^t_{a,a} - \beta_\gamma(t), \bar{x}^t_{a,a} + \beta_\gamma(t)\right]$, $I_{a,a'} = \left[\bar{x}^t_{a,a'} - \beta_\gamma(n^t_{a,a'}), \bar{x}^t_{a,a'} + \beta_\gamma(n^t_{a,a'})\right]$, respectively. As expected, the interval amplitude $\beta_\gamma(n)$ depends on the number of samples n on which the empirical average is computed and on the target level of confidence 1-2γ associated with the interval. Agent a will then consider a' to belong to its *estimated* similarity class $\mathcal{C}^t_a$ if the two intervals overlap, i.e., $I_{a,a} \cap I_{a,a'} \neq \emptyset$, or equivalently if the optimistic distance $d^t_\gamma(a, a')$ is zero or less:

$$d^t_\gamma(a, a') := |\bar{x}_{a,a} - \bar{x}_{a,a'}| - \beta_\gamma(t) - \beta_\gamma(n^t_{a,a'}) \leqslant 0. \quad (1)$$

To achieve the most accurate estimation of its similarity class, agent a could retrieve from each other peer a' the most recent estimate $\bar{x}^t_{a,a'}$ at each time t. However, this approach would result in a per-agent communication burden of $\mathcal{O}(|\mathcal{A}|)$. To mitigate this effect, node a cyclically queries a *single* node from $\mathcal{C}^{t-1}_a$ at each time instant t, according to a *Round-Robin* scheme. This leads to the following update rule for $\mathcal{C}^t_a$:

$$\mathcal{C}^t_a = \{a' \in \mathcal{C}^{t-1}_a : d^t_\gamma(a, a') \leqslant 0\}, \quad (2)$$

and we observe that by construction $\mathcal{C}^t_a \subseteq \mathcal{C}^{t-1}_a$. Initially, the agent sets $\mathcal{C}^0_a = \mathcal{A}$ and progressively makes irreversible decisions to remove agents that are deemed too dissimilar.

2) Estimating the Mean. Each node a computes an *estimate* $\hat{\mu}^t_a$ of its true mean μ_a combining the available estimates according to a *simple weighting* scheme, where the number of samples $n^t_{a,a'}$ are the weights:

$$\hat{\mu}^t_a = \sum_{a' \in \mathcal{C}^t_a} \frac{n^t_{a,a'}}{\sum_{a' \in \mathcal{C}^t_a} n^t_{a,a'}} \bar{x}^t_{a,a'}.$$

ColME's Theoretical Guarantees. Asadi et al. (2022) prove that if data distributions $\{D_a\}_{a \in \mathcal{A}}$ are sub-Gaussian and the amplitude of the confidence intervals is selected as:

$$\beta_\gamma(n) = \sigma \sqrt{\frac{2}{n}\left(1 + \frac{1}{n}\right) \ln(\sqrt{(n+1)}/\gamma)}, \quad (3)$$

then, with high probability, client a correctly identifies its similarity class ($\mathcal{C}^t_a = \mathcal{C}_a$) and obtains an ϵ-accurate estimate for t larger than two opportune constants ζ_a and τ_a, respectively (see Appendix A for the expressions and the results' statements).

ColME's Limitations. Asadi et al. (2022) acknowledge in their paper the main limitations of ColME: each agent requires a memory footprint and computational complexity proportional to $|\mathcal{A}|$. Indeed, each agent a must store all neighbors' local estimates $\bar{x}^t_{a,a'}$ and the corresponding sample counts $n^t_{a,a'}$. Additionally, as agent a receives a new sample at each time t, it updates its local estimate $\bar{x}^t_{a,a}$ (and also $n^t_{a,a}$). This update affects the distance $d^t_\gamma(a, a')$, which thus must be recomputed for all $a' \in \mathcal{C}^t_a$. This leads to a per-agent time and space complexity of $\mathcal{O}(|\mathcal{A}|)$ per time slot, which becomes impractical in large-scale systems. Moreover, while the *Round-Robin* query scheme reduces the communication burden, it introduces a significant delay in the estimation as $\zeta_a \in \mathcal{O}(|\mathcal{A}|)$.

4 Scalable Algorithms Over a Graph $\mathcal{G}$

Online mean estimation can be made $\tilde{\mathcal{O}}(1)$ with our *scalable* approaches: C-ColME (Sec. 4.1), and B-ColME (Sec. 4.2). Both algorithms consider agents $\mathcal{A}$ organized in a fixed graph $\mathcal{G}(\mathcal{A}, \mathcal{E})$ and restrict communication to pairs of agents adjacent in $\mathcal{G}$. Let $\mathcal{N}_a$ and $\mathcal{N}^d_a$ denote the set of neighbors of agent a and the set of agents at distance at most d from a, respectively. Let r represent the maximum size of any agent's neighborhood in $\mathcal{G}$, i.e., $|\mathcal{N}_a| \leq r$ for all $a \in \mathcal{A}$.

Consider the subgraph $\mathcal{G}'$ of $\mathcal{G}$ induced by the agents in $\mathcal{C}_a$. Let $\mathcal{CC}_a$ denote the (initially unknown) set of agents in the connected component of $\mathcal{G}'$ to which a belongs, and let $\mathcal{CC}^d_a \subset \mathcal{CC}_a$ represent the subset of agents within $\mathcal{CC}_a$ that are at most d hops away from a.

Each agent a aims to identify which nodes in its neighborhood $\mathcal{N}_a$ belong to its similarity class $\mathcal{C}_a$. To achieve this, agent a receives at time t an updated local mean estimate $\bar{x}^{t'}_{a,a'}$ from each neighbor $a' \in \mathcal{N}_a$. We denote with $\mathcal{C}^t_a \subseteq \mathcal{N}_a$ the set of neighbors that agent a deems to belong to its own similarity class at time t, initially: $\mathcal{C}^0_a = \mathcal{N}_a$. Similarly to ColME, at each time t, agent a first computes the distance $d^t_\gamma(a, a')$ for every $a' \in \mathcal{C}^{t-1}_a$ according to (1) and then updates $\mathcal{C}^t_a$ according to (2). As for ColME, $\mathcal{C}^t_a \subseteq \mathcal{C}^{t-1}_a$ and as soon as a removes a' from $\mathcal{C}^t_a$, it stops communicating with a'. Subsequently, communication occur over the pruned graph $\mathcal{G}^t = (\mathcal{A}, \mathcal{E}^t)$, where $\mathcal{E}^t = \{(a, a') \in \mathcal{E} : a' \in \mathcal{C}^t_a\}$.

The theoretical guarantees of our algorithms hold under more general settings than those in (Asadi et al. 2022). In particular, they apply to any set of distributions $\{D_a\}_{a \in \mathcal{A}}$ for which the following assumption is satisfied:

Assumption 1. *There exists a positive function $\beta_\gamma(\cdot) \in o(1)$ such that the true mean belongs to all intervals centered in $\bar{x}^t_{a,a}$ of width $\pm\beta_\gamma(t)$ for $t \in \mathbb{N}$ with confidence $1 - 2\gamma$, namely:*

$$\mathbb{P}\left(\forall t \in \mathbb{N}, \left|\bar{x}^t_{a,a} - \mu\right| < \beta_\gamma(t)\right) \geq 1 - 2\gamma, \forall a \in \mathcal{A}. \quad (4)$$

Assumption 1 is satisfied by sub-Gaussian distributions (SGD) with parameter σ^2, by selecting $\beta_\gamma(\cdot)$ as in (3). In Appendix B, we show that the assumption also holds for bounded fourth-moment distributions (BFMD) for $\beta_\gamma(\cdot)$ chosen as follows:

$$\beta_\gamma(n) = \left(2\frac{(\kappa+3)\sigma^4}{\gamma}\right)^{\frac{1}{4}} \left(\frac{1 + \ln^2 n}{n}\right)^{\frac{1}{4}}, \quad (5)$$

where σ^2 bounds the variance of the distributions $\{D_a, \forall a \in \mathcal{A}\}$ and $\kappa\sigma^4$ their fourth moment. When all the variables are identically distributed, κ corresponds to the kurtosis. Moreover, for distributions with a larger number of bounded moments, tighter expressions can be derived for $\beta_\gamma(\cdot)$ (see Remark 1 in Appendix B). In what follows, we assume that Assumption 1 is always satisfied.

We aim first to determine the time needed for all agents in the connected component $\mathcal{CC}_a$ to identify the subset of neighbors residing in their similarity class, i.e., $\mathcal{C}_{a'}^t = \mathcal{C}_{a'} \cap \mathcal{N}_{a'}, \forall a' \in \mathcal{CC}_a$. Following a similar approach to Asadi et al. (2022, Theorem 1) we can prove that:

Theorem 1. [Proof in Appendix B] *Considering an arbitrarily chosen agent a in $\mathcal{A}$, for any $\delta \in (0,1)$, employing either B-ColME or C-ColME we have:*

$$\mathbb{P}\left(\exists t > \zeta_a^D, \exists a' \in \mathcal{CC}_a : \mathcal{C}_{a'}^t \neq \mathcal{C}_{a'} \cap \mathcal{N}_{a'}\right) \leqslant \frac{\delta}{2}, \qquad (6)$$

with $\zeta_a^D = n_\gamma^\star\left(\frac{\Delta_a}{4}\right) + 1$, $\Delta_a = \min_{a' \in \mathcal{A}\setminus\mathcal{C}_a} \Delta_{a,a'}$, $\gamma = \frac{\delta}{4r|\mathcal{CC}_a|}$. $n_\gamma^\star(x)$ denotes the minimum number of samples that are needed to ensure $\beta_\gamma(n) < x$, i.e., $n_\gamma^\star(x) = \lceil \beta_\gamma^{-1}(x) \rceil$.

This result demonstrates that the time required for all agents a' in $\mathcal{CC}_a$ (the connected component to which a belongs) to correctly identify their neighbors within the same similarity class $\mathcal{C}_a$ is bounded by $n_\gamma^\star(\frac{\Delta_a}{4})+1$. Here, $n_\gamma^\star(\frac{\Delta_a}{4})$ represents the number of samples needed to distinguish (with confidence $1-2\gamma$) the true mean of agent a from that of an agent belonging to the 'closest' similarity class (i.e., the one with the closest true mean). The additional 1 accounts for the unit delay in communicating with the neighbors.

When comparing performance of B-ColME and C-ColME (Theorem 1) with ColME (Asadi et al. 2022) [Theorem 1] (reported in Appendix A as Theorem 7 for completeness), we observe that for large systems, if $r|\mathcal{CC}_a| \in \Theta(|\mathcal{A}|)$, $\zeta_a \approx |\mathcal{A}| + \zeta_a^D$, showing that, as expected, agents can identify much faster similar agents in their neighborhood than in the whole population $\mathcal{A}$.[1] See Sec. 5 for a detailed comparison of ColME, C-ColME, and B-ColME.

4.1 Consensus-Based Algorithm: C-ColME

This section introduces the first collaborative mean estimation approach, inspired by consensus algorithms in dynamic settings, as in (Montijano et al. 2014; Franceschelli and Gasparri 2019). The basic idea is that each agent maintains two metrics: the empirical average of its local samples $\bar{x}_{a,a}^t$, and the 'consensus' estimate $\hat{\mu}_a^t$. The consensus variable is updated at time t by computing a convex combination of the local empirical average $\bar{x}_{a,a}^t$ and a weighted sum of the consensus estimates in its (close) neighborhood $\{\hat{\mu}_{a'}^{t-1}, a' \in \mathcal{C}_a^{t-1} \cup \{a\}\}$, see Algorithm 1.

The dynamics of all estimates are captured by:

$$\hat{\boldsymbol{\mu}}^{t+1} = (1-\alpha_t)\,\bar{\mathbf{x}}^{t+1} + \alpha_t W_t \hat{\boldsymbol{\mu}}^t, \qquad (7)$$

where $(W_t)_{a,a'} = 0$ if $a' \notin \mathcal{C}_a^t$ and $\alpha_t \in (0,1)$ is the memory parameter. Once the agents cease pruning their neighbors, say at time τ, the matrix W_t does not need to change anymore, i.e., $W_t = W$ for any $t \geqslant \tau$ with $W_{a,a'} > 0$ if and only if $a' \in \mathcal{C}_a \cap \mathcal{N}_a$. In order to achieve consensus, the matrix W needs to be doubly stochastic (Xiao and Boyd 2004) and we also require it to be symmetric. By time τ, the original communication graph is split into C connected components, where component c includes n_c agents. By an opportune permutation of the agents, we can write the matrix W as follows

$$W = \begin{pmatrix} {}_1W & 0_{n_1\times n_2} & \cdots & 0_{n_1\times n_C} \\ 0_{n_2\times n_1} & {}_2W & \cdots & 0_{n_2\times n_C} \\ \cdots & \cdots & \cdots & \cdots \\ 0_{n_C\times n_1} & 0_{n_C\times n_2} & \cdots & {}_CW \end{pmatrix}, \qquad (8)$$

where each matrix ${}_cW$ is an $n_c \times n_c$ symmetric stochastic matrix. For $t \geqslant \tau$, the estimates in the different components evolve independently. We can then focus on a given component c. All agents in the same component share the same expected value, which we denote by $\mu(c)$. Moreover, let ${}_c\boldsymbol{\mu} = \mu(c)\mathbf{1}_c$. We denote by ${}_c\mathbf{x}^t$ and ${}_c\hat{\boldsymbol{\mu}}^t$ the n_c-dimensional vectors containing the samples' empirical averages and the consensus estimates for the agents in component c and by $\lambda_{2,c}$ the second largest module of the eigenvalues of ${}_cW$.

Note that the actual evolution of W_t is challenging to characterize due to topology changes during the graph pruning phase. However, our main results (Theorems 2 and 3) remain applicable to any system where the sequence of matrices W_t for $t \leq \tau$ is arbitrarily set.

Algorithm 1: C-ColME over a Time Horizon H

Input: $\mathcal{G} = (\mathcal{A}, \mathcal{E})$, $(D_a)_{a\in\mathcal{A}}$, $\epsilon \in \mathbb{R}^+$, $\delta \in (0,1]$
Output: $\hat{\mu}_a$, $\forall a \in \mathcal{A}$ with $\mathbb{P}\left(|\hat{\mu}_a - \mu_a| < \epsilon\right) \geqslant 1-\delta$
$\mathcal{C}_a^0 \leftarrow \mathcal{N}_a, \forall a \in \mathcal{A}$
for time t in $\{1, .., H\}$ **do**
  In parallel for all nodes $a \in \mathcal{A}$
  Draw $x_a^t \sim D_a$
  $\bar{x}_a^t \leftarrow \frac{t-1}{t}\bar{x}_a^{t-1} + \frac{1}{t}x_a^t$
  Compute $\beta_\gamma(t)$ with Eq. (3) or Eq. (5)
  for neighbor a' in $\mathcal{N}_a \cap \mathcal{C}_a^{t-1}$ **do**
    $d_\gamma^t(a,a') \leftarrow \left|\bar{x}_a^t - \bar{x}_{a'}^{t-1}\right| - \beta_\gamma(t) - \beta_\gamma(t-1)$
  end for
  $\mathcal{C}_a^t \leftarrow \left\{a' \in \mathcal{N}_a \cap \mathcal{C}_a^{t-1} \text{ s.t. } d_\gamma^t(a,a') \leqslant 0\right\}$
  $\hat{\mu}_a^t \leftarrow (1-\alpha_t)\bar{x}_a^t + \alpha_t \sum_{a'\in\mathcal{C}_a\cup\{a\}} (W_t)_{a,a'}\hat{\mu}_{a'}^{t-1}$
end for

Theorem 2. [Proof in Appendix F] *Consider a system which evolves according to* (7) *with $W_t = W$ in* (8)*, for $t \geqslant \tau$. Let ${}_cP = 1/n_c \mathbf{1}_c\mathbf{1}_c^\intercal$. For $\alpha_t = \frac{t}{t+1}$, it holds:*

$$\mathbb{E}\left[\|{}_c\hat{\boldsymbol{\mu}}^{t+1} - {}_c\boldsymbol{\mu}\|^4\right] \in \mathcal{O}\left(\sup_{W_1,\cdots,W_{\zeta_D}} \frac{\mathbb{E}\left[\|{}_c\hat{\boldsymbol{\mu}}^{\zeta_D} - {}_c\boldsymbol{\mu}\|^4\right]}{(t+1)^4}\right)$$
$$+ \mathcal{O}\left(\frac{(1-1/\ln\lambda_{2,c})^2}{(1-\lambda_{2,c})^2}\frac{\mathbb{E}\left[\|{}_c\mathbf{x} - {}_cP\,{}_c\mathbf{x}\|^4\right]}{(t+1)^4}\right)$$
$$+ \mathcal{O}\left(\mathbb{E}\left[\|{}_cP\,{}_c\mathbf{x} - {}_c\boldsymbol{\mu}\|^4\right]\left(\frac{1+\ln t}{1+t}\right)^2\right).$$

The theorem shows that the error, quantified through the fourth moment, can be decomposed into three terms decreasing over time. The first term depends on the estimates' error at time τ. The second term captures the effect of the consensus averaging, i.e., how effective is the algorithm in bringing

[1] For a fairer comparison, we should let ColME query r other agents at each time t, where r is the average degree of $\mathcal{G}$. In this case, $\zeta_a \approx |\mathcal{A}|/r + \zeta_a^D$ and the conclusion does not change.

the local estimates ${}_c\mathbf{x}$ close to their empirical value ${}_cP\,{}_c\mathbf{x}$ (for example it is minimized if $\lambda_2 = 0$, which corresponds to ${}_cW = {}_cP$, the ideal choice for the matrix ${}_cW$). Finally, the third term represents the minimum possible error, which would be obtained by averaging the estimates of all agents in the component using the matrix ${}_cP$.

Theorem 3 shows that C-ColME achieves a speedup proportional to the size of the connected component $|\mathcal{CC}_a|$.

Theorem 3. [Proof in Appendix F] *Consider a graph component c and pick uniformly at random an agent a in c. Let $g(x) \coloneqq x\ln^2(ex)$ and $\alpha_t = \frac{t}{t+1}$. Under BFMD, it holds:*

$$\mathbb{P}\left(\forall t > \tau_a^C,\, |\hat{\mu}_a^t - \mu_a| < \epsilon\right) \geqslant 1-\delta,$$

with $\tau_a^C = \max\left\{\zeta_a^D, g\left(\frac{\mathbb{E}\left[\|{}_cP\,{}_c\mathbf{x}-{}_c\boldsymbol{\mu}\|^4\right]}{|\mathcal{CC}_a|\epsilon^4\delta}\right) \in \tilde{\mathcal{O}}\left(\frac{\widetilde{n}_{\frac{\delta}{2}}(\varepsilon)}{|\mathcal{CC}_a|}\right)\right\}$ *and* $\widetilde{n}_{\frac{\delta}{2}}(\varepsilon) = \left\lceil \frac{2(\kappa+3)\sigma^4}{\delta\varepsilon^4}\right\rceil$.

The theorem shows that the time to reach an ϵ-accurate estimate with high probability is the maximum of the time for the agents in $\mathcal{CC}_a$ to identify their neighbors in the same similarity class and the time required for those agents to obtain and ϵ-accurate estimate if they could share their own samples. Indeed, we observe that $n^\star_{\delta/2}(\epsilon)$ is the number of samples sufficient to ensure that $\mathbb{P}(|\hat{\mu}_a - \mu_a| > \epsilon) < \delta/2$ (see details in Appendix E) and that the nodes in $\mathcal{CC}_a$ collectively gather this number of samples by time $t = \left\lceil \frac{n^\star_{\delta/2}(\epsilon)}{|\mathcal{CC}_a|}\right\rceil$.

Appendix F also presents convergence results for the case $\alpha_t = \alpha$, but they do not enjoy the same speedup factor.

4.2 Message-Passing Algorithm: B-ColME

In B-ColME, each node $a \in \mathcal{A}$ continuously exchanges *messages* with its direct neighbors $a' \in \mathcal{C}_a^t$. This enables node a to acquire not only the neighbor's local estimates $\{\bar{x}^t_{a,a'}, a' \in \mathcal{C}_a^t\}$, but also aggregated estimates from nodes up to a distance d in the graph $\mathcal{G}^t$ (where d is a tunable parameter). Indeed, each neighbor a' acts as a *forwarder*, granting node a access to the records from its own neighbors $a'' \in \mathcal{C}^t_{a'} \setminus \{a\}$. Provided each agent correctly identifies all similar nodes in its neighborhood, agent a can potentially access the (delayed) local estimates of all agents in $\mathcal{CC}_a^d$.

In our message-passing scheme, at time t, agent $a \in \mathcal{A}$ receives a message $M^{t,a'\to a}$ from all neighbors $a' \in \mathcal{C}_a^t$. The message $M^{t,a'\to a}$ is a $d \times 2$ table whose elements $m^{t,a'\to a}_{h,1}$ contain a sum of samples, while $m^{t,a'\to a}_{h,2}$ indicates the number of samples contributing to this sum. In particular, at each time t, the first row of the table is set as: $m^{t,a'\to a}_{1,1} = \sum_{\tau=1}^t x^\tau_{a'}$ and $m^{t,a'\to a}_{1,2} = t$, i.e., the immediate neighbors' sum of local samples. The remaining entries are computed through the following recursion:

$$m^{t,a'\to a}_{h,i} = \sum_{a''\in\mathcal{C}^t_{a'},\, a''\neq a} m^{t-1,a''\to a'}_{h-1,i},$$

for $h \in \{2,\dots,d\}$ and $i \in \{1,2\}$. This captures information extending beyond immediate neighbors. For additional details on B-ColME see Algorithm 2 and Fig. 5 in Appendix D.

If $\mathcal{G}^t \cap \mathcal{N}_a^d$ is a tree, then $m^{t,a'\to a}_{h,1}$ contains the sum of all samples generated within time $t-h+1$ by agents $a'' \in \mathcal{G}^t$ at distance $h-1$ from a' and distance h from a, while $m^{t,a'\to a}_{h,2}$ contains the corresponding number of samples (the proof is by induction on h). Agent a can estimate its mean as:

$$\hat{\mu}_a^t = \frac{\sum_{\tau=1}^t x_a^\tau + \sum_{a'\in\mathcal{C}_a^t}\sum_{h=1}^d m^{t,a'\to a}_{h,1}}{t + \sum_{a'\in\mathcal{C}_a^t}\sum_{h=1}^d m^{t,a'\to a}_{h,2}}. \tag{9}$$

Under the local tree structure assumption, this corresponds to performing an empirical average over all the samples generated by all agents in $\mathcal{G}^t$ at distance $0 \leqslant h \leqslant d$ from a up to time $t-h$. If $\mathcal{G}^t \cap \mathcal{N}_a^d$ is not a tree, samples collected by a given agent a'' may be included in messages received by a through different parallel paths (from a to a''). As a result, these samples are erroneously counted multiple times in (9). The parameter d must be chosen to prevent this issue with high probability, as discussed in Sec. 4.3.

Algorithm 2: B-ColME over a Time Horizon H

Input: $\mathcal{G} = (\mathcal{A}, \mathcal{E})$, $(D_a)_{a\in\mathcal{A}}$, $\varepsilon \in \mathbb{R}^+$, $\delta \in (0,1]$
Output: $\hat{\mu}_a$, $\forall a \in \mathcal{A}$ with $\mathbb{P}\left(|\hat{\mu}_a - \mu_a| < \varepsilon\right) \geqslant 1-\delta$
$\mathcal{C}_a^0 \leftarrow \mathcal{N}_a, \forall a \in \mathcal{A}$
for time t in $\{1,..,H\}$ **do**
  In parallel for all nodes $a \in \mathcal{A}$
  Draw $x_a^t \sim D_a$
  $\bar{x}_a^t \leftarrow \frac{t-1}{t}\bar{x}_a^{t-1} + \frac{1}{t}x_a^t$
  Compute $\beta_\gamma(t)$ with Eq. (3) or Eq. (5)
  for neighbor a' in $\mathcal{C}_a^{t-1}$ **do**
    $d^t_\gamma(a,a') \leftarrow \left|\bar{x}_a^t - \bar{x}_{a'}^{t-1}\right| - \beta_\gamma(t) - \beta_\gamma(t-1)$
    if $d^t_\gamma(a,a') > 0$ **then**
      $\mathcal{C}_a^t \leftarrow \mathcal{C}_a^t \setminus \{a'\}$
    end if
  end for
  for neighbor a' in $\mathcal{C}_a^t$ **do**
    Compute $M^{t,a\to a'}$ and send it to a'
  end for
  Wait for messages $M^{t,a'\to a}$ $\forall a' \in \mathcal{C}_a^t$
  $\hat{\mu}_a^t \leftarrow \frac{\sum_{\tau=1}^t x_a^\tau + \sum_{a'\in\mathcal{C}_a^t}\sum_{h=1}^d m^{t-1,a'\to a}_{h,1}}{t+\sum_{a'\in\mathcal{C}_a^t}\sum_{h=1}^d m^{t,a'\to a}_{h,2}}$
end for

Theorem 4 presents the (ϵ,δ) convergence result for B-ColME, which enjoys a speedup proportional to $|\mathcal{CC}_a^d|$.

Theorem 4. [Proof in Appendix E] *Provided that $\mathcal{CC}_a^d$ is a tree, for any $\delta \in (0,1)$, employing B-ColME, we have:*

$$\mathbb{P}\left(\forall t > \tau_a^B,\, |\hat{\mu}_a^t - \mu_a| < \varepsilon\right) \geqslant 1-\delta$$

where $\tau_a^B = \max\left[\zeta_a^D + d, \frac{\widetilde{n}_{\frac{\delta}{2}}(\varepsilon)}{|\mathcal{CC}_a^d|} + d\right]$ *and* $\widetilde{n}_{\frac{\delta}{2}}(\varepsilon) = \left\lceil -\frac{2\sigma^2}{\varepsilon^2}\ln\left(\frac{\delta}{4}(1-\mathrm{e}^{-\frac{\varepsilon^2}{\sigma^2}})\right)\right\rceil$ *for SGD and as in Theorem 3 for BFMD.*

Similar considerations to those for Theorem 3 apply. The additional term d accounts for the delay introduced by the message-passing scheme.

Corollary 5. *Let $\mathbb{P}(\mathcal{CC}_a^d$ is not a tree$) = \delta'$, then for any $\delta \in (0,1)$, employing B-ColME, we have:*

$$\mathbb{P}\left(\forall t > \tau_a^B \,:\, |\hat{\mu}_a^t - \mu_a| < \varepsilon\right) \geqslant 1 - \delta - \delta'.$$

Theorem 23 (Appendix G) provides upper bounds for δ' when $\mathcal{G}$ is a random regular graph. In particular, as long we set d as in Proposition 6 (Sec. 4.3), δ' converges to 0 as the number of agents $|\mathcal{A}|$ increases.

4.3 Choice of the Graph and Other Parameters

The selection of the graph $\mathcal{G}(\mathcal{A}, \mathcal{E})$ is crucial for the effectiveness of our algorithms. Here we state the key desirable properties of $\mathcal{G}(\mathcal{A}, \mathcal{E})$. First, Theorems 3 and 4 show that learning timescales, τ_a^B and τ_a^C, decrease as the size of the collaborating agent groups, $\mathcal{CC}_a^d$ and $\mathcal{CC}_a$, increase. Therefore, a highly connected graph is preferred to promote the formation of large clusters of agents belonging to the same similarity class after the disconnection of inter-class edges. Second, the spatial and temporal complexities of B-ColME and C-ColME are directly proportional to the agents' degree within the graph. Hence, we want the degree to be small and possibly uniform across the agents to balance computation across agents. Note that the first two criteria partially conflict, as a higher degree generally leads to larger groups $\mathcal{CC}_a^d$ and $\mathcal{CC}_a$, while a smaller degree ensures better spatial and temporal complexities. A third criterion, specific to B-ColME, is that each agent's neighborhood should have a tree-like structure extending up to d hops, with d as large as possible.

Considering these criteria, we opt for the class of *simple* random regular graphs $\mathcal{G}_0(N, r)$. These graphs are sampled uniformly at random from the set of all r-regular simple graphs with N nodes, i.e., graphs without parallel edges or self-loops, and in which every node has exactly r neighbors. Note that an even product rN guarantees the set is not empty. The class $\mathcal{G}_0(N, r)$ exhibits strong connectivity properties for small values of r. Specifically, for any $r \geqslant 3$, the probability that the sampled graph is connected approaches one as N increases. Moreover, the sampled graph demonstrates a local tree-like structure with high probability (proof in Appendix G). The choice of r (agents' degree) illustrates the trade-off discussed above between reducing complexity (low r) and having large connected components (high r). A sensible rule is to select r sufficiently large to guarantee that most agents in the smallest (most critical) class belong to the same connected component. Consider a class including a fraction p_{k_a} of agents, Table 3 in Appendix G shows the average fraction of agents in this class that is not connected to the main connected component as a function of r. To keep this fraction below e.g. 10^{-2} a good rule of thumb is $r = 4/p_{k_a}$.

A final key parameter for B-ColME is the maximum distance d over which local estimates from agents are propagated. This parameter must be carefully calibrated: it should be small enough to ensure that $\mathcal{CC}_a^d$, for a randomly chosen $a \in \mathcal{A}$, has a tree-like structure with high probability. However, choosing a d that is too small could unnecessarily restrict the size of $\mathcal{CC}_a^d$, thereby undermining the effectiveness of the estimation process (Theorem 4). A comprehensive analysis of how the parameters r and d influence both the structure of $\mathcal{N}_a^d$ and the size of $\mathcal{CC}_a^d$ can be found in Appendix G. Here, we informally summarize the main result:

Proposition 6. *By selecting $d = \left\lfloor \frac{1}{2} \log_{r-1} \frac{|\mathcal{A}|}{\log_{r-1} ||\mathcal{A}|} \right\rfloor$ the number of nodes $a \in \mathcal{A}$, whose d-neighborhood is not a tree, is $o(|\mathcal{A}|)$ with a probability tending to 1 as $|\mathcal{A}|$ increases. For the same d and $r \in \Theta(\log(1/\delta))$, $|\mathcal{CC}_a^d|$ is in $\Omega(|\mathcal{A}|^{\frac{1}{2}-\phi})$ for any arbitrarily small $\phi > 0$ with probability arbitrarily close to 1.*

Finally, for C-ColME, the consensus matrix W could be chosen to minimize the second largest module $\lambda_{2,c}$ of the eigenvalues of each block ${}_cW$ in order to minimize the bound in Theorem 2. This optimization problem has been studied by Xiao and Boyd (2004) and requires in general a centralized solution. In what follows, we consider the following simple, decentralized configuration rule: $(W_t)_{a,b} = \frac{1}{\max\{|\mathcal{C}_a^t|, |\mathcal{C}_b^t|\}+1}, \forall b \in \mathcal{C}_a^t$ and $(W_t)_{a,a} = 1 - \sum_{b \in \mathcal{C}_a^t} (W_t)_{a,b}$, making W symmetric and doubly stochastic.

5 Algorithms' Comparison

Table 2 presents a comparative analysis of the three algorithms: ColME, C-ColME, and B-ColME. For a fair comparison, we consider a variant of ColME, where each agent can communicate with r agents at each time t, so that all three algorithms incur the same communication overhead.

The second column of Table 2 outlines the space and time complexities of the algorithms. Notably, even when r and d are allowed to increase logarithmically with the number of agents $|\mathcal{A}|$, B-ColME retains its efficiency advantage over ColME. C-ColME demonstrates even greater improvements, further reducing the per-agent burden compared to the savings achieved by B-ColME.

The third and fourth columns detail the characteristic times required to achieve (ϵ, δ) convergence for the estimates generated by the three algorithms, considering both sub-Gaussian local data distributions and distributions with bounded fourth moment. The characteristic times correspond to τ_a, τ_a^C, and τ_a^B in Theorem 7 in Appendix A, Theorem 3, and 4, respectively. The table reports their asymptotic behavior as the number of agents $|\mathcal{A}|$ increases ignoring logarithmic factors. The detailed derivations of these results are provided in Appendix H.

Three factors contribute to the characteristic times. The first factor is the time required to correctly identify potential collaborators. For ColME, this involves each agent classifying the other $|\mathcal{A}| - 1$ agents, leading to a term that scales as $\log |\mathcal{A}|$ or $|\mathcal{A}|$, depending on the assumed properties of the local distribution. For B-ColME and C-ColME, $|\mathcal{A}|$ is replaced by $|\mathcal{CC}_a|r$, which represents an upper bound on the number of connections the agents in $\mathcal{CC}_a$ may have initially established with agents from different classes. This substitution may not be immediate, as one might initially expect the relevant scale to be simply r. However, this adjustment accounts for the potential ripple effect of classification er-

	Per-agent space/ time complexity	Convergence time sub-Gaussian	Convergence time bounded 4-th moment
ColME (r communications)	$\lvert\mathcal{A}\rvert$	$\frac{1}{\Delta_a^2}\log\frac{\lvert\mathcal{A}\rvert}{\Delta_a\delta}+\frac{\lvert\mathcal{A}\rvert}{r}+\frac{1}{\lvert\mathcal{C}_a\rvert}\frac{1}{\varepsilon^2}\log\frac{1}{\delta\varepsilon^2}$	$\frac{1}{\Delta_a^4}\frac{\lvert\mathcal{A}\rvert}{\delta}+\frac{\lvert\mathcal{A}\rvert}{r}+\frac{1}{\lvert\mathcal{C}_a\rvert}\frac{1}{\delta\varepsilon^4}$
C-ColME [Thm. 3] ($\alpha_t=\frac{t}{t+1}$)	r	$-$	$\frac{1}{\Delta_a^4}\frac{\lvert\mathcal{CC}_a\rvert r}{\delta}+\frac{1}{\lvert\mathcal{CC}_a\rvert}\frac{1}{\delta\varepsilon^4}$
B-ColME [Thm. 4]	rd	$\frac{1}{\Delta_a^2}\log\frac{\lvert\mathcal{CC}_a\rvert r}{\Delta_a\delta}+d+\frac{1}{\lvert\mathcal{CC}_a^d\rvert}\frac{1}{\varepsilon^2}\log\frac{1}{\delta\varepsilon^2}$	$\frac{1}{\Delta_a^4}\frac{\lvert\mathcal{CC}_a\rvert r}{\delta}+d+\frac{1}{\lvert\mathcal{CC}_a^d\rvert}\frac{1}{\delta\varepsilon^4}$

Table 2: Comparison of collaborative estimation algorithms. The convergence time is provided in order sense.

rors: a mistake by any agent a can impact the estimates of all agents within the same connected component $\mathcal{CC}_a$.

The second factor contributing to the characteristic times is the time each agent needs to collect all relevant information. For ColME, this time is proportional to $|\mathcal{A}|/r$, as an agent queries all other agents. For B-ColME, the time is specifically tied to d, the maximum number of hops messages propagate. Notably, this term does not appear for C-ColME, as it is dominated by the final term.

The final term represents the time needed for accurate mean estimation after collaborators have been identified, highlighting the benefits of collaboration. In ColME, the collaboration's benefit is particularly striking, as all agents within the same class work together to improve their estimates. This collective effort effectively reduces the convergence time by a factor proportional to the size of the collaborating group, $|\mathcal{C}_a|$. For B-ColME and C-ColME, although the speed-up remains proportional to the number of collaborating agents, the actual numbers of collaborators, $|\mathcal{CC}_a^d|$ for B-ColME and $|\mathcal{CC}_a|$ for C-ColME, are in general smaller.

In conclusion, while ColME potentially offers the most accurate estimates, it requires longer convergence times and greater memory and computational resources. In contrast, B-ColME and C-ColME present more efficient alternatives, achieving faster convergence with reduced per-agent resource demands. However, this efficiency may come at the expense of the maximum attainable accuracy. The next section quantifies this trade-off experimentally.

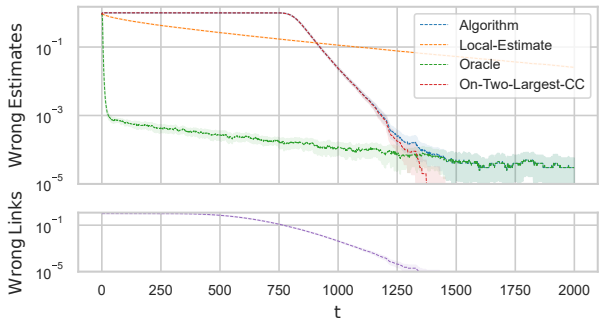
6 Numerical Experiments

We evaluate the proposed algorithms on the class $\mathcal{G}_0(N,r)$ of simple regular graphs (see Sec. 4.3). In this setting, each agent connects to r other agents chosen uniformly at random in $\mathcal{A}$. This setup also provides the tree-like local structure required for B-ColME. Agents belong to one of two classes, associated with Gaussian distributions $D_1 \sim \mathcal{N}(\mu_1 = 0, \sigma^2 = 4)$ and $D_2 \sim \mathcal{N}(\mu_2 = 1, \sigma^2 = 4)$. Each node is assigned to one of the two classes with equal probability. Unless otherwise stated, in the experiments $|\mathcal{A}| = N = 10000$, $r = 10$, $d = 4$, $\varepsilon = 0.1$, $\delta = 0.1$, and $\beta_\gamma(n)$ as in (3). In Appendix C and I, we provide additional experiments for the multidimensional case and varying the system's parameters.

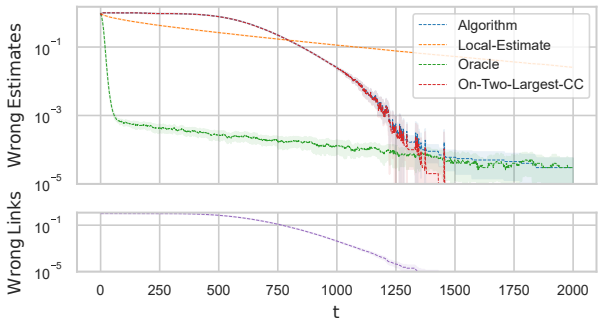
Figure 1 showcases the performance of B-ColME and C-ColME using two key metrics: the fraction of agents with *incorrect estimates* ($\hat{\mu}_a^t$ more than ε away from the true mean μ_a), and the fraction of *wrong links* still in use (a wrong link connects agents from different classes). We compare our algorithms against two benchmarks. The first benchmark has each agent independently relying on its *local* estimate $\bar{x}_{a,a}^t$. In the second benchmark, an *oracle* provides each agent with precise knowledge of which neighbors belong to the same similarity class (i.e., $\mathcal{C}_a^t = \mathcal{C}_a \cap \mathcal{N}_a, \forall a, t$). The figure reveals that B-ColME has a longer transient phase but then exhibits a slightly steeper convergence than C-ColME. Notably, B-ColME's estimates show no apparent improvement until about 90% of the wrong links have been removed, whereas C-ColME's estimates begin to improve as soon as the first edges are eliminated. This phenomenon can be explained as follows. In B-ColME, the estimates at agent a are not influenced by the removal of some wrong links as long as its d-hop neighborhood $\mathcal{N}_a^d$ remains unchanged. For instance, a given node $a' \notin \mathcal{C}_a$ is removed from $\mathcal{N}_a^d$ only when *all* paths of length at most d between agent a and agent a' are eliminated. In contrast, in C-ColME, agent a' contributes to the weighted estimate at agent a with a weight equal to the sum over all paths between a and a' of the product of the consensus weights along the path. As paths are progressively removed, the negative impact of a' on agent a's estimate is gradually reduced. However, once all wrong links are removed, B-ColME benefits from its estimates being computed solely on agents belonging to the same class, while C-ColME requires some additional time for the effect of past estimates to fade away.

We also compare the proposed algorithms with ColME and a simplified version (s-ColME) where the optimistic distance $d_\gamma^t(a,a')$ is recomputed only for the r agents queried at time t, achieving an $\mathcal{O}(r)$ per-agent computational cost (the memory cost remains $\mathcal{O}(|\mathcal{A}|)$). As predicted by the theoretical analysis, B-ColME and C-ColME are faster than ColME, but at the cost of a higher asymptotic error because agent a collaborates only with the smaller group of nodes in $\mathcal{CC}_a^d$ for B-ColME, and $\mathcal{CC}_a$ for C-ColME. ColME pays for this asymptotic improvement with a $\mathcal{O}(|\mathcal{A}|)$ space-time complexity per agent, impractical for large-scale systems. Note that s-ColME improves ColME's complexity at the cost of a much slower discovery of same-class neighbors.

While we focused on online mean estimation, our approach can be adapted to decentralized federated learning. To illustrate this possibility, we adapt the consensus-based decentralized federated learning algorithm, by letting agents progressively exclude neighbors they identify as belonging to a different class. The cosine dissimilarity of agents' updates, the same metric used in Clus-



(a) B-ColME



(b) C-ColME

Figure 1: Fraction of agents with estimate deviates by more than ϵ from the true value, i.e., $|\{a \in \mathcal{A} : |\hat{\mu}_a^t - \mu_a| > \varepsilon\}|/|\mathcal{A}|$ (top) and fraction of *wrong links* (bottom) for B-ColME (a) and C-ColME (b), over 20 realizations with 95% confidence intervals.

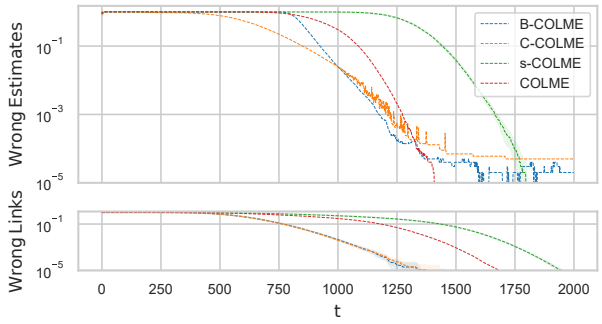


Figure 2: Comparison of our algorithms and two versions of ColME, over 10 realizations.

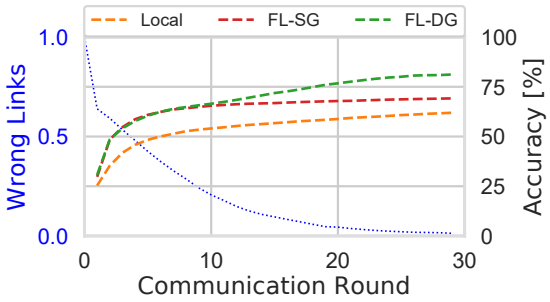


Figure 3: Accuracy of a local model (Local), a decentralized FL over a static graph (FL-SG), and our approach over a dynamic graph (FL-DG). We also show the fraction of links between classes (*wrong links*) over time for FL-DG.

teredFL (Ghosh et al. 2020; Chen et al. 2021), replaces the optimistic distance $d_\gamma^t(a, a')$ (details in Appendix J). Figure 3 shows the performance of our *decentralized FL over a dynamic graph* (FL-DG) with $|\mathcal{A}| = 100$ agents initially organized over a complete graph. Two different distributions are obtained from MNIST (Deng 2012) by swapping/maintaining some labels and each client progressively receives new data samples from one of the two distributions. As the graph is progressively split in two clusters of clients belonging to the same class, each agent's model benefits from cooperating only with similar clients and it achieves a higher accuracy.

7 Conclusions

In this paper, we introduced B-ColME and C-ColME, two scalable and fully distributed algorithms for collaborative local mean estimation. We thoroughly evaluated their performance through both theoretical and empirical analyses. Additionally, we adapted our approach for personalized federated learning, applying it to the task of handwritten digit recognition using the MNIST dataset.

This work points to several future research directions. Here we have allowed agents only to sever existing connections, but not to establish new ones. Investigating scenarios where agents can rewire their connections to communicate with new agents outside their original neighborhood would be an interesting extension. Additionally, we assumed that agents are partitioned into similarity classes, with agents in the same class generating data with identical true mean. Extending our approach to accommodate more general scenarios, where each agent generates data with potentially different true mean, would be a valuable avenue for further exploration.

Acknowledgements

This research was supported in part by the European Network of Excellence dAIEDGE under Grant Agreement Nr. 101120726 and by the Groupe La Poste, sponsor of the Inria Foundation, in the framework of the FedMalin Inria Challenge. It was also funded by the Nokia-Inria challenge LearnNet, and by the French government National Research Agency (ANR) through the UCA JEDI (ANR-15-IDEX-0001), EUR DS4H (ANR-17-EURE-0004), and the 3IA Côte d'Azur Investments in the Future project (ANR-19-P3IA-0002).

Computational resources provided by hpc@polito,which is a project of Academic Computing within the Department of Control and Computer Engineering at the Politecnico di Torino (http://hpc.polito.it).

References

Adi, E.; Anwar, A.; Baig, Z.; and Zeadally, S. 2020. Machine learning and data analytics for the IoT. *Neural Computing and Applications*, 32(20): 16205–16233.

Asadi, M.; Bellet, A.; Maillard, O.-A.; and Tommasi, M. 2022. Collaborative Algorithms for Online Personalized Mean Estimation. *Transactions on Machine Learning Research*.

Beaussart, M.; Grimberg, F.; Hartley, M.-A.; and Jaggi, M. 2021. WAFFLE: Weighted Averaging for Personalized Federated Learning.

Chayti, E. M.; Karimireddy, S. P.; Stich, S. U.; Flammarion, N.; and Jaggi, M. 2022. Linear Speedup in Personalized Collaborative Learning.

Chen, M.; Yang, Z.; Saad, W.; Yin, C.; Poor, H. V.; and Cui, S. 2021. A Joint Learning and Communications Framework for Federated Learning Over Wireless Networks. *IEEE Transactions on Wireless Communications*, 20(1): 269–283.

Deng, L. 2012. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6): 141–142.

Ding, S.; and Wang, W. 2022. Collaborative Learning by Detecting Collaboration Partners. *Advances in Neural Information Processing Systems*, 35: 15629–15641.

Donahue, K.; and Kleinberg, J. 2021. Model-Sharing Games: Analyzing Federated Learning Under Voluntary Participation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Dorner, F. E.; Konstantinov, N.; Pashaliev, G.; and Vechev, M. 2024. Incentivizing honesty among competitors in collaborative learning and optimization. *Advances in Neural Information Processing Systems*, 36: 7659–7696.

Even, M.; Massoulié, L.; and Scaman, K. 2022. On Sample Optimality in Personalized Collaborative and Federated Learning. In *Advances in Neural Information Processing Systems*.

Fallah, A.; Mokhtari, A.; and Ozdaglar, A. 2020. Personalized Federated Learning with Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach. In *Advances in Neural Information Processing Systems*.

Franceschelli, M.; and Gasparri, A. 2019. Multi-stage discrete time and randomized dynamic average consensus. *Automatica*, 99: 69–81.

Galante, F.; Neglia, G.; and Leonardi, E. 2024. Scalable Decentralized Algorithms for Online Personalized Mean Estimation. arXiv:2402.12812.

Ghosh, A.; Chung, J.; Yin, D.; and Ramchandran, K. 2020. An Efficient Framework for Clustered Federated Learning. In *Advances in Neural Information Processing Systems*.

Grimberg, F.; Hartley, M.-A.; Karimireddy, S. P.; and Jaggi, M. 2021. Optimal Model Averaging: Towards Personalized Collaborative Learning.

Kairouz, P.; McMahan, H. B.; Avent, B.; Bellet, A.; Bennis, M.; Nitin Bhagoji, A.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; D'Oliveira, R. G. L.; Eichner, H.; El Rouayheb, S.; Evans, D.; Gardner, J.; Garrett, Z.; Gascón, A.; Ghazi, B.; Gibbons, P. B.; Gruteser, M.; Harchaoui, Z.; He, C.; He, L.; Huo, Z.; Hutchinson, B.; Hsu, J.; Jaggi, M.; Javidi, T.; Joshi, G.; Khodak, M.; Konecný, J.; Korolova, A.; Koushanfar, F.; Koyejo, S.; Lepoint, T.; Liu, Y.; Mittal, P.; Mohri, M.; Nock, R.; Özgür, A.; Pagh, R.; Qi, H.; Ramage, D.; Raskar, R.; Raykova, M.; Song, D.; Song, W.; Stich, S. U.; Sun, Z.; Suresh, A. T.; Tramèr, F.; Vepakomma, P.; Wang, J.; Xiong, L.; Xu, Z.; Yang, Q.; Yu, F. X.; Yu, H.; and Zhao, S. 2021. Advances and Open Problems in Federated Learning. *Foundations and Trends® in Machine Learning*, 14(1–2): 1–210.

Li, T.; Hu, S.; Beirami, A.; and Smith, V. 2021. Ditto: Fair and Robust Federated Learning Through Personalization. In *Proceedings of the 38th International Conference on Machine Learning*.

Li, T.; Sahu, A. K.; Talwalkar, A.; and Smith, V. 2020. Federated Learning: Challenges, Methods, and Future Directions. *IEEE Signal Processing Magazine*, 37(3): 50–60.

Mansour, Y.; Mohri, M.; Ro, J.; and Suresh, A. T. 2020. Three Approaches for Personalization with Applications to Federated Learning.

Marfoq, O.; Neglia, G.; Bellet, A.; Kameni, L.; and Vidal, R. 2021. Federated Multi-Task Learning under a Mixture of Distributions. In *Advances in Neural Information Processing Systems*.

Montijano, E.; Montijano, J. I.; Sagüés, C.; and Martínez, S. 2014. Robust discrete time dynamic average consensus. *Automatica*, 50(12): 3131–3138.

Sattler, F.; Müller, K.-R.; and Samek, W. 2021. Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8): 3710–3722.

Tan, A. Z.; Yu, H.; Cui, L.; and Yang, Q. 2023. Towards Personalized Federated Learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12): 9587–9603.

Tsoy, N.; Mihalkova, A.; Todorova, T. N.; and Konstantinov, N. 2024. Provable Mutual Benefits from Federated Learning in Privacy-Sensitive Domains. In *International Conference on Artificial Intelligence and Statistics*.

Xiao, L.; and Boyd, S. 2004. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1): 65–78.