Robust Federated Clustering under Heterogeneity and Adversaries

Martín Bravo
University of Chile
martin.bravo@ug.uchile.cl

Sebastian Dalleiger
KTH Royal Institute of Technology
sdall@kth.se

Abstract

Clustering distributed and private data is an increasingly important task across domains that handle sensitive information, such as life sciences and clinical research. In federated settings, clustering faces three challenges: heterogeneous client data distributions, adversarial behavior, and strict privacy requirements. Existing approaches often exhibit significant performance degradation under these conditions and fail to return accurate solutions. To overcome these limitations, we introduce a novel federated clustering algorithm that combines client-side differential privacy with Byzantine-robust aggregation at the server, based on a novel efficient and robust clustering procedure. Our method comes with theoretical robustness guarantees, and through extensive experiments on synthetic and real-world data, we demonstrate that it produces high-quality clusters in just a few communication rounds, even in scenarios where state-of-the-art methods fail.

Code —`https://doi.org/10.5281/zenodo.19298175`

1 INTRODUCTION

Federated learning (FL) has become a central paradigm for privacy-preserving data analysis, enabling distributed institutions to collaborate without sharing raw data (Kairouz et al., 2021). For example, in clinical and biomedical settings, where data are highly sensitive and siloed across hospitals or biobanks, federated clustering (FC) offers a natural solution for collaborative data analysis (Garst and Reinders, 2024). In FC, each location (client) shares only a handful of local cluster representatives, which the server aggregates into global centers. While this setup has been studied in distributed or multi-view clustering, our data are so sensitive that transmitting information requires guarantees that user privacy is preserved under any circumstance. Recent work in federated optimization has shown that a small number of communication rounds can often suffice in heterogeneous settings (Li et al., 2020). Motivated by the concentration of client-reported centers, we seek similar communication efficiency for federated clustering under privacy noise.

Despite this, existing methods fail when data is corrupted, center locations are compromised, or when a center consensus is difficult to attain. We summarize these into two failure modes: (1) unreliable local center estimates caused by privacy noise, highly corrupted datasets, distribution shifts, or adversarial manipulation from *Byzantine clients*; and (2) heterogeneity across clients, including non-i.i.d. data, inconsistent local cluster counts, and imbalanced coverage.

By simply applying k-means or k-median on the union of local centers, naive aggregation quickly fails under mild corruption. Robust aggregation techniques, such as Krum filtering (Blanchard et al., 2017) or medoid-based heuristics (Kaufman and Rousseeuw, 1990), improve resilience but are computationally highly demanding. On the other hand, state-of-the-art methods often reduce communication overhead under idealized circumstances but are challenged when these assumptions are violated. Recent methods improve resilience by using additional weights transmitted by clients. This not only requires trust, but also increases the worst-case sensitivity, which exacerbates the required level of differential privacy noise (Nguyen et al., 2021).

While honest clients produce highly concentrated, often overlapping estimates around the true centers, existing methods fail to utilize this fact. Instead, they often rely on inefficient, general-purpose clustering. We introduce a federated k-median framework that explicitly leverages these concentration patterns at the server. There, a robust k-centers procedure first selects reliable *core*

Proceedings of the 29th International Conference on Artificial Intelligence and Statistics (AISTATS) 2026, Tangier, Morocco. PMLR: Volume 300.

points, suppressing sparse or adversarial regions; then we refine each core into a global center via a weighted *geometric median* of its local *cover* neighborhood, yielding metric medians resilient to privacy noise and large perturbations. Not only is our method computationally efficient, but it also converges in a few rounds and comes with provable identification and estimation guarantees under mild separation and concentration conditions. In summary, our contributions are:

1. A federated k-median clustering framework that aggregates client centers via robust cores and geometric medians.
2. A robust k-centers algorithm that efficiently identifies reliable candidates under data heterogeneity and adversarial manipulation.
3. Formal identification guarantees under mild separation and concentration assumptions.
4. Formal robustness and convergence guarantees.
5. Extensive evaluation on synthetic and clinical-style datasets, demonstrating superior robustness, privacy, efficiency, and scalability compared to existing federated clustering baselines.

2 RELATED WORK

Our work is related to federated clustering, robust clustering, as well as Byzantine resilience.

Federated Clustering Not to be confused with clustered federated learning (Ghosh et al., 2020), federated clustering is a research area that extends federated learning (McMahan et al., 2017; Karimireddy et al., 2020) to the clustering of sensitive distributed data. One-shot algorithms like k-FED (Dennis et al., 2021) minimize communication under idealized circumstances. Like our method, k-FED exploits the center distribution for improved k-means initialization, but it has low resilience when confronted with suboptimal circumstances. To achieve stronger empirical performance, multi-round methods (Garst and Reinders, 2024) have been developed, typically by communicating cluster centers to a server, while coresets have been employed (Huang et al., 2022) in vertical federated learning. Furthermore, recent privacy-preserving methods adapt aggregation to be robust against privacy noise (Scott et al., 2025; Nguyen et al., 2021), but they expect a relatively low noise level or server-side data, whereas we consider highly compromised data. Similar to federated clustering, recent federated matrix factorization (Dalleiger et al., 2025; Dalleiger and Gionis, 2025) methods also decompose distributed data into latent structure. They, however, fundamentally set different objectives and do not address the Byzantine or privacy challenges. Federated clustering has also been paired with fairness (Zhu et al., 2023; Zheng et al., 2023) and machine unlearning (Pan et al., 2023). However, all these methods are highly susceptible to compromised data, outliers, attacks, and noise. Unlike our method, they largely do not algorithmically exploit structure and employ comparatively inefficient general-purpose aggregation algorithms.

Robust Clustering Robust clustering under outliers and adversarial corruption has been extensively studied in centralized settings. Classical approaches include trimmed k-means (Cuesta-Albertos et al., 1997), which removes a fraction of points before clustering, and M-estimators that downweight outliers (García-Escudero et al., 2008). More recently, Charikar et al. (2017) provided polynomial-time algorithms for clustering with outliers under separation assumptions, while, in the context of k-median clustering, Chawla and Gionis (2013) analyzed robustness properties and approximation guarantees. The geometric median estimator (Minsker and Strawn, 2024) has recently been applied to robust mean estimation but was not considered for robust federated clustering. While they could be translated to the federated setting, they do not recognize that the server's input consists of pre-aggregated local centers rather than raw data points, making them computationally less efficient choices. Furthermore, existing robust clustering algorithms typically do not provide privacy guarantees. Our approach adapts robust estimation principles (weighted neighborhood geometric median) to the federated clustering paradigm while maintaining differential privacy and efficiency.

Byzantine Resilience Byzantine-robust aggregation has become central to federated learning with adversarial clients. Blanchard et al. (2017) select the client whose update is closest to its neighbors, providing robustness guarantees when fewer than half the clients are Byzantine. Yin et al. (2018) analyzed coordinate-wise median and trimmed mean aggregators, proving convergence under bounded Byzantine clients. Pillutla et al. (2022) proposed robust aggregation via the geometric median of client updates. However, these approaches are primarily designed for supervised federated learning where gradients or model parameters are aggregated. We focus on the unsupervised case where Byzantine clients may create spurious high-quality local clusters to mislead aggregation.

3 PROBLEM FORMULATION

Data are distributed across ℓ locations, where each location i holds data that is well represented by a set of k_i local centers $C_i^* = \{c_{i1}^*, \ldots, c_{ik_i}^*\}$. There is a global set $C^* = \{c_1^*, \ldots, c_k^*\}$ of k true clusters, and each C_i^* is (approximately) a subset of C^*. Our goal is to identify the true set of k centers that best summarize all locations, while preserving privacy by preventing leakage of sensitive local data. To this end, we resort to the framework of federated clustering, where locations are represented by *clients*, and the global summarization is done by the *server*. Each client is restricted to communicating only its k_i centers to the server, which we must aggregate into a consistent global solution of size k, where $k_i \leq k$. Overall, we seek to optimize the k-median objective

$$\operatorname*{arg\,min}_{C \subset \mathbb{R}^d, |C|=k} \sum_{i=1}^{\ell} \sum_{x \in X_i} \min_{c \in C} \|x - c\|_2,$$

where X_i is the dataset at location i, C is the set of k global centers, and $\|\cdot\|_2$ denotes the Euclidean distance. Note that the proposed solution will only require minor modifications when targeting the coordinate-wise median (L_1) instead. Each client i aims to find k_i representative centers by solving a local k-median problem

$$C_i \leftarrow \operatorname*{arg\,min}_{C_i \subset \mathbb{R}^d, |C_i|=k_i} \sum_{x \in X_i} \min_{c \in C_i} \|x - c\|_2,$$

where C_i is the set of local centers communicated to the server. Each client thus contributes a compact core summary of its data to the global objective without sharing sensitive raw samples. However, data transmitted from clients may be highly compromised. We distinguish between honest-but-cautious clients and Byzantine clients. *Honest-but-cautious* clients transmit data under controlled noise mechanisms (e.g., differentially private perturbations). *Byzantine clients*, in contrast, may transmit highly corrupted or adversarial data, including outliers or noise.

Honest clients Even when clients are honest, sharing local centers may inadvertently leak sensitive information. For example, a client might internally compute or obtain k medoids, or local centers could fall very close to actual data points, making it possible to infer individual samples. To prevent this information leakage, we ensure that all communicated centers satisfy formal (ϵ, δ)-*differential privacy* (DP) guarantees. Intuitively, DP allows communicating useful aggregate information while provably protecting individual contributions.

Definition 1 ((ϵ, δ)-Differential Privacy (Dwork et al., 2014))**.** *A randomized mechanism $\mathcal{M} : \mathcal{D} \to \mathcal{R}$ satisfies (ϵ, δ)-DP if, for any two adjacent datasets $X, X' \in \mathcal{D}$ (differing by at most one data point) and any subset of outputs $S \subseteq \mathcal{R}$, it holds that*

$$\mathbb{P}(\mathcal{M}(X) \in S) \leq e^{\epsilon} \mathbb{P}(\mathcal{M}(X') \in S) + \delta.$$

Definition 2 (Gaussian Mechanism (Dwork et al., 2014))**.** *The mechanism $\mathcal{M}_{Gauss}(X) = f(X) + \mathcal{N}(0, \sigma^2 \mathbf{I})$ achieves (ϵ, δ)-DP by adding Gaussian noise with covariance $\sigma^2 \mathbf{I}$, where $\sigma^2 = \frac{2\,\Delta_2^2(f) \log(1.25/\delta)}{\epsilon^2}$, and $\Delta_2(f) = \max_{X \sim X'} \|f(X) - f(X')\|_2$ is the sensitivity.*

For algorithms communicating over T rounds, basic composition yields $\epsilon_{\text{total}} = \sum_{t=1}^{T} \epsilon_t$ and $\delta_{\text{total}} = \sum_{t=1}^{T} \delta_t$.

At each round, the server supplies public or previously privatized anchor centers $A_i = \{a_{i1}, \ldots, a_{ik_i}\}$. Client i computes its local centers and communicates center updates, analogous to norm-clipped

$$u_{ij} = \frac{c_{ij} - a_{ij}}{\max\{1, \|c_{ij} - a_{ij}\|_2 / r\}}, \qquad j \in [k_i],$$

where r is a public radius. Since $\|u_{ij}\|_2 \leq r$ independently of the client data, any two possible values of the jth update differ by at most $2r$. A substitution may affect all k_i updates, so the joint ℓ_2-sensitivity satisfies

$$\Delta_{2,i} \leq \sqrt{\sum_{j=1}^{k_i} (2r)^2} = 2\sqrt{k_i}\, r \leq 2\sqrt{k}\, r.$$

The client applies the Gaussian mechanism to the complete clipped-update message, and the server reconstructs the privatized centers as $\widetilde{c}_{ij} = a_{ij} + \widetilde{u}_{ij}$. We set $r = D_2/(2s)$ for a public clipping-strength parameter $s \geq 1$, yielding $\Delta_{2,i} \leq \sqrt{k_i} D_2/s$. In practice, we set s to a public, data-independent lower bound on the local cluster size. For the Laplace mechanism, we analogously project in ℓ_1 using radius $r_1 = D_1/(2s)$, which gives the sensitivity $\Delta_{1,i} \leq k_i D_1/s$.

Byzantine clients Clients may contribute noisy, corrupted, or even adversarial information. Beyond the inherent stochasticity from honest-but-cautious clients, some clients may operate on datasets that deviate from the global distribution, or they may transmit manipulated centers. This leads to significant corruption at the server, such as extreme *outliers*, low-density centers, or systematic manipulations. To ensure reliability, it is therefore paramount to introduce robustness into the aggregation, yielding the following problem formulation.

Problem 1 (Robust Federated Clustering)**.** *Each client $\mathcal{F}_i : \mathcal{X}_i \to \mathcal{C}_i$ maps its local dataset $X_i \in \mathcal{X}_i$ to a summary set of k_i centers. Consider ℓ clients partitioned into two groups $H \sqcup B = \{\mathcal{F}_i\}_i$: a fraction $1 - \alpha \in (0, 1)$ of* honest-but-cautious *DP clients H, and a fraction α of* Byzantine clients *B. Aggregate $\mathcal{A} : \biguplus_{i=1}^{\ell} \mathcal{C}_i \to \mathcal{C}$ local candidates into a global set of k centers $C = \mathcal{A}(C_1, \ldots, C_\ell)$ such that*

$$C = \operatorname*{arg\,min}_{C \subset \mathbb{R}^d, |C|=k} \sum_{i \in H} \sum_{x \in X_i} \min_{c \in C} \|x - c\|_2 \,.$$

Algorithm 1 FORK: Federated Clustering

Require: Number of medians k, number of rounds T
Ensure: Global medians C

function CLIENT$_i(C)$
  $\mathcal{I} \leftarrow \emptyset$ if $t = 1$; otherwise BESTMATCH(C, C_i)
  $C_i \leftarrow k_i$-MEDIAN$(X_i, \mathcal{I})$
  if *differentially private* **then**
    $C_{ij} \leftarrow C_{ij} + \eta_{ij},\ \eta_{ij} \sim \mathcal{N}(0, \sigma^2 \mathrm{I})\ \forall\ j$
  end if
  return C_i
end function
function SERVER$(\widetilde{C})$
  return CORK$(\widetilde{C}, k)$
end function
$C \leftarrow \emptyset$
for $t = 1$ to T **do**
  for all client i **in parallel do**
    $C_i \leftarrow$ CLIENT$_i(C)$
  end for
  $C \leftarrow$ SERVER$(\dot{\bigcup}_i C_i)$
end for
return C

4 METHOD

All **proofs** are deferred to Appendix A.

Our method estimates k global cluster centers from local client estimates. Clients perform a local clustering, and the server aggregates centers into a global estimate. Each client i solves a local k_i-median problem for the local data $\mathcal{X}_i$ by solving

$$\textbf{Client:} \quad C_i \leftarrow \operatorname*{arg\,min}_{S, |S| = k_i} \sum_{x \in \mathcal{X}_i} \min_{s \in S} \|x - s\|_2 \ .$$

After adding privacy noise, these centers C_i are then communicated to the server. In the first round, where no global centers C yet exist, clients initialize their local centers using a local method like k-median++ (Arthur and Vassilvitskii, 2006). Then, let $C = \{c_1, \ldots, c_k\}$ be the set of global centers from the previous server aggregation. In later rounds, we initialize clients by selecting the most relevant subset of C of size k_i, identified by the Hungarian algorithm (Kuhn, 1955), as a starting point for local clustering.

The server's role is to aggregate the multiset $\widetilde{C} = \dot{\bigcup}_i C_i$ of all local centers communicated, using the objective

$$\textbf{Server:} \quad C \leftarrow \operatorname*{arg\,min}_{S, |S| = k} \sum_{c \in \widetilde{C}} \min_{s \in S} \|c - s\|_2 \ ,$$

that minimizes the overall k-median cost through client center contributions. Then, as detailed in Algorithm 1, we communicate the global estimates to clients, and repeat until convergence, which usually means a few communication rounds (e.g., 5).

Naturally, the first aggregation approaches that come to mind are k-means and k-median. However, they are not robust against significant attacks, even when combined with state-of-the-art mitigation procedures like costly Krum-based filtering (Blanchard et al., 2017). While more robust, k-medoids heuristics (Kaufman and Rousseeuw, 1990) are also not sufficiently resilient. Worse, they do not return metric medians, nor are they efficient in large-scale deployments. We, therefore, need an alternative that not only offers guarantees, robustness, and efficiency; but also returns metric medians. For this, we take the k-centers approach below.

4.1 Robust Metric k-Centers

Classical k-center heuristics, such as Gonzalez (1985), operate by greedily selecting the farthest point from a set $\widetilde{C}$. While this is efficient, and accurate when data is clear, it is susceptible to picking outliers when data is noisy. To make k-centers robust, we prevent selecting outliers by using the fact that honest clients concentrate around true center locations and thereby create dense regions. Using local density, we can then mitigate robustness issues. Formally, let $(\mathcal{X}, d)$ be a metric space, and let $C^* = \{c_1^*, \ldots, c_k^*\} \subset \mathcal{X}$ denote the latent true centers. The server observes candidate points $\widetilde{C} = \widetilde{C}_H \,\dot{\cup}\, \widetilde{C}_B$, where $\widetilde{C}_H$ are candidates produced by honest clients and $\widetilde{C}_B$ are arbitrary Byzantine candidates. For a fixed $\varepsilon > 0$, and for every true center c_j^*, assume that there are at least $m_j \in \mathbb{N}$ honest candidates within ε of c_j^*

$$\left|\{x \in \widetilde{C}_H : d(x, c_j^*) \le \varepsilon\}\right| \ = \ m_j \ \ge \ 1\ \forall j \in [k] \ .$$

A high density is a strong indicator for a consensus for a location of true centers from honest clients. As k-center methods only select centers from candidates $\widetilde{C}$, we need to aggregate these high-density neighborhoods into actual metric medians. In a nutshell, we first greedily identify k *core points* in high-concentration regions, and then aggregate their *cover* neighborhoods into k geometric medians which are then transmitted to the clients.

Cores To find core points, we employ a w-weighted variant of k-centers. Intuitively, we pull low-density candidates significantly closer (ideally $w_x \ll 1$), and push high-density regions further ($w_x \ge 1$), thereby discouraging unwanted non-core candidates. For given weights $w : \widetilde{C} \to \mathbb{R}_+$, the robust objective is

$$\min_{\substack{S \subseteq \widetilde{C} \\ |S| = k}} \ \max_{x \in \widetilde{C}} \ w_x \cdot \min_{c \in S} d(c, x) \ ,$$

which we solve by taking a greedy approach. Starting with the point $S_1 = \{\arg\max_{x \in \widetilde{C}} w_x\}$, we select subsequent cores using a *farthest-heaviest-first* heuristic

$$S_i \;\leftarrow\; S_{i-1} \cup \underset{x \in \widetilde{C} \setminus S_{i-1}}{\arg\max} \; w_x \cdot \min_{c \in S_{i-1}} d(c, x),$$

for weights w_x. To pick appropriate weights, we introduce a correctness condition below.

Lemma 1 (Robust Cores)**.** *Let the honest candidates be partitioned into nonempty sets $H_1, \ldots, H_k$. Let $D_H^{\min}$ be their minimum inter-cluster distance, $D_H^{\max}$ their maximum intra-cluster diameter, and $D^{\max}$ the diameter of the full candidate set. Write $w_H^{\min}$ and $w_H^{\max}$ for the minimum and maximum honest weights, and $w_B^{\max}$ for the maximum Byzantine weight. The algorithm selects exactly one core from each honest set H_j (i.e., $S_k \subset \widetilde{C}_H$ and $|S_k \cap H_j| = 1$ for all $j \in [k]$) if*

(1) The honest centers form well-separated clusters

$$w_H^{\min} \cdot D_H^{\min} > w_H^{\max} \cdot D_H^{\max} \,.$$

(2) The minimum honest score is greater than the maximum Byzantine score

$$w_H^{\min} \cdot D_H^{\min} > w_B^{\max} \cdot D^{\max} \,.$$

A good weight function assigns high scores to points in dense, honest regions and low scores to isolated or adversarial ones. This leads to weights proportional to the local density, such as *Krum* (Blanchard et al., 2017), the inverse average neighbor distance $w_x = |N_x| / \sum_{y \in N_x} d(x, y)$, and the median neighbor distance $w_x = \mathrm{median}_{y \in N_x} d(x, y)^{-1}$, where N_x denotes the k-nearest neighborhood of x. Due to perturbations from adversarial clients or privacy noise, the targeted metric centers are almost surely not in our candidate set $\widetilde{C}$. However, as honest candidates concentrate in dense regions around cores, the server should aggregate them.

Covers To find centers, we refine the k *cores* in $\widetilde{C}$ into robust geometric medians representing their clusters. Leveraging that honest points concentrate around true centers, we aggregate neighborhoods $N_s \subset \widetilde{C}$ of each core s, thereby computing a consensus cluster center. We define the cover of each core s as the L_2-ball $N_s = \mathbb{B}(s, r_s) \cap \widetilde{C}$, with radius r_s set to half the *separation distance* between core points $r_{\text{sep}} \leftarrow \frac{1}{2} \min_{t \in C \setminus s} d(s, t)$. While straightforward, aggregating neighborhoods N_s into metric centers may still yield high-variance estimates due to perturbations. To address this, we first trim extreme outliers by shrinking the radius r_s to a maximum $r_{\max}$, and limit the neighborhood to contain at most $k_{\max} \leq |N_s|$ points. Then, we aggregate the remaining candidates using a weighted variant of the geometric median (Minsker, 2015)

$$\mathrm{m}(S) \coloneqq \underset{t \in \mathbb{R}^d}{\arg\min} \sum_{s \in S} w_s \|t - s\|_2 \,.$$

Note that weighting does not improve the geometric median's intrinsic robustness, but improves the class of perturbations under which the majority condition holds. The geometric median is close to the true center whenever the in-ball weight fraction q is a strict majority; the precise weighted bound is stated next. In the weighted version, we control the honest *mass*, for which we use the following breakdown-point bound.

Lemma 2 (Cover bound)**.** *For ball $B(c, r)$, if $q > \frac{1}{2}$ then the geometric median satisfies $\|m - c\|_2 \leq \frac{2q}{2q-1}\, r$.*

Therefore, any cover whose honest weight fraction exceeds the breakpoint $1/2$ yields a center estimate within $\frac{2q}{2q-1}\, r = \mathcal{O}(r)$ of the true center. Lemma 2 aggregates any cover whose honest weight exceeds $1/2$ into an accurate center estimate. It does not guarantee that such a cover exists, which we address next.

Lemma 3 (Cover mass concentration)**.** *Let $\mathcal{B}$ be a finite family of N_{s} candidate balls and let $\{(x_i, w_i)\}_{i=1}^n$ be independent weighted samples with fixed nonnegative weights, total mass W, and effective sample size $n_{\text{eff}} = W^2 / \sum_i w_i^2$. For $B \in \mathcal{B}$, let $\widehat{\mu}(B) = W^{-1} \sum_i w_i \mathbf{1}\{x_i \in B\}$ and $\mu(B) = \mathbb{E}[\widehat{\mu}(B)]$ denote its empirical and population weighted mass, respectively. Assume that for every true center c_j^* there is a ball $B_j \in \mathcal{B}$ such that $B_j \subseteq \mathrm{B}(c_j^*, 2r)$ and $\mu(B_j) \geq p_{\min}$. For any $\rho \in (0, 1)$, if $n_{\text{eff}} \gtrsim \frac{\log(N_{\mathrm{s}}/\delta)}{\rho^2 p_{\min}^2}$, then with probability at least $1 - \delta$, every such ball satisfies $\widehat{\mu}(B_j) \geq (1 - \rho) p_{\min}$.*

Thus, Lem. 3 ensures that each true center has a nearby, high-mass cover candidate. This concentration result does not by itself imply an honest majority: when the Byzantine mass in each selected cover is smaller than the concentrated honest mass, the condition $q_j > 1/2$ holds and Lem. 2 applies. Under this majority condition, Lem. 2 converts the covers into bounded geometric-median estimates. We now state the resulting conditional convergence guarantee.

Theorem 4 (Convergence)**.** *Assume honest concentration and exactly one core per honest cluster. For each cover N_j, let n_j be its total mass and $h_j > n_j/2$ its honest mass, and define $\beta_j = \frac{2h_j}{2h_j - n_j}$ and $\beta_{\max} = \max_j \beta_j$. Suppose that, for every cluster and round,*

$$\|c_j^{(t+1)} - c_j^*\|_2 \leq \beta_j \mu \, \|c_j^{(t)} - c_j^*\|_2 + \beta_j (\varepsilon + \eta),$$

where $\mu \in [0, 1)$ is the local contraction factor, ε the honest-concentration radius, and η a perturbation bound. Set $\kappa \coloneqq \beta_{\max} \mu, R \coloneqq \beta_{\max}(\varepsilon + \eta)$, and $\Delta_t \coloneqq \max_j \|c_j^{(t)} - c_j^*\|_2$. If $\kappa < 1$ then $\Delta_{t+1} \leq \kappa\, \Delta_t + R$, and

Algorithm 2 CORK

Require: Candidate set $\widetilde{C}$, number of centers k.
Ensure: Metric centers $C \subseteq \mathbb{R}^d$ with $|C| = k$

```
Optionally trim obvious outliers from C̃
C ← {arg max_{x∈C̃} w_x}
for i = 2 to k do
    C ← C ∪ {arg max_{x∈C̃\C} w_x · min_{c∈C} d(c, x)}
end for
return {m(N_c) | ∀c ∈ C}
```

therefore

$$\limsup_{t\to\infty} \Delta_t \leq \frac{R}{1-\kappa} = \frac{\beta_{\max}(\varepsilon+\eta)}{1-\beta_{\max}\mu}.$$

Computational and communication complexity Let ℓ be the number of clients, n_i the local sample size at client i, k_i the number of local centers (with $\kappa = \sum_{i=1}^{\ell} k_i$), k the global target, d the data dimension, and T the number of communication rounds. Communication per round is $\mathcal{O}(\kappa d)$ real numbers (each client transmits k_i d-vectors); over T rounds the total is $\mathcal{O}(T\kappa d)$. Client work is dominated by the local k-median solver: using a Lloyd-style heuristic this costs $\mathcal{O}(T_{\text{loc}} \sum_i k_i n_i d)$ where T_{loc} is the number of local iterations; the Hungarian matching to pick k_i centers from k global centers costs $\mathcal{O}(\ell \cdot k^3)$ in the worst case. Our server's work is dominated by pairwise distance computations, the greedy weighted k-centers, and cover/median refinements. A naive implementation builds the full $\kappa\times\kappa$ distance matrix at cost $\mathcal{O}(\kappa^2 d)$, greedy core selection costs up to $\mathcal{O}(k\cdot\kappa)$ distance checks, and computing all geometric medians (Weiszfeld) costs $\mathcal{O}(T_{\text{gm}}\kappa d)$ where T_{gm} is the number of median iterations, which in practice depends on the number of covered elements and is typically dominated by the remaining operations. Thus, the server asymptotic bottleneck is $\mathcal{O}(\kappa^2 d + T_{\text{gm}}\kappa d)$. Using kd-trees, we reduce server costs to near $\mathcal{O}(\kappa \log \kappa)$ in typical regimes, if approximations are used and d is small.

5 EXPERIMENTS

Having ascertained that our method FORK works in theory, we now validate the practical performance. To this end, we seek to answer the following questions.

Q1 How well can FORK handle heterogeneous clients?
Q2 How robust is FORK against privacy noise?
Q3 How resilient is FORK against attacks?
Q4 How well does FORK stabilize local estimators?
Q5 How well does FORK perform on real-world data?
Q6 How scalable is FORK with client counts?

We compare median-weighted FORK with the state-of-the-art in federated clustering. k-FED (one-shot k-means (Dennis et al., 2021)) and FKM (few-shot k-means). Moreover, we compare to direct adaptations of centralized methods to the federated setup CSKM (federated core-set k-means (Bahmani et al., 2012)) and FKMED (few-shot k-median).

To benchmark against robust baselines, we introduce KRUMKMED, a novel trimmed k-median aggregation method that filters individual points—rather than entire clients—using a Q_n-estimator criterion on their *Krum scores*, resulting in an $\mathcal{O}(TM^2)$ computational cost, where M is the number of client-center candidates. Moreover, because simple weights might inflate when neighbors are extremely close, only a few candidates suffice to obfuscate outlier sets—which is easily exploitable by adversaries. To mitigate this, we pair FORK with the *reachability adjusted density* (LDR)

$$w_x = \left(\tfrac{1}{|N_k(x)|}\textstyle\sum_{y\in N_k(x)} \max\{d(x,y),\, d_k(y)\}\right)^{-\gamma},$$

where $N_k(x)$ denotes the k nearest neighbors of x and $d_k(y)$ the k-distance of y, for a user-defined scale $\gamma \geq 0$ typically 1 (Breunig et al., 2000).

Whenever ground-truth labels are known, we evaluate the clustering quality using the *adjusted Rand index* (ARI) (Santos and Embrechts, 2009). We also report the *Calinski–Harabasz index* (CHI) and the normalized *root-mean-square center error* as

$$\text{MCE} \coloneqq \frac{1}{D}\sqrt{\frac{1}{k}\sum_{j=1}^{k} \|\widehat{c}_j - c_j^*\|_2^2},$$

where D is the data diameter and the estimated and reference centers are optimally matched by minimum squared Euclidean distance before evaluation. We repeat each experiment for 10 independent trials and report the mean and standard deviation. For full reproducibility, we share all details, including code, in the online supplementary material and our appendix.[1] We also include additional experiments in *Sec. B*.

5.1 Synthetic

To answer Q1–Q5, we need precise control over data, for which we make use of a synthetic data generator, allowing us to control cluster sizes, separation, noise, and adversarial characteristics. In a nutshell, we first sample well-separated centers on a hypercube and then sample Gaussian points around these centers with known correlation and cluster-specific standard deviations. Clients can receive either IID or non-IID subsets of clusters. In the IID case, each client has samples from all centers, in the non-IID case, each

[1] `https://doi.org/10.5281/zenodo.19298175`

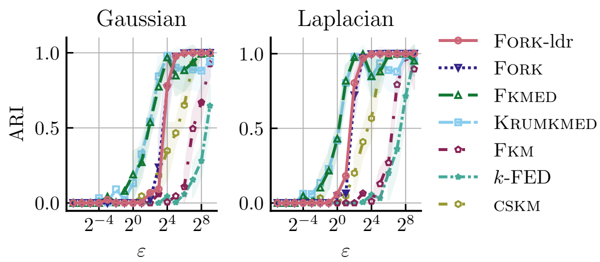

Figure 1: Our method is privacy preserving and resilient against center perturbations. For varying privacy budgets, we show the ARI on synthetic data.

client mainly covers samples from a fraction of centers (varying in our experiments). Then, we add a fraction of outliers, or Byzantine clients modify local data, if required. In most synthetic experiments, we consider 100 clients, each with 100 samples in 10 dimensions, and 5 cluster centers.

Byzantine Next, we study robustness of our method. For this, we simulate the following failure modes and attack types. *Random centers* contribute uniformly random points $c_{ij} \sim \mathcal{U}(\mathcal{B})$ within some bounded region $\mathcal{B}$. *Outlier centers* contribute points c_{ij} sampled to have significantly different distributions than the data distribution. *Out-of-manifold centers* are in low-density regions far from the true data manifold. *Mirrored centers* form an attack in which Byzantine clients flip true centers across decision boundaries, i.e., $c_{ij} = 2\mu - c^*_{ij}$ for some reference point μ. We then use these attacks to create *adversarial clients* that communicate perturbed information (which may vary each round), and clients that analyze *corrupted data* (which is constant across communication rounds). We vary the fraction of adversarial clients or corrupted data from 0 to 80%.

First, we consider **model poisoning** through *adversarial model behavior* of a fraction of clients. Fig. 3 shows that, across the board, all our variants return accurate solutions, even when confronted with numerous adversarial clients. Competing methods, however, degrade even under mild attack. In regimes exceeding $1/3$, adversarial clients create artificial high-density mirror centers, which are indistinguishable from true centers. Then, the quality drops to baseline level for all methods, including ours. Next, we consider **data poisoning**, where some clients operate on corrupted data. Fig. 2 shows that Fork almost always identifies the correct solution, while competing methods still fail.

Privacy Next, we explore center perturbations through *privacy noise* $\tilde{c}_{ij} = c_{ij} + \eta_{ij}$ with $\eta_{ij} \sim \mathcal{N}(0, \sigma^2\mathbf{I})$. For Gaussian and Laplacian DP-mechanisms with calibrated sensitivity estimate Sec. 3, we vary the budget ε, compounded over all communication rounds. When the noise is extreme, the server's input is indistinguishable from noise, thereby reducing performance significantly. In Fig. 1, we see that cluster centers become identifiable under a mild budget. While Fork is a bit more conservative than Fkm and Fkmed, it swiftly increases the ARI even for low to moderate budgets. On the other hand, to achieve a similar performance, cskm and k-FED require a significantly higher privacy budget.

Outliers Now, we explore how clients benefit from aggregating solutions into a global set of centers. That is, whenever clients may struggle to identify the correct locations, do they benefit from sharing information from other clients? To this end, we introduce outliers to each local dataset (in contrast to the past where we introduced outliers to a fraction of clients). The more outliers there are, the harder it becomes to identify good local centers using a local k-median. Additionally, when the majority of clients are uncertain about a location, the consensus at the server also suffers. Therefore, we test limits and benefits by adding a varying number of outliers to each client. In Fig. 4 we show ARI, CHI, and MCE for a varying fraction of outliers per client. We see that Fork outperforms all baselines across the board not only in ARI, but it also exhibits the lowest center displacement. While in previous experiments, Krumkmed was capable of handling adversaries and corruption well, here it falls far behind Fork.

Scalability To study scalability with no Byzantine clients and IID data, we increase the number of clients from 2 to 2048, while either keeping the sample size per client fixed (*data abundance*), or we distribute a fixed number of points to an increasing number of clients (*data scarcity*). In Fig. 5, we show ARI and runtime for data abundance, data scarcity, and participation. There, we see that Fork efficiently scales even to large numbers of clients under data abundance and scarcity. Robust aggregation adds negligible overhead.

Furthermore, in the *participation* setting, we fix the number of clients and randomly vary how many contribute local results in each communication round. With 1/3 Byzantine clients and a non-IID fraction of 1/3, reduced participation first diminishes the influence of Byzantine clients, while heterogeneity dominates when participation sufficiently becomes low.

5.2 Real World

Assured that Fork works well on synthetic data, we evaluate it on 7 real-world datasets from 3 different domains, annotating targeted cluster count and client count by *(k/client count)* below. In computer vi-

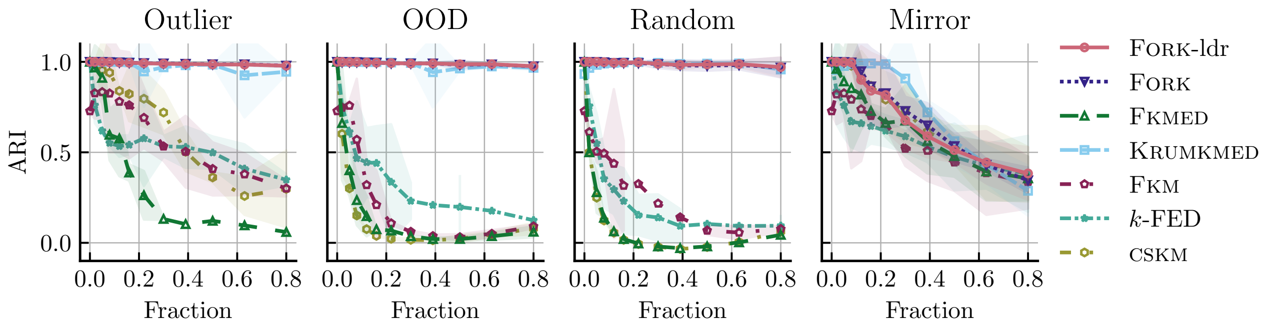

Figure 2: Our method is robust against compromised client data. We see that our method returns the best (often perfect) results. We show ARI for different ways of compromising the local data-generating process (outlier, OOD, random, mirror) and for fractions of compromised clients up to 80%.

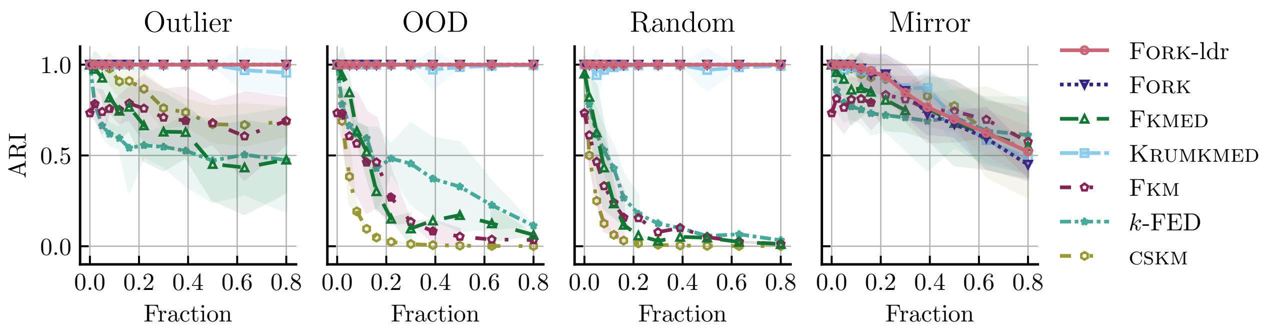

Figure 3: Our method is robust against adversarial client behavior. Fork demonstrates the highest performance across the board, often achieving perfect solutions. We show ARI for different adversarial behaviors (outlier, OOD, random, mirror), and for varying adversarial client fractions of up to 80%.

sion, we use *CIFAR-10* (Krizhevsky and Hinton, 2009), *Fashion MNIST* (Xiao et al., 2017), and *MNIST* (LeCun and Cortes, 2005) (all 10/100). For natural language processing, we use *ArXiv* (ArXiv.org Collaboration, 2025) (30/50) abstracts from the *cs.LG* category (as 384-dimensional word embeddings), and *20 Newsgroups* (Mitchell, 1997) (20/50; 768-dimensional embeddings). In the life sciences, we include gene-expression data from *TCGA* (National Cancer Institute, 2005) (33/10), and single-cell RNA sequencing data from *CELLxGENE* (Abdulla et al., 2024) (20/50). All datasets were standardized, and we reduced the dimensionality of high-dimensional datasets (*TCGA*, *CIFAR 10*, and *CELLxGENE*) using PCA to retain 90% variance or at most 100 dimensions.

Here, we consider non-IID data partitioning. That is, for each client, we randomly select half of the classes as preferred. Then we sample with a bias (here exclusively) in favor of picking data points that originally belonged to that class until all data points have been distributed to all clients.

In Tab. 1, we show the average Calinski-Harabasz Index (CHI). Across diverse real-world datasets, using a median weight, Fork consistently achieves competitive or top CHI scores, often outperforming baseline federated clustering methods. The LDR variant of Fork performs similarly. Although k-FED employs a greedy strategy, it struggles to distinguish between outliers and core points, which compromises the aggregation.

Table 1: Our method outperforms the competition in federated clustering on real-world data with non-IID partitioning. We show the average CHI over multiple trials.

Dataset	k-FED	Fkmed	Fkm	Fork	Fork-ldr	cskm
ArXiv	534.9	537.7	547.3	**561.1**	555.0	377.7
C×G	978.9	882.6	922.7	**1064.1**	1054.8	823.5
CIFAR 10	84.0	41.3	143.2	**163.7**	159.1	116.3
FMNIST	8961.0	7771.4	9468.9	9321.8	9257.5	**9512**
MNIST	91.5	100.4	**110.1**	105.2	97.3	109.6
20 News	475.8	477.8	479.6	**488.6**	482.3	349.7
TCGA	226.9	219.4	**229.4**	229.2	225.3	164.4

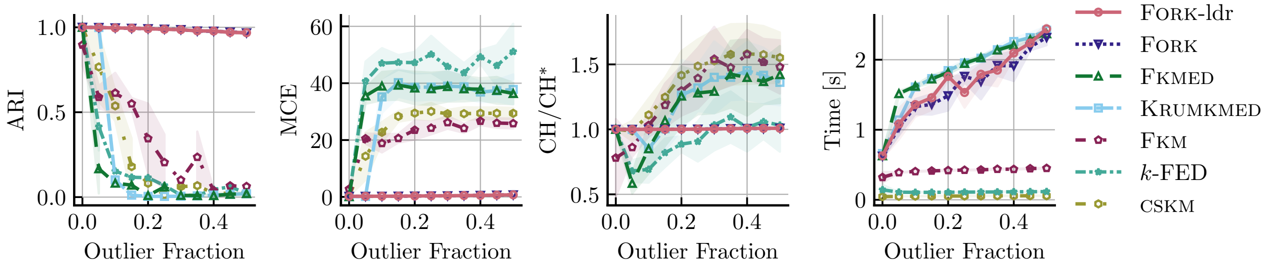

Figure 4: Our method is resilient against compromised client data. FORK almost always robustly estimates global centers, even when local data is highly compromised, while the competition fails. For a growing fraction of up to 50% outliers per client data, we show ARI, MCE, CHI, and runtime (seconds).

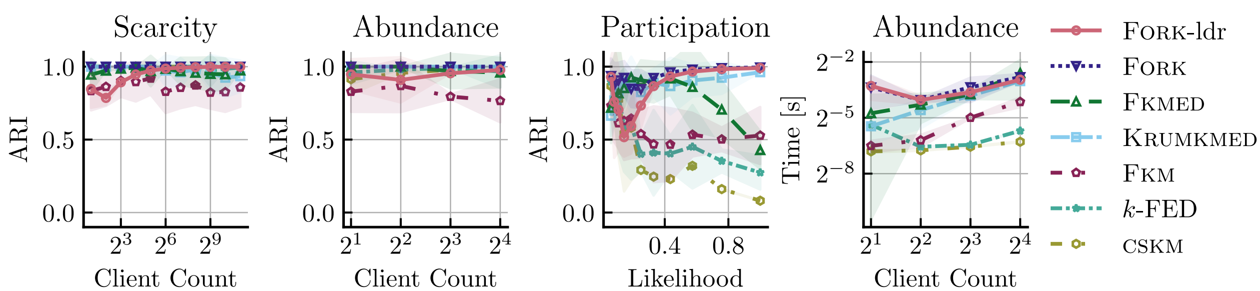

Figure 5: Our method is scalable to high client counts and participation rates. We show the performance on synthetic data. First, we vary client counts (with growing, *abundant* data or constant, *scarce* data), and second, we vary the fraction of *participating* clients per communication round.

5.3 Discussion

We introduced a new federated clustering algorithm, FORK, that addressed the failure modes of adversarial clients, compromised data, and heterogeneous distributions. We developed a robust aggregation algorithm that exploited the agreement regarding center locations among honest clients to efficiently reconstruct global centers. Our approach enhances the robustness of the farthest-first heuristic against adversarial conditions, yielding core points whose covers we aggregate with robust geometric medians. We provided formal guarantees for identifiability, robustness, and convergence. Through experiments, we confirmed low runtime, high resilience, and accurate clusters; our method consistently outperformed existing approaches. FORK achieved stronger robustness, lower computational complexity, and theoretical guarantees compared to the state of the art.

Despite these benefits, our method has limitations. It requires a sufficiently high concentration of honest points to reliably recover clusters; when adversarial clients are highly coordinated, *all methods*, including FORK, struggle to separate true clusters from artificial ones. In such cases, additional signals (such as side information, cross-client consistency checks, or domain priors) will be necessary to achieve reliable results. In future work, we aim to explore how to integrate such signals. While our work focused on server-side robustness, improving client-side algorithms (e.g., local filtering, weighting, or privacy-aware consistency tests) might further improve resilience.

In sum, we proposed a federated clustering framework that is robust, efficient, and theoretically grounded, paving the way for secure and collaborative data analysis, and the development of new robust federated clustering algorithms.

Acknowledgments This work was conducted while Martín Bravo was visiting KTH Royal Institute of Technology. This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The computations were partially supported by the National Academic Infrastructure for Supercomputing in Sweden (NAISS).

References

Abdulla, S., Aevermann, B., Assis, P., Badajoz, S., Bell, S. M., Bezzi, E., Cakir, B., Chaffer, J., Chambers, S., Cherry, J., Chi, T., Chien, J., Dorman, L., Garcia-Nieto, P., Gloria, N., Hastie, M., Hegeman, D., Hilton, J., Huang, T., Infeld, A., Istrate, A.-M., Jelic, I., Katsuya, K., Kim, Y. J., Liang, K., Lin, M., Lombardo, M., Marshall, B., Martin, B., McDade, F., Megill, C., Patel, N., Predeus, A., Raymor, B., Robatmili, B., Rogers, D., Rutherford, E., Sadgat, D., Shin, A., Small, C., Smith, T., Sridharan, P., Tarashansky, A., Tavares, N., Thomas, H., Tolopko, A., Urisko, M., Yan, J., Yeretssian, G., Zamanian, J., Mani, A., Cool, J., and Carr, A. (2024). Cz cellxgene discover: a single-cell data platform for scalable exploration, analysis and modeling of aggregated data. *Nucleic Acids Research*, 53(D1):D886–D900.

Arthur, D. and Vassilvitskii, S. (2006). k-means++: The advantages of careful seeding. Technical report, Stanford.

ArXiv.org Collaboration (2025). arxiv dataset and metadata of 1.7m+ scholarly papers across stem.

Bahmani, B., Moseley, B., Vattani, A., Kumar, R., and Vassilvitskii, S. (2012). Scalable k-means++. *arXiv preprint arXiv:1203.6402*.

Blanchard, P., El Mhamdi, E. M., Guerraoui, R., and Stainer, J. (2017). Machine learning with adversaries: Byzantine tolerant gradient descent. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, page 93–104, New York, NY, USA. Association for Computing Machinery.

Charikar, M., Steinhardt, J., and Valiant, G. (2017). Learning from untrusted data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 47–60, New York, NY, USA. Association for Computing Machinery.

Chawla, S. and Gionis, A. (2013). k-means-: A unified approach to clustering and outlier detection. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics.

Cuesta-Albertos, J. A., Gordaliza, A., and Matrán, C. (1997). Trimmed k-means: An attempt to robustify quantizers. *The Annals of Statistics*, 25(2):553–576.

Dalleiger, S. and Gionis, A. (2025). Creating coherence in federated non-negative matrix factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16135–16143.

Dalleiger, S., Vreeken, J., and Kamp, M. (2025). Federated binary matrix factorization using proximal optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 16144–16152.

Dennis, D. K., Li, T., and Smith, V. (2021). Heterogeneity for the win: One-shot federated clustering. *Proceedings of the 38th International Conference on Machine Learning*, 139:2611–2620.

Dwork, C., Roth, A., et al. (2014). The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3–4):211–407.

García-Escudero, L. A., Gordaliza, A., Matrán, C., and Mayo-Iscar, A. (2008). A general trimming approach to robust cluster analysis. *The Annals of Statistics*, 36(3).

Garst, S. and Reinders, M. (2024). Federated k-means clustering. *International Conference on Pattern Recognition*, pages 107–122.

Ghosh, A., Chung, J., Yin, D., and Ramchandran, K. (2020). An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 19586–19597. Curran Associates, Inc.

Gonzalez, T. F. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306.

Huang, L., Li, Z., Sun, J., and Zhao, H. (2022). Coresets for vertical federated learning: Regularized linear regression and k-means clustering. *Advances in Neural Information Processing Systems*, 35:29566–29581.

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.

Karimireddy, S. P., Kale, S., Mohri, M., Reddi, S., Stich, S., and Suresh, A. T. (2020). SCAFFOLD: Stochastic controlled averaging for federated learning. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5132–5143. PMLR.

Kaufman, L. and Rousseeuw, P. J. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley.

Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario.

Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97.

LeCun, Y. and Cortes, C. (2005). The MNIST database of handwritten digits.

Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., and Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine learning and systems*, 2:429–450.

McMahan, B., Moore, E., Ramage, D., Hampson, S., and Agüera y Arcas, B. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Minsker, S. (2015). Geometric median and robust estimation in banach spaces. *Bernoulli*, 21(4).

Minsker, S. and Strawn, N. (2024). The geometric median and applications to robust mean estimation. *SIAM Journal on Mathematics of Data Science*, 6(2):504–533.

Mitchell, T. (1997). Twenty Newsgroups. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C5C323.

National Cancer Institute (2005). The cancer genome atlas program (tcga).

Nguyen, H. L., Chaturvedi, A., and Xu, E. Z. (2021). Differentially private k-means via exponential mechanism and max cover. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):9101–9108.

Pan, C., Sima, J., Prakash, S., Rana, V., and Milenkovic, O. (2023). Machine unlearning of federated clusters. In *The Eleventh International Conference on Learning Representations*.

Pillutla, K., Kakade, S. M., and Harchaoui, Z. (2022). Robust Aggregation for Federated Learning. *IEEE Transactions on Signal Processing*, 70:1142–1154.

Santos, J. M. and Embrechts, M. (2009). On the use of the adjusted rand index as a metric for evaluating supervised classification. In *International conference on artificial neural networks*, pages 175–184. Springer.

Scott, J., Lampert, C. H., and Saulpic, D. (2025). Differentially private federated k-means clustering with server-side data. In *Forty-second International Conference on Machine Learning*.

Xiao, H., Rasul, K., and Vollgraf, R. (2017). Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

Yin, D., Chen, Y., Kannan, R., and Bartlett, P. (2018). Byzantine-robust distributed learning: Towards optimal statistical rates. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 5650–5659. PMLR.

Zheng, L., Zhu, Y., and He, J. (2023). *Fairness-aware Multi-view Clustering*, pages 856–864. SDM.

Zhu, S., Xu, Q., Zeng, J., Wang, S., Sun, Y., Yang, Z., and Peng, Z. (2023). F3km: Federated, fair, and fast k-means. *Proceedings of the ACM on Management of Data (SIGMOD)*, 1(4):Article 241.

Checklist

1. For all models and algorithms presented, check if you include:
   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. Yes
   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. Yes
   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. Yes
2. For any theoretical claim, check if you include:
   (a) Statements of the full set of assumptions of all theoretical results. Yes
   (b) Complete proofs of all theoretical results. Yes
   (c) Clear explanations of any assumptions. Yes
3. For all figures and tables that present empirical results, check if you include:
   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). Yes
   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). Yes
   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). Yes
   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). Yes
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:
   (a) Citations of the creator If your work uses existing assets. Yes
   (b) The license information of the assets, if applicable. Yes
   (c) New assets either in the supplemental material or as a URL, if applicable. Yes
   (d) Information about consent from data providers/curators. Yes
   (e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. Not Applicable
5. If you used crowdsourcing or conducted research with human subjects, check if you include:
   (a) The full text of instructions given to participants and screenshots. Not Applicable
   (b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. Not Applicable
   (c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. Not Applicable

Appendix

A ADDITIONAL PROOFS

Proof of Lem. 1. Let $\widetilde{C} = \widetilde{C}_H \dot\cup \widetilde{C}_B$ and let $H_1, \ldots, H_k$ be the honest clusters. Define

$$D_H^{\max} := \max_{j\in[k]} \sup_{x,y\in H_j} d(x,y), \qquad D_H^{\min} := \min_{\substack{j,\ell\in[k]\\ j\neq\ell}} \inf_{x\in H_j,\, y\in H_\ell} d(x,y),$$

$$D^{\max} := \max_{x,y\in\widetilde{C}} d(x,y).$$

Assume honest weights satisfy $w_x \in [w_H^{\min}, w_H^{\max}]$ for all $x \in \widetilde{C}_H$ and Byzantine weights satisfy $w_x \leq w_B^{\max}$ for all $x \in \widetilde{C}_B$.

We prove by induction on the selection index i that after i selections the chosen set S_i contains at most one point from each honest cluster and that every selected point is honest.

Base ($i = 1$) The first chosen point is $S_1 = \arg\max_x w_x$. Because $D^{\max} \geq D_H^{\min} > 0$, condition (2) of Lem. 1 implies $w_H^{\min} > w_B^{\max}$; hence the maximizer lies in $\widetilde{C}_H$. Hence, S_1 is honest.

Induction Suppose S_{t-1} contains honest points and contains at most one point from each honest cluster. Consider any honest cluster H_j that has not yet contributed a point to S_{t-1}. For any $x \in H_j$ and any $c \in S_{t-1}$ we have $d(c,x) \geq D_H^{\min}$ by definition of the inter-cluster gap. Therefore,

$$\max_{x\in H_j} w_x \min_{c\in S_{t-1}} d(c,x) \;\geq\; w_H^{\min}\, D_H^{\min}.$$

Now consider any point y that is either in a cluster already represented in S_{t-1} or Byzantine. If y lies in an already represented cluster, the distance to the representing core is at most the intra-cluster diameter, so

$$w_y \min_{c\in S_{t-1}} d(c,y) \;\leq\; w_H^{\max}\, D_H^{\max}\ .$$

If y is Byzantine then it has to hold that

$$w_y \min_{c\in S_{t-1}} d(c,y) \;\leq\; w_B^{\max}\, D^{\max}\ .$$

By the two strict inequalities in the lemma, every unselected honest cluster achieves strictly larger weighted distance than any represented honest cluster point or any Byzantine point

$$w_H^{\min} D_H^{\min} \;>\; \max\{w_H^{\max} D_H^{\max},\; w_B^{\max} D^{\max}\}\ .$$

Consequently, the $\arg\max$ in the greedy rule picks some point from an honest cluster not yet represented, adding exactly one new honest core. This preserves the induction hypothesis.

Therefore, after k iterations we have selected exactly one point from each honest cluster. □

The following proof is inspired by Minsker and Strawn (2024). For the setting of Lem. 2, write $\{(x_i, w_i)\}_{i=1}^n \subset \mathbb{R}^d \times \mathbb{R}_+$, with total weight $W = \sum_{i=1}^n w_i > 0$, and define the *weighted geometric median*

$$m \in \arg\min_{z\in\mathbb{R}^d} \sum_{i=1}^n w_i \|z - x_i\|_2\ .$$

For a fixed reference point $c \in \mathbb{R}^d$ and radius $r > 0$, let $H = \{i : \|x_i - c\|_2 \leq r\}$ denote the indices inside the ball $B(c,r)$ and define the *honest weight fraction* $q = \frac{1}{W}\sum_{i\in H} w_i$. When $q > 1/2$, the claimed bound is

$$\|m - c\|_2 \leq \frac{2q}{2q-1}\, r\ .$$

Proof of Lem. 2. Let $F(z) = \sum_{i=1}^{n} w_i \|z - x_i\|_2$, and because m minimizes F, we have $F(m) \leq F(c)$. Define $d = \|m - c\|_2$. We derive lower and upper bounds on $F(m) - F(c)$.

Case "inside the ball": For $i \in H$, and using the triangle inequality we get

$$\|m - x_i\|_2 \geq \|m - c\|_2 - \|x_i - c\|_2 \geq d - r \ .$$

Hence, through substituting, we obtain $\sum_{i \in H} w_i \|m - x_i\| \geq W_H(d - r)$ and $\sum_{i \in H} w_i \|c - x_i\| \leq W_H r$, where $W_H = \sum_{i \in H} w_i = qW$. The contribution difference therefore satisfies

$$\sum_{i \in H} w_i(\|m - x_i\| - \|c - x_i\|) \geq W_H(d - 2r) \ .$$

Case "outside the ball": For $i \notin H$, *reverse* triangle inequality yields $\|m - x_i\|_2 \geq \|c - x_i\|_2 - d$. Therefore, $\sum_{i \notin H} w_i(\|m - x_i\| - \|c - x_i\|) \geq -W_B d$, where $W_B = W - W_H = (1 - q)W$.

Summing both parts gives $F(m) - F(c) \geq W_H(d - 2r) - W_B d$. By using $F(m) \leq F(c)$, we get $0 \geq W_H(d - 2r) - W_B d$, and through substituting $W_H = qW$, $W_B = (1 - q)W$, we derive $0 \geq qW(d - 2r) - (1 - q)Wd$. Divide by $W > 0$ yields $0 \geq (2q - 1)d - 2qr$.

Since $q > 1/2$ by assumption, we obtain $d \leq \frac{2q}{2q-1} r$, and therefore,

$$\|m - c\|_2 \leq \frac{2q}{2q - 1} r \ .$$

$\square$

Proof of Lem. 3. For any fixed $B \in \mathcal{B}$ and $t > 0$, weighted Hoeffding concentration gives

$$\mathbb{P}(|\widehat{\mu}(B) - \mu(B)| > t) \leq 2 \exp\bigl(-2 n_{\text{eff}} t^2\bigr) \ .$$

Setting $t = \rho p_{\min}$ and applying a union bound over $\mathcal{B}$ yields

$$\mathbb{P}(\exists B \in \mathcal{B} : |\widehat{\mu}(B) - \mu(B)| > \rho p_{\min}) \leq 2 N_{\text{s}} \exp\bigl(-2 n_{\text{eff}} \rho^2 p_{\min}^2\bigr) \ .$$

The right-hand side is at most δ when

$$n_{\text{eff}} \geq \frac{\log(2 N_{\text{s}}/\delta)}{2 \rho^2 p_{\min}^2}.$$

On this event, every assumed ball B_j satisfies

$$\widehat{\mu}(B_j) \geq \mu(B_j) - \rho p_{\min} \geq (1 - \rho) p_{\min},$$

and its containment in $\mathrm{B}(c_j^*, 2r)$ holds by assumption. $\square$

Proof of Thm. 4. Let $\Delta_t = \max_j \|c_j^{(t)} - c_j^*\|_2$. Taking the maximum of the assumed per-cluster inequality gives $\Delta_{t+1} \leq \kappa \Delta_t + R$. Iterating this recursion yields

$$\Delta_t \leq \kappa^t \Delta_0 + R \sum_{s=0}^{t-1} \kappa^s = \kappa^t \Delta_0 + R \frac{1 - \kappa^t}{1 - \kappa}.$$

Because $\kappa < 1$, taking the limit superior gives $\limsup_{t \to \infty} \Delta_t \leq R/(1 - \kappa)$. $\square$

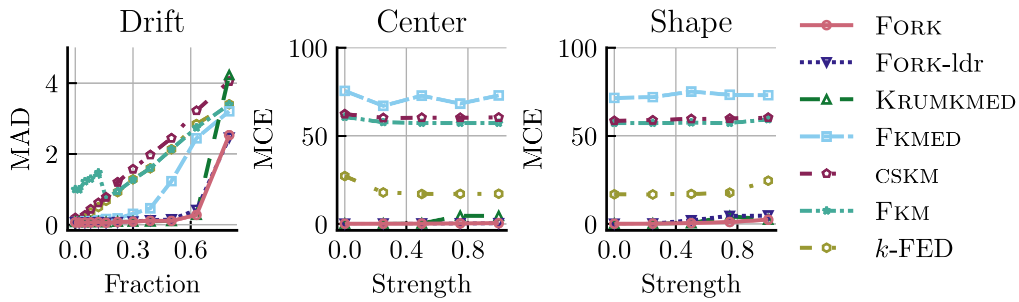

Figure 6: FORK is robust under temporally structured drift attacks and distributional heterogeneity. We show the median center displacement (MAD) for increasing Byzantine drift fractions of up to 80% (m=200 non-IID clients, 50 rounds). Then, we depict MCE to the true centers for increasing center perturbation strength (domain shift fixed at 0.2) and increasing shape-shift strength (center perturbation fixed at 0.1), with Byzantine fraction 1/3.

B ADDITIONAL EXPERIMENTS

Byzantine Robustness under Drift Attacks We study a temporal *drift* attack, in which Byzantine clients continuously shift their communicated centers across rounds with increasing strength, thereby attempting to slowly move the centers until they are "out of place" upon convergence. We use $m = 200$ non-IID clients with shared cluster fraction 1/2, run for up to 50 communication rounds, and sweep the Byzantine fraction from 0 to 80%. In Fig. 6, we report the median absolute center displacement (MAD). FORK consistently recovers accurate centers across all Byzantine fractions, whereas competing methods degrade rapidly once the drift accumulates over rounds.

Non-IID Heterogeneity under Center Perturbation and Shape Shifts We further evaluate robustness against distributional shifts in the data generation. In the *center perturbation* experiment, we displace the local cluster centroids of individual clients by a varying strength $s \in \{0, 0.25, 0.5, 0.75, 1.0\}$, while holding the domain shift fixed at 0.2. In the *shape shift* experiment, we deform the local data manifold by a varying strength $s \in \{0, 0.25, 0.5, 0.75, 1.0\}$, while fixing the center perturbation at 0.1. To shift, we stretch the cluster shapes along a random direction while keeping centers fixed. Both settings use $m = 200$ non-IID clients with a fixed Byzantine fraction of 1/3. We report MCE to the true centers, which uses one-to-one center matching and directly reflects localization quality under geometric distortion. As shown in Fig. 6, FORK maintains accurate center estimates across the full range of perturbation and shift strengths, while competing methods exhibit substantial degradation already at moderate heterogeneity levels.

C HYPERPARAMETERS

Our experiments use the following configuration. For synthetic data generation, we set the number of clients to $m = 100$ (varied in scalability experiments from 2 to 2048), local dataset size $n = 100$ points per client, number of clusters $k = 5$, and dimensionality $d = 10$ (varied from 4 to 1024 in dimensionality experiments). Cluster standard deviation is $\sigma = 1.0$ with cluster separation distance of 5.0 and cluster imbalance ratio up to 16 (maximum cluster size divided by minimum cluster size). For non-IID settings, we vary the shared cluster fraction from 0.0 to 1.0. Byzantine attacks are tested with fractions from 0.0 to 0.8 using four attack types: random, outlier, flip, and out-of-manifold. For differential privacy experiments, we vary ϵ from 2^{-7} to 2^{9} using Gaussian and Laplacian mechanisms. We average our results over 10 independent seeds. We use $k_{\text{nn}} = 6$ nearest neighbors and IQR threshold factor $f = 1.25$ to estimate the outlier fraction. Our runs use at most 20 communication rounds and generally converge within 1–2 rounds.

Real-world datasets (*CIFAR-10, Fashion-MNIST, MNIST, TCGA, 20-Newsgroups, ArXiv, CELLxGENE*) are standardized and, above 100 dimensions, reduced by PCA to at most 100 components while retaining up to 90% variance. We use dataset-specific values for $k \in [10, 33]$ and $m \in [10, 100]$ to reflect the number of ground-truth classes and dataset size.