FEDERATED CLUSTERING WITH UNKNOWN LOCAL AND GLOBAL CLUSTER CARDINALITIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated clustering methods that do not require the global number of clusters K still assume that each client knows its local number K_g. This assumption is hard to justify when clients know no more about their data than the server does, as in fault diagnosis across independently operated industrial sites. We propose a two-phase framework in which neither count is known: each client first estimates K_g from its own data, and an aggregator that requires local counts, such as FedGEM, then uses these estimates in place of the true values. For the first phase we introduce Adaptive Split–Merge (ASM), which grows a spherical Gaussian mixture by BIC-driven splitting and then merges excess components. ASM uses no labels, selects its hyperparameters on held-out client data only, and makes no assumption about how clusters are shared across clients. We derive a closed-form split criterion whose critical cluster size falls with anisotropy and rises with dimension, and show empirically that over-fragmentation grows with the number of points per cluster, which federation divides among clients. Across eight datasets, ASM with FedGEM attains a mean ARI of 0.333, against 0.256 for the next best label-free estimator and 0.361 when the true local counts are supplied. It also gives the most reliable global estimates of K and is robust when client size is decoupled from local cardinality.

1 INTRODUCTION

Federated clustering recovers cluster structure from unlabeled data held by clients that cannot pool it for reasons of privacy, regulation or cost. Each client can find structure in its own data, but the server must decide which of these locally discovered structures correspond to the same global cluster, without ever seeing the data behind them.

A concrete instance arises for original equipment manufacturers (OEMs) of capital-intensive industrial systems such as power generators, which are contractually bound to guarantee the reliability of equipment operated by their clients (Ibrahim et al., 2026). Meeting these guarantees requires the OEM to detect and diagnose faults from sensor data collected across client sites, and the problem is unsupervised and federated by necessity. The OEM does not know every fault class in advance. Labels supplied by clients are unreliable because labeling standards and maintenance practices differ from site to site. Raw sensor data cannot be shared. FedGEM (Ibrahim et al., 2026) addresses this setting without requiring the global number of clusters K, but it still assumes that each client knows its own local number K_g. However, it is reasonable to assume that the plant operators know no more about the equipment's fault classes than the OEM that designed it, so the reasons K is unknown apply equally to every K_g.

The same independence constrains how any method can be tuned. Each site operates its equipment under its own maintenance regime and has no access to other sites' data, so hyperparameters cannot be tuned jointly, either across sites or between a site and the OEM. Sites can, however, exchange information with the OEM at the level of parameters and summaries, repeatedly if needed. This naturally separates the problem into two phases: each site first decides on its own how finely to describe its data, and the OEM then reconciles these descriptions across sites through repeated exchange of parameters.

The granularity with which a site describes its data matters in two ways. Clients can get their own class-counts wrong, and what the server can do about it depends on the direction of the error. If a

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

client reports two components where one was present, the server can merge them from the reported parameters alone. If it reports one where two were present, no correction is possible: many different pairs of components share the same mean and covariance, and resolving the ambiguity would require a change in the information sharing structure. That argues for reporting finely, but the same quantity governs disclosure: a site reporting a single component reveals little beyond the gross statistics of its operation, while one reporting a component per measurement has shared the raw data it was contractually unable to send. The number of reported components therefore sits on a continuum, with recoverability increasing along it and confidentiality decreasing, and the first phase must place itself deliberately between the two. Finally, how many points a site holds may or may not reflect how many clusters it holds, depending on how its data were collected (for example, periodic logging, event-triggered logging, curated logging such as during diagnostic tests etc.), so an estimator of K_g must not rely on client size as a proxy.

We propose a two-phase framework in which each client first estimates K_g from its own data and a federated aggregator that requires local counts, such as FedGEM, then takes these estimates in place of the true values. For the first phase we introduce the Adaptive Split–Merge (ASM) estimator, which grows a spherical Gaussian mixture (and thus, ASM-S) by recursive binary splitting: each candidate split is initialized along the principal direction, as in G-means (Hamerly & Elkan, 2003), fitted by EM, and accepted when it improves the Bayesian Information Criterion, as in X-means (Pelleg & Moore, 2000). ASM adds a merge stage that undoes excess splits, and selects all hyperparameters on held-out client data, without labels.

Contributions.

- **A two-phase framework for federated clustering in which neither the local nor the global number of clusters is known.** The framework is modular in both phases: any centralized method that estimates the number of clusters can serve as the local estimator, and any federated aggregator that requires local counts, such as FedGEM, can serve as the second phase, without modification to either.
- **ASM, a label-free local cardinality estimator** that combines BIC-driven splitting with a merge stage and selects its hyperparameters on each client's own held-out data (Section 4).
- **An analysis of when BIC splitting fragments a single cluster.** A closed-form criterion gives a critical cluster size that falls with anisotropy and rises with dimension. Synthetic experiments confirm it once small-sample inflation of anisotropy is accounted for and show that, beyond the critical size, splitting continues until the minimum-mass constraint binds. On pooled benchmarks, sample size governs the over-fragmentation of the split stage, and a bound derived from class anisotropy predicts where merging begins on seven of eight datasets (Section 4).
- **A federated evaluation that decouples client size from local cardinality.** Across eight datasets, ASM-S with FedGEM attains a mean ARI of 0.333, against 0.256 for the next best label-free method and 0.361 when the true local counts are supplied. It gives the most reliable global estimates, with $|\hat{K} - K^*| \leq 21$ against up to 188 and 263 for DP-GMM and G-means, and is unaffected when client size is made independent of K_g (Section 5).

2 Related work

Table 1 summarizes the methods discussed below by what each requires as input.

Federated clustering with known K. k-FED (Dennis et al., 2021) runs k-means on each client and clusters the pooled centroids into K global clusters in a single round. FedKmeans (Garst & Reinders, 2024) performs federated Lloyd iterations under a shared K. The federated fuzzy c-means methods FFCM-avg1 and FFCM-avg2 (Stallmann & Wilbik, 2022) use fuzzy memberships and differ in how local prototypes are averaged. All of these require K at the server.

Federated clustering with unknown K. AFCL (Zhang et al., 2025) initializes each client with more seed points than clusters and lets redundant seeds merge through interaction with the server, so that the number of distinct clusters emerges as a by-product; clients may also communicate with

Table 1: Clustering methods by the cardinality information each requires. † AFCL does not take K_g, but its initial number of seeds is set with reference to K. ‡ ASM fits spherical Gaussian components on each client; the global cluster model is that of the second-phase aggregator.

Method	Federated	Unknown K	Unknown K_g	Cluster model
k-means / GMM	✗	✗	–	spherical / Gaussian
X-means (Pelleg & Moore, 2000)	✗	✓	–	spherical Gaussian
G-means (Hamerly & Elkan, 2003)	✗	✓	–	Gaussian
DP-GMM (Antoniak, 1974)	✗	✓	–	Gaussian
k-FED (Dennis et al., 2021)	✓	✗	✗	spherical
FedKmeans (Garst & Reinders, 2024)	✓	✗	✗	spherical
FFCM (Stallmann & Wilbik, 2022)	✓	✗	✗	fuzzy / spherical
SPAHM (Yurochkin et al., 2019)	✓	✓	✗	model-agnostic
AFCL (Zhang et al., 2025)	✓	✓	(✓)†	centroid-based
FedGEM (Ibrahim et al., 2026)	✓	✓	✗	isotropic Gaussian
This work	✓	✓	✓	spherical Gaussian‡

the server asynchronously. FedGEM (Ibrahim et al., 2026) casts federated clustering as federated generalized EM: each client fits a local mixture with K_g components together with an uncertainty set for each component, and the server groups components whose sets overlap, estimating K as the number of groups. Because this grouping never uses a cluster count, FedGEM can serve as the second phase of our framework (Section 5). SPAHM (Yurochkin et al., 2019) takes a related approach for models trained independently on separate datasets, matching their parameters under a Bayesian nonparametric prior so that the number of global components is inferred. These methods relax the assumption of a known K but retain other cardinality inputs: FedGEM takes each K_g as given, SPAHM aggregates local models whose sizes were fixed beforehand, and the published experiments of AFCL draw the initial number of seeds from $[K, 2K]$.

Local cardinality. To our knowledge, no federated clustering method estimates both the local counts K_g and the global count K from data without cardinality inputs. Our framework treats both as unknown and supplies the local counts that aggregators such as FedGEM require.

Choosing the number of clusters in centralized clustering. X-means (Pelleg & Moore, 2000) splits a centroid when the split improves the Bayesian Information Criterion, and G-means (Hamerly & Elkan, 2003) splits when a one-dimensional projection of the assigned points fails a normality test, initializing the children along the principal direction. Split-and-merge EM (Ueda et al., 2000) uses split and merge moves to escape local optima at a fixed number of components, whereas Figueiredo & Jain (2002) select the number of components by annihilating weak components under a minimum-message-length criterion. Dirichlet-process mixtures (Antoniak, 1974; Blei & Jordan, 2006) infer the number of components under a nonparametric prior. ASM shares the BIC-driven recursive splitting of X-means and the principal-direction initialization of G-means, but fits each candidate split by EM under a spherical Gaussian model. It also introduces a merge stage that operates on the same components, selection of all hyperparameters on held-out client data alone, and an analysis of when the BIC split criterion fragments a single cluster (Section 4), which applies to X-means-type estimators generally. We compare against DP-GMM, X-means and G-means as first-phase estimators in Section 5.

Clustered federated learning. Clustered federated learning (Ghosh et al., 2020; Sattler et al., 2020) groups clients with similar data distributions in order to train a separate model for each group. It clusters clients rather than data points and is complementary to the problem studied here.

3 Problem setting

A server coordinates G clients. Client $g \in [G]$ holds $\mathcal{X}_g = \{\boldsymbol{x}_{gi}\}_{i=1}^{N_g} \subset \mathbb{R}^d$ and may share model parameters or summary statistics with the server, but never observations. Clients cannot tune hyperparameters jointly, either with one another or with the server: any hyperparameter used on client g

must be selected from $\mathcal{X}_g$ alone. Client-server communication is limited to parameters and summary statistics, and the local phase requires no iteration with the server. The pooled data $\mathcal{X} = \bigcup_g \mathcal{X}_g$ belong to K global clusters indexed by $[K]$. Client g holds points from a subset $S_g \subseteq [K]$ of local cardinality $K_g = |S_g|$, where

$$\bigcup_{g=1}^{G} S_g = [K], \qquad 2 \le K_g < K \ \text{ for all } g, \qquad S_g \cap S_{g'} \neq \emptyset \ \text{ for some } g \neq g'. \tag{1}$$

Clients differ both in which clusters they hold and in the proportions of those clusters, so the local data distributions are non-IID. Neither the server nor the clients know K, $\{K_g\}_{g=1}^G$ or $\{S_g\}_{g=1}^G$. The goal is to partition $\mathcal{X}$ into global clusters and to estimate K without moving data between clients.

The framework has two phases. First, each client independently computes an estimate $\hat{K}_g = \mathcal{A}(\mathcal{X}_g)$ of its local cardinality. Second, an aggregator that requires local cardinalities, such as FedGEM (Ibrahim et al., 2026), takes $\{\hat{K}_g\}$ in place of the true values and, using client-side computations, returns global cluster parameters and $\hat{K}$. We make no generative assumption about the data: ASM uses Gaussian components as a working model, and the analysis of Section 4 is carried out for Gaussian clusters to characterize its behavior, but neither the method nor the experiments require the data to follow a Gaussian mixture. In many of the experiments, clusters are the classes of benchmark datasets. Class labels are used only to construct the client subsets S_g and to evaluate the results, never by the methods.

4 Adaptive split–merge (ASM) algorithm and Analysis

4.1 Algorithm and Analysis

ASM starts from a single component containing all of the client's points and grows the model by recursive binary splitting. To decide whether to split a component, it computes the eigendecomposition of its covariance matrix: the leading eigenvector v_1, with eigenvalue λ_1, gives the direction of greatest spread. Two candidate centroids are formed by perturbing the component mean along this direction in opposite senses, $\mu_\pm = \mu \pm \delta\sqrt{\lambda_1}\, v_1$, where δ is the split factor, and are used to initialize an EM run restricted to the component's points and capped at 15 iterations. The split is accepted if it improves the Bayesian Information Criterion (BIC) over the unsplit component (Appendix B) and each child retains an effective mass, the sum of its responsibilities, of at least τ_{thresh}. The points of an accepted split are reassigned to the child with the highest responsibility, and the children are tested for further splitting in the next pass. Once no split is accepted, components are merged greedily, one pair at a time, whenever their centroids are closer than a multiple of their combined spread,

$$\|\mu_i - \mu_j\|_2 \ \le \ \alpha\,(\sigma_i + \sigma_j), \tag{2}$$

where $\sigma_k = \sqrt{\mathrm{tr}(\Sigma_k)/d}$ is the root-mean-square per-dimension deviation of component k and $\alpha > 0$ is the merge factor. Under the spherical model, a component has standard deviation σ_k along every direction, so equation 2 tests whether two components overlap along the line joining their means, independently of the dimension; the criterion is of the Davies–Bouldin form (Davies & Bouldin, 1979). The pooled mean and spread are recomputed after each merge, and the estimated local count K_g is the number of components that remain. Pseudocode is given in Appendix A.

When does ASM split? Under simplifying assumptions, the split rule admits a closed form that explains ASM's operating envelope. For a single Gaussian cluster of n points cut at its median along the leading eigenvector, the split criterion (equation 4 in Appendix B) is met when:

$$\frac{n}{\log n} \ > \ \frac{d+2}{g(r,d)}, \qquad g(r,d) = -d\log\Big(1 - \frac{2r}{\pi d}\Big) \approx \frac{2r}{\pi}, \qquad r = \frac{\lambda_1}{\mathrm{tr}\,\Sigma/d}. \tag{3}$$

The rule is invariant to feature scale and defines a critical cluster size that falls with the anisotropy r and rises with the dimension d. Three consequences follow. Isotropic clusters are protected up to a size that grows with d, which accounts for near-exact recovery of isotropic clusters in high dimension. Anisotropy is the binding limitation: an elongated cluster reaches its critical size at roughly $1/r$ of the isotropic value and is then split even though it is a single true cluster. And because every cluster has a finite critical size, larger clients split more readily which makes the merge stage in the algorithm essential.

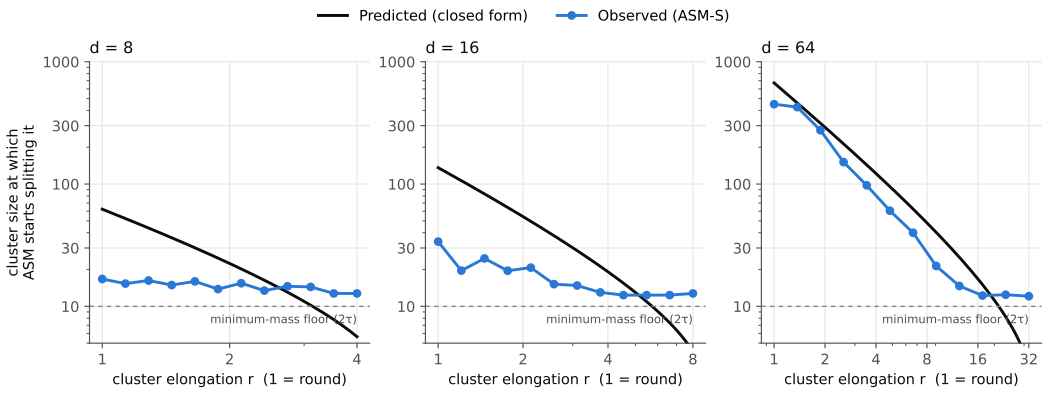

Figure 1: Size at which ASM-S begins to split a single Gaussian cluster, as a function of its anisotropy r ($r = 1$ is round). Black: prediction of equation 3 with the population anisotropy. Blue: size above which at least half of the proposed splits are accepted (50 draws per point, $\delta = 1$, $\tau_{\text{thresh}} = 5$). Below a curve the cluster is kept whole; above it, it is split. Dashed: $2\tau_{\text{thresh}}$, the smallest size the minimum-mass constraint allows to be split.

Validating the split analysis. To test the analysis directly, we apply the split test of ASM-S (ASM with Spherical Gaussians) to a single Gaussian cluster, for which every accepted split is spurious. The cluster has anisotropy r, set independently of the dimension, and for each (d, r) we record the size above which at least half of the proposed splits are accepted (Appendix C). Figure 1 compares this onset with the prediction of equation 3. At $d = 64$ the prediction tracks the observed onset across an order of magnitude in both r and n: a round cluster is kept whole up to about 450 points, whereas one with $r \approx 9$ is split from about 20. At $d = 8$ and 16 the prediction is optimistic: ASM-S splits even round clusters from 15 to 35 points, and the onset sits just above the smallest size the minimum-mass constraint permits. Both effects have a single cause. When n is not much larger than d, the anisotropy estimated from a cluster's own points exceeds its population value, and with r measured on the sample the closed form reproduces the idealized split decision in 94–97% of draws. Once a split is accepted, the recursion continues well below the onset, because halving a cluster inflates the sample anisotropy of its children. In low dimension a single cluster of 500 points yields about 50 components before merging at $d = 8$ and about 20 at $d = 16$, approaching the bound set by the minimum-mass constraint (Figure 2). In this regime the merge stage determines the final count.

4.2 ASM ON POOLED BENCHMARK DATA

The analysis in the previous subsection predicts that ASM is least accurate when a single client holds a large dataset. We therefore evaluate it first in the pooled setting on eight benchmark datasets: four tabular (Waveform, Abalone, FrogA, FrogB) and four image datasets (MNIST, FMNIST, EMNIST, CIFAR-10). Hyperparameters of ASM-S and DP-GMM are selected without labels by the silhouette objective, and results are reported as held-out ARI over ten seeds. As reference points we include X-means, G-means and k-means given the true K^* (Table 2).

Comparison with other estimators. On the tabular datasets, ASM-S is the only method whose estimate is close to K^* throughout ($\hat{K} = 3.0, 7.5, 10.6, 10.8$). X-means and G-means miss by an order of magnitude or reach the cap of 200 components, and DP-GMM over-estimates on the Frog datasets by a factor of ten. On Waveform and Abalone, the ARI of ASM-S equals that of k-means with K^* given (0.256 vs. 0.254 and 0.087 vs. 0.086), so ARI there is limited by the class structure. On the Frog datasets, ASM-S recovers the count but not the partition, reaching about half the ARI of k-means with K^* given. The image datasets behave differently. k-means with K^* given reaches ARI of 0.43–0.65, so the representations separate the classes well, yet ASM-S over-estimates K by a factor of 4 to 15, and DP-GMM achieves higher ARI on three of the four. The rest of this section uses the analysis to explain this gap.

Sample size. The number of components produced by the split stage grows with the number of points per cluster (Figure 2). Reducing each image dataset to a fixed random 25% (Table 3) lowers

Table 2: Pooled (single-client) benchmark: held-out ARI $\pm$ 95% CI half-width (t, $n = 10$ seeds 42–51); second line $\hat{K}$ = mean estimated K. ASM-S and DP-GMM chosen by the label-free silhouette objective; X-/G-Means have none. Oracle: k-means with the true K given (k-means++ initialisation, 10 restarts).The other methods do not use the true K.

Dataset	K^*	ASM-S	DP-GMM	X-Means	G-Means	k-means, K known
Waveform	3	0.256 ± 0.006 $\hat{K}$ 3.0	0.273 ± 0.005 $\hat{K}$ 6.3	0.308 ± 0.016 $\hat{K}$ 6.2	0.208 ± 0.073 $\hat{K}$ 46.2	0.254 ± 0.002 K = 3
Abalone	8	0.087 ± 0.006 $\hat{K}$ 7.5	0.084 ± 0.008 $\hat{K}$ 13.5	0.058 ± 0.003 $\hat{K}$ 28.3	0.023 ± 0.003 $\hat{K}$ 129.1	0.086 ± 0.004 K = 8
FrogA	10	0.273 ± 0.013 $\hat{K}$ 10.6	0.090 ± 0.003 $\hat{K}$ 100.3	0.124 ± 0.005 $\hat{K}$ 77.6	0.070 ± 0.006 $\hat{K}$ 200.0	0.452 ± 0.034 K = 10
FrogB	8	0.153 ± 0.016 $\hat{K}$ 10.8	0.054 ± 0.002 $\hat{K}$ 99.8	0.078 ± 0.004 $\hat{K}$ 76.9	0.046 ± 0.002 $\hat{K}$ 200.0	0.369 ± 0.006 K = 8
MNIST	10	0.162 ± 0.010 $\hat{K}$ 81.0	0.224 ± 0.019 $\hat{K}$ 19.5	0.112 ± 0.006 $\hat{K}$ 2.0	0.078 ± 0.000 $\hat{K}$ 200.0	0.643 ± 0.003 K = 10
FMNIST	10	0.070 ± 0.001 $\hat{K}$ 152.5	0.049 ± 0.027 $\hat{K}$ 2.5	0.060 ± 0.001 $\hat{K}$ 200.0	0.062 ± 0.002 $\hat{K}$ 200.0	0.431 ± 0.002 K = 10
EMNIST	47	0.180 ± 0.003 $\hat{K}$ 193.2	0.202 ± 0.001 $\hat{K}$ 198.1	0.017 ± 0.001 $\hat{K}$ 2.0	0.200 ± 0.001 $\hat{K}$ 200.0	0.326 ± 0.002 K = 47
CIFAR-10	10	0.156 ± 0.009 $\hat{K}$ 87.7	0.259 ± 0.012 $\hat{K}$ 71.3	0.083 ± 0.001 $\hat{K}$ 200.0	0.084 ± 0.002 $\hat{K}$ 200.0	0.647 ± 0.002 K = 10

Table 3: Class anisotropy, onset of merging, and the effect of sample size on ASM-S in the pooled setting. r_{med}: median over classes of $\lambda_1(\Sigma_k)/(\operatorname{tr}\Sigma_k/d)$, computed from the labels for analysis only. $0.8\sqrt{r_{\text{med}}}$: predicted merge factor at which fragments of a typical class are reunited (Appendix B). $\alpha_{1/2}$: interval of grid values between which the estimated count first falls to half its value without merging (Table 6). $\hat{K}$ and ARI: ASM-S on the full data $\rightarrow$ a fixed random 25% of each image dataset, with hyperparameters re-selected by the silhouette objective on the subsample; mean over seeds 42–51. Oracle: k-means with K^* given.

						ASM-S		Oracle
Dataset	d	K^*	r_{med}	$0.8\sqrt{r_{\text{med}}}$	$\alpha_{1/2}$	$\hat{K}$	ARI	ARI
Waveform	21	3	6.39	2.02	1.0–1.25	3.0	0.256	0.254
Abalone	7	8	6.34	2.01	1.5–2.0	7.5	0.087	0.086
FrogA	21	10	9.74	2.50	2.0–3.0	10.6	0.273	0.452
FrogB	21	8	10.11	2.54	2.0–3.0	10.8	0.153	0.369
MNIST	10	10	2.07	1.15	1.0–1.25	$81.0 \rightarrow 26.5$	$0.162 \rightarrow 0.331$	$0.643 \rightarrow 0.638$
FMNIST	64	10	21.12	3.68	> 3.0	$152.5 \rightarrow 38.4$	$0.070 \rightarrow 0.203$	$0.431 \rightarrow 0.429$
EMNIST	16	47	4.07	1.61	1.25–1.5	$193.2 \rightarrow 45.9$	$0.180 \rightarrow 0.222$	$0.326 \rightarrow 0.315$
CIFAR-10	64	10	8.16	2.29	2.0–3.0	$87.7 \rightarrow 17.3$	$0.156 \rightarrow 0.425$	$0.647 \rightarrow 0.643$

the estimate of ASM-S by a factor of 3 to 5 and raises its ARI on all four datasets, while k-means with K^* given is unchanged. The subsample is therefore not an easier clustering problem and the improvement comes from the smaller sample. Because the hyperparameters are re-selected on the subsample, the comparison reflects both the smaller sample and the re-selected configuration.

The merge stage. Table 6 in the Appendix holds the splits fixed and varies the merge factor. Without merging, the split stage over-estimates K on every dataset, by a factor of 1.5 to 15, as expected for pooled sample sizes. The merge rule does not act for $\alpha \leq 0.75$ on any dataset. It removes most of the excess as it starts acting. For instance, on Waveform it reduces 28.8 components to the three classes and triples ARI. The analysis estimates that merging begins for fragments of a class with anisotropy r only when $\alpha \gtrsim 0.8\sqrt{r}$ (Appendix B). Evaluated at the median class anisotropy of each dataset (Table 3), this bound matches, to within one grid step, the value of α at which the estimated count first halves on seven of the eight datasets. The exception is Waveform,

whose merging begins earlier. The bound also accounts for FMNIST. Its median class anisotropy is 21, so merging requires $\alpha \approx 3.7$, beyond every value tested, and the estimate stays at the split-stage count. On MNIST and EMNIST, the bound is met at moderate α, but larger values begin to join distinct classes, and the estimate passes from tens of components to about one between adjacent grid values, so no single α recovers K^*. The silhouette objective also leaves some ARI unused on these datasets (0.200 at $\alpha = 1.25$ against 0.162 selected on MNIST).

Summary. The pooled benchmarks confirm the role of sample size identified by the analysis. The anisotropy bound predicts where merging begins, but not whether it can stop at the right count. On the tabular datasets a single merge factor brings the estimate close to K^*; on the image datasets none does. Merging is absent within the searched range on FMNIST, whose classes are the most anisotropic, and on MNIST and EMNIST the estimate passes from tens of components to about one between adjacent values of α. On these data the single-parameter merge rule may be the main limitation of ASM-S.

5 ASM IN THE FEDERATED SETTING

We evaluate the two-phase pipeline end to end. In Phase 1, each client estimates its local cardinality K_g from its own data; in Phase 2, FedGEM receives the estimates $\{\hat{K}_g\}$ in place of the true values and returns the global cluster structure and $\hat{K}$. We compare ASM-S with three other label-free Phase 1 methods, DP-GMM, X-means and G-means, and with an oracle that supplies the true K_g. Every client selects its hyperparameters independently, by the silhouette of its Phase 1 clustering on its own held-out points, and passes only $\hat{K}_g$ to FedGEM.

Client partitions. For each dataset we use G clients (Table 8), and each client is assigned a random subset of the classes, $0.55K$ on average, subject to equation 1. Points of a class are disjoint across the clients that hold it. We construct two partitions. In partition A, each class is divided equally among the clients that hold it. Since each class is held by about $0.55K$ clients, a client holds on average about $1/(0.55K)$ of a class's points: about 18% for $K = 10$ and 4% for EMNIST, below the 25% subsample of Section 4.2 on every image dataset. Under partition A, however, client size grows almost in proportion to K_g, which could let a Phase 1 method infer K_g from client size rather than from cluster structure. In partition B, each client keeps the same classes, but its size is drawn independently of K_g from $[0.5N_{\text{med}}, 2N_{\text{med}}]$, where N_{med} is the median client size under partition A. On the image datasets, the elasticity of client size with respect to K_g is about one under partition A and about zero under partition B (Appendix G.1); construction details are given in Appendix G.

Results under partition A. Table 4 reports downstream ARI and the error of the global estimate. ASM-S attains the highest mean ARI among the label-free methods on four of the eight datasets (Waveform, FrogA, FrogB and MNIST) and trails the best label-free method by at most 0.024 on the other four. Averaged over the eight datasets, its ARI is 0.333, against 0.256 for X-means, 0.214 for DP-GMM, 0.163 for G-means and 0.361 for the oracle that supplies the true K_g. ASM-S is within 0.03 of the oracle on five datasets and exceeds it on FrogA. Its global estimates are also the most reliable: $|\hat{K} - K^*|$ never exceeds 21 for ASM-S and averages 6.9, compared with 5.4 for the oracle, whereas it reaches 188 for DP-GMM and 263 for G-means. DP-GMM is competitive in ARI on Waveform, Abalone and EMNIST, but its global estimate is off by 44.3 on FrogA, 28.9 on FrogB, 101.1 on FMNIST and 187.8 on CIFAR-10. The image datasets on which ASM-S over-fragmented in the pooled setting behave differently here: on MNIST, FMNIST and CIFAR-10 its ARI rises from 0.162, 0.070 and 0.156 in the pooled setting to 0.418, 0.250 and 0.380, consistent with the smaller per-client cluster sizes.

Results under partition B. Partition B tests whether a Phase 1 method recovers K_g from cluster structure or merely from client size. The test matters most for ASM, whose split criterion depends on sample size equation 3. The estimates of ASM-S are unaffected by the change of partition: pooled over datasets and seeds, the relative error of its global estimate $\hat{K}$ is statistically equivalent across the two partitions within a margin of ± 0.2 (two one-sided tests), and no individual dataset shows a significant difference. X-means and G-means, by contrast, change significantly on three and

Table 4: Full federated pipeline, Regime A, per-client hyperparameter selection. Global ARI ± 95% CI half-width (t, n = 10 seeds 42–51); second line $\hat{K}$ = mean estimated global K, $|\Delta K|$ = mean $\hat{K} - K^*$. Mean row = unweighted mean over the 8 datasets. CIFAR-10 G-Means: 5 timed-out seeds filled from an identical rerun.

Dataset	ASM-S	DP-GMM	X-Means	G-Means	Oracle-K
Waveform	0.258 ± 0.011 $\hat{K}$ 4.5 · $\lvert\Delta K\rvert$ 1.5	0.256 ± 0.027 $\hat{K}$ 4.2 · $\lvert\Delta K\rvert$ 1.2	0.249 ± 0.027 $\hat{K}$ 5.4 · $\lvert\Delta K\rvert$ 2.4	0.111 ± 0.023 $\hat{K}$ 29.9 · $\lvert\Delta K\rvert$ 26.9	0.339 ± 0.024 $\hat{K}$ 2.0 · $\lvert\Delta K\rvert$ 1.0
Abalone	0.089 ± 0.015 $\hat{K}$ 4.9 · $\lvert\Delta K\rvert$ 3.1	0.100 ± 0.005 $\hat{K}$ 8.3 · $\lvert\Delta K\rvert$ 2.9	0.098 ± 0.007 $\hat{K}$ 16.7 · $\lvert\Delta K\rvert$ 8.7	0.080 ± 0.012 $\hat{K}$ 24.7 · $\lvert\Delta K\rvert$ 16.7	0.103 ± 0.007 $\hat{K}$ 6.0 · $\lvert\Delta K\rvert$ 2.0
FrogA	0.634 ± 0.118 $\hat{K}$ 7.9 · $\lvert\Delta K\rvert$ 3.7	0.243 ± 0.088 $\hat{K}$ 54.3 · $\lvert\Delta K\rvert$ 44.3	0.381 ± 0.104 $\hat{K}$ 30.6 · $\lvert\Delta K\rvert$ 20.6	0.174 ± 0.034 $\hat{K}$ 131.2 · $\lvert\Delta K\rvert$ 121.2	0.601 ± 0.104 $\hat{K}$ 7.9 · $\lvert\Delta K\rvert$ 2.1
FrogB	0.401 ± 0.097 $\hat{K}$ 8.8 · $\lvert\Delta K\rvert$ 3.4	0.234 ± 0.076 $\hat{K}$ 36.9 · $\lvert\Delta K\rvert$ 28.9	0.252 ± 0.059 $\hat{K}$ 29.2 · $\lvert\Delta K\rvert$ 21.2	0.070 ± 0.014 $\hat{K}$ 165.6 · $\lvert\Delta K\rvert$ 157.6	0.451 ± 0.093 $\hat{K}$ 5.9 · $\lvert\Delta K\rvert$ 2.1
MNIST	0.418 ± 0.036 $\hat{K}$ 18.5 · $\lvert\Delta K\rvert$ 8.5	0.355 ± 0.041 $\hat{K}$ 8.1 · $\lvert\Delta K\rvert$ 1.9	0.198 ± 0.044 $\hat{K}$ 3.6 · $\lvert\Delta K\rvert$ 6.4	0.287 ± 0.039 $\hat{K}$ 43.8 · $\lvert\Delta K\rvert$ 33.8	0.446 ± 0.048 $\hat{K}$ 9.4 · $\lvert\Delta K\rvert$ 1.0
FMNIST	0.250 ± 0.029 $\hat{K}$ 16.6 · $\lvert\Delta K\rvert$ 7.0	0.178 ± 0.044 $\hat{K}$ 111.1 · $\lvert\Delta K\rvert$ 101.1	0.274 ± 0.026 $\hat{K}$ 21.1 · $\lvert\Delta K\rvert$ 11.1	0.249 ± 0.023 $\hat{K}$ 38.3 · $\lvert\Delta K\rvert$ 28.3	0.252 ± 0.027 $\hat{K}$ 8.0 · $\lvert\Delta K\rvert$ 2.0
EMNIST	0.237 ± 0.006 $\hat{K}$ 48.3 · $\lvert\Delta K\rvert$ 7.3	0.250 ± 0.008 $\hat{K}$ 67.8 · $\lvert\Delta K\rvert$ 20.8	0.193 ± 0.022 $\hat{K}$ 25.7 · $\lvert\Delta K\rvert$ 21.3	0.164 ± 0.009 $\hat{K}$ 309.9 · $\lvert\Delta K\rvert$ 262.9	0.257 ± 0.011 $\hat{K}$ 49.5 · $\lvert\Delta K\rvert$ 2.5
CIFAR-10	0.380 ± 0.029 $\hat{K}$ 30.9 · $\lvert\Delta K\rvert$ 20.9	0.097 ± 0.020 $\hat{K}$ 197.8 · $\lvert\Delta K\rvert$ 187.8	0.404 ± 0.058 $\hat{K}$ 18.2 · $\lvert\Delta K\rvert$ 8.2	0.169 ± 0.016 $\hat{K}$ 97.2 · $\lvert\Delta K\rvert$ 87.2	0.442 ± 0.017 $\hat{K}$ 40.8 · $\lvert\Delta K\rvert$ 30.8

four of the eight datasets, respectively (Appendix G.2). The accuracy of ASM-S under partition A therefore does not rely on the correlation between client size and K_g. The true and estimated local cardinalities of every client, for all methods and both partitions, are shown in Figures 3 and 4.

Computational cost. Table 5 reports the mean Phase 1 wall-clock time per client. ASM-S is the fastest method on six of the eight datasets and has the lowest mean, 339 ms against 558 ms for G-means, 1,110 ms for DP-GMM and 1,943 ms for X-means; G-means is faster on MNIST and FMNIST. Timings reflect the implementations used (Appendix F) and were measured on CPU-only Google Cloud virtual machines running Debian 12, using Python 3.11, scikit-learn 1.6, and NumPy 2.0. No GPUs were used. Runtime measurements were performed on an N2 instance with 32 vCPUs (Intel Xeon), while other experiments used an E2 instance with 12 vCPUs (AMD EPYC 7B12, 12 GB RAM).

Table 5: Mean Phase-1 wall-clock time per client (ms) in the federated benchmark. Bold: fastest.

Dataset	ASM-S	DP-GMM	X-Means	G-Means
Waveform	**266**	309	865	500
Abalone	**193**	529	1,654	490
FrogA	**410**	1,041	3,217	679
FrogB	**314**	1,072	2,883	710
MNIST	96	399	565	**70**
FMNIST	211	528	3,478	**114**
EMNIST	**1,135**	4,839	2,643	1,763
CIFAR-10	**90**	166	238	136
Mean	**339**	1,110	1,943	558

6 Conclusion

We studied federated clustering when neither the local nor the global number of clusters is known, a setting that arises when independent sites, such as the clients of an equipment manufacturer, know no more about their cluster structure than the server does. We proposed a two-phase framework in which each client estimates its local count independently and a federated aggregator reconciles the results, and introduced ASM as a label-free first phase that combines BIC-driven splitting with a merge stage.

Our analysis characterizes when such an estimator works. The split criterion admits a critical cluster size that falls with anisotropy and rises with dimension, and beyond it a single cluster is fragmented until the minimum-mass constraint binds, so the merge stage, rather than the split criterion, determines the final count. On pooled benchmarks, over-fragmentation grows with the number of points per class and shrinks when the sample is reduced, and the onset of merging follows a bound derived from class anisotropy. In the federated setting, where each client holds only part of each class, ASM-S with FedGEM comes within 0.03 ARI of the oracle with true local counts on five of eight datasets, gives the most reliable global estimates of the label-free methods, is robust when client size is decoupled from local cardinality, and is the fastest on average.

Several limitations remain. On the pooled image datasets no single merge factor recovers the true count: merging either does not begin within the searched range or passes from tens of components to about one between adjacent values, and the silhouette objective does not always select the most accurate setting. A merge criterion that accounts for component shape, or that stops at a data-driven count, is a natural next step. In deployment, clusters that dominate a site's data, such as normal operation in the OEM setting, can exceed the critical size at every site, and their fragmentation must then be corrected by the merge stage or the aggregator. Our treatment of disclosure is informal, and we do not provide formal privacy guarantees. Finally, the analysis relies on idealizations (Gaussian clusters, hard assignments and population moments) whose effects we quantify empirically rather than bound.

GENERATIVE AI USAGE STATEMENT

We used a large language model assistant (Claude, Anthropic) in the following ways. *Paper writing:* drafting and editing of text in all sections, including the introduction, related work, problem setup, experimental narrative and conclusion; and suggesting keywords and the choice of primary area. *Literature:* suggesting related work (X-means, G-means, split-and-merge EM, parameter-matching aggregation and clustered federated learning). *Theory:* deriving the closed-form BIC split criterion for the spherical Gaussian model, its reduction to an anisotropy threshold, the merge-factor bound $\alpha \gtrsim 0.8\sqrt{r}$, and the termination argument (Section 4 and Appendix B), and editing the wording of the experimental sections and appendices. *Methodology and implementation:* proposing diagnostic experiments (the oracle baseline, the merge-factor sweep with fixed splits, the subsampling experiment and the class-level anisotropy diagnostic), and writing the code and figures for the single-cluster validation experiment (Section 4 and Appendix C). *Code:* assisting with the implementation of ASM and the experimental pipeline and running and debugging experiment and analysis scripts on top of the implementation of ASM and FedGEM, *Interpretation:* suggesting explanations of the benchmark results, several of which we tested and revised or discarded. The authors checked all derivations, verified every reported number against the experimental outputs, and take full responsibility for the content of the paper.

REPRODUCIBILITY STATEMENT

The ASM algorithm is specified in Section 4, and complete pseudocode in Appendix A. The analysis of our implementation of FedGEM compared to the paper's results is in Appendix E. The hyperparameter selection protocol, including the construction of the held-out client split, number of clients and per-client sample sizes, is described in Appendix F and G; no labels or ground-truth cluster counts are used at any point in selection. The codebase is attached as a zip file in supplementary material.

REFERENCES

Charles E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, pp. 1152–1174, 1974.

David M Blei and Michael I Jordan. Variational inference for dirichlet process mixtures. 2006.

David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.

Don Kurian Dennis, Tian Li, and Virginia Smith. Heterogeneity for the win: One-shot federated clustering. In *International Conference on Machine Learning*, pp. 2611–2620. PMLR, 2021.

Mario A. T. Figueiredo and Anil K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on pattern analysis and machine intelligence*, 24(3):381–396, 2002.

Swier Garst and Marcel Reinders. Federated k-means clustering. In *International Conference on Pattern Recognition*, pp. 107–122. Springer, 2024.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19586–19597, 2020.

Greg Hamerly and Charles Elkan. Learning the k in k-means. *Advances in Neural Information Processing Systems*, 16, 2003.

M. Ibrahim, N. Gebraeel, and W. Xie. A federated generalized expectation-maximization algorithm for mixture models with an unknown number of components. *arXiv preprint arXiv:2601.21160*, 2026. International Conference on Learning Representations (ICLR).

Iain M Johnstone. On the distribution of the largest eigenvalue in principal components analysis. *The Annals of statistics*, 29(2):295–327, 2001.

Fabian Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Dan Pelleg and Andrew W. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 1, pp. 727–734, 2000.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE transactions on neural networks and learning systems*, 32(8):3710–3722, 2020.

Morris Stallmann and Anna Wilbik. Towards federated clustering: A federated fuzzy c-means algorithm (FFCM). *arXiv preprint arXiv:2201.07316*, 2022.

Naonori Ueda, Ryohei Nakano, Zoubin Ghahramani, and Geoffrey E Hinton. Smem algorithm for mixture models. *Neural computation*, 12(9):2109–2128, 2000.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, and Nghia Hoang. Statistical model aggregation via parameter matching. *Advances in neural information processing systems*, 32, 2019.

Y. Zhang, Y. Zhang, Y. Lu, M. Li, X. Chen, and Y.-m. Cheung. Asynchronous federated clustering with unknown number of clusters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 22695–22703, 2025.

SUPPLEMENTARY MATERIAL

A THE ASM ALGORITHM PSEUDOCODE

Algorithm 1 Adaptive Split–Merge (ASM)

Require: Local dataset $\mathcal{D}_g$, split factor δ, minimum mass threshold τ_{thresh}, merge factor α, EM iteration cap $T = 15$

$K_g \leftarrow 1$
Initialize Gaussian using the empirical mean of $\mathcal{D}_g$
repeat
  SplitAccepted ← False
  for all components $k = 1, \ldots, K_g$ **do**
    Fit a one-component model to the points of component k by EM ($\leq T$ iterations)
    Compute the largest eigenvalue λ_1 and corresponding eigenvector v_1 of the responsibility-weighted covariance
    Initialize child means as $\mu_k^{(1)}, \mu_k^{(2)} \leftarrow \mu_k \pm \delta\sqrt{\lambda_1}\, v_1$
    Fit a two-component model to the same points by EM ($\leq T$ iterations), initialized at $\mu_k^{(1)}, \mu_k^{(2)}$
    Compute the BIC improvement Δ_{BIC} of the two-component model over the one-component model
    if $\Delta_{\text{BIC}} > 0$ **and** both children have effective mass $\geq \tau_{\text{thresh}}$ **then**
      Mark k for splitting; each point of k goes to the child it is most responsible for
      SplitAccepted ← True
    end if
  end for
  Replace every marked parent by its two children $\{K_g \leftarrow K_g + \#\text{splits}\}$
until SplitAccepted = False
repeat
  MergeOccurred ← False
  if some pair (i, j) satisfies $\|\mu_i - \mu_j\|_2 \leq \alpha(\sigma_i + \sigma_j)$, where $\sigma_k = \sqrt{\text{tr}(\Sigma_k)/d}$ **then**
    Merge the first such pair: size-weighted mean, pooled points, σ recomputed $\{K_g \leftarrow K_g - 1\}$
    MergeOccurred ← True
  end if
until MergeOccurred = False
return Refined local mixture model

B ANALYSIS OF THE ASM SPLIT CRITERION

B.1 THE CRITERION

ASM accepts a proposed split of component k when it improves a BIC-style score computed on the n_k points of component k alone,

$$\Delta_{\text{BIC}} \;=\; L_2 - L_1 - \frac{d+2}{2}\log n_k, \tag{4}$$

and the split is accepted when $\Delta_{\text{BIC}} > 0$, that is, when

$$2(L_2 - L_1) \;>\; (d+2)\log n_k. \tag{5}$$

Local log-likelihoods. L_M, $M \in \{1, 2\}$, is computed from an M-component spherical Gaussian mixture, $\Sigma_m = s_m^2 I_d$, fitted by EM to the n_k points. The EM objective decomposes into a mixing-weight term and a component term,

$$Q \;=\; \sum_{i=1}^{n_k}\sum_{m=1}^{M}\gamma_{im}\log w_m \;+\; \underbrace{\sum_{i=1}^{n_k}\sum_{m=1}^{M}\gamma_{im}\log\mathcal{N}\big(x_i \mid \mu_m, s_m^2 I_d\big)}_{L_M}, \tag{6}$$

where γ_{im} are the responsibilities from the final E-step and w_m the mixing weights. We use the component term as the local log-likelihood L_M. For $M = 1$, $\gamma_{i1} = 1$ and $\log w_1 = 0$, so L_1 is the ordinary single-Gaussian log-likelihood. For $M = 2$, L_2 omits the mixing-weight term, which at the M-step equals $-n_k H(w)$ with H the entropy of the mixing weights; L_2 is therefore not the observed-data log-likelihood $\sum_i \log \sum_m w_m \mathcal{N}(x_i \mid \mu_m, s_m^2 I_d)$, and relative to the complete-data objective it credits a split with up to $n_k \log 2$.

Penalty. Splitting one component into two adds a d-dimensional mean, a variance and a mixing weight, that is, $d + 2$ free parameters. If the covariance is instead fixed at I_d, the variance is not estimated and the penalty becomes $\frac{d+1}{2} \log n_k$.

B.2 Closed form under hard assignment

Under the spherical model, the maximum-likelihood estimates are $\hat{\mu} = \bar{x}$ and $\hat{s}^2 = \operatorname{tr} \hat{\Sigma}/d$, and at the optimum the quadratic term of the log-likelihood reduces to a constant, giving

$$L_1 = -\frac{n_k d}{2} \log\bigl(2\pi \hat{s}^2\bigr) - \frac{n_k d}{2}. \tag{7}$$

Assigning each point wholly to one of two components of sizes ρn_k and $(1 - \rho) n_k$, with variances $\hat{s}_1^2$ and $\hat{s}_2^2$, each component contributes a term of the same form, so

$$L_2 = -\frac{d}{2} \sum_{m=1}^{2} n_m \log\bigl(2\pi \hat{s}_m^2\bigr) - \frac{n_k d}{2}. \tag{8}$$

The constants cancel. Writing $\tilde{s}^2 = (\hat{s}_1^2)^{\rho} (\hat{s}_2^2)^{1-\rho}$ for the weighted geometric mean of the component variances,

$$L_2 - L_1 = \frac{n_k d}{2} \log \frac{\hat{s}^2}{\tilde{s}^2}, \tag{9}$$

and equation 5 becomes

$$n_k d \log \frac{\hat{s}^2}{\tilde{s}^2} \;>\; (d + 2) \log n_k. \tag{10}$$

Because the variances enter only through their ratio, the criterion is invariant under any rescaling $x \mapsto cx$ of the feature space. This does not hold when the covariance is fixed at I_d: the gain is then $\frac{1}{2} \frac{n_1 n_2}{n_k} \|\hat{\mu}_1 - \hat{\mu}_2\|^2$, which scales as c^2 while the penalty does not. Had the mixing-weight term been retained in L_2, the left-hand side of equation 10 would carry an additional $-2n_k H(\rho)$.

B.3 Reduction to an anisotropy threshold

Consider a single Gaussian cluster with covariance eigenvalues $\lambda_1 \geq \cdots \geq \lambda_d$, cut at its median along the leading eigenvector, as ASM's initialization proposes, so that $\rho = \frac{1}{2}$. On either side of the cut, the coordinate along the split direction is half-normal with variance $\lambda_1(1 - 2/\pi)$, while the orthogonal coordinates, being independent of it, are unchanged. In the large-sample limit, therefore,

$$\hat{s}^2 = \bar{\lambda} \equiv \frac{\operatorname{tr} \Sigma}{d}, \qquad \tilde{s}^2 = \bar{\lambda} - \frac{2}{\pi} \frac{\lambda_1}{d}. \tag{11}$$

Define the anisotropy ratio $r = \lambda_1/\bar{\lambda} \in [1, d]$, equal to one for an isotropic cluster and large for an elongated one. Since $-\log(1 - x) \geq x$, the left-hand side of equation 10 is at least $2n_k r/\pi$, with near equality when $2r/(\pi d) \ll 1$. The split is therefore accepted when

$$\frac{n_k}{\log n_k} \;>\; \frac{\pi(d + 2)}{2r}. \tag{12}$$

B.4 Consequences

Every cluster has a critical size. Equation equation 12 defines, for each anisotropy r, a critical size $n^*(r)$: a single true cluster with fewer points is not split, and one with more is. Since $r \geq 1$, no cluster larger than about $2\tau_{\text{thresh}}$ is protected at every size; an isotropic cluster is split once $n_k/\log n_k$ exceeds $(d + 2)/g(1, d)$.

Anisotropy is the binding limitation. Since $g(r, d) \approx 2r/\pi$, the critical size decreases roughly as $1/r$, so an elongated cluster is fragmented at a fraction of the size at which an isotropic cluster of the same dimension would be.

Dimension protects against splitting. The critical size increases with d: the penalty grows with the number of parameters, while the gain per point from a cut along a single direction, $g(r, d)$, does not grow with d and decreases toward $2r/\pi$. Isotropic clusters in high dimension are therefore split only at large sample sizes.

More data per client makes splitting more permissive. For a fixed cluster shape, the criterion is met once n_k exceeds $n^*(r)$, so ASM is best matched to the federated regime of moderately sized clients and over-splits toward the centralized limit, in which one client holds the entire dataset.

Termination. The minimum-mass constraint guarantees termination: a component is split only if both children retain effective mass at least τ_{thresh}, so no component with fewer than about $2\tau_{\text{thresh}}$ points is split and the local count before merging satisfies $K_g \lesssim n_g/\tau_{\text{thresh}}$. At the population level the criterion would stop the recursion much earlier. A median cut halves the number of points, and since each child's leading eigenvalue is at most λ_1 while its trace falls by $(2/\pi)\lambda_1$, its anisotropy is at most $r/\big(1 - 2r/(\pi d)\big)$; for a round cluster this is only $d/(d - 2/\pi)$. The children would then fail the criterion after a number of levels logarithmic in n_k. In finite samples, however, halving a component inflates the sample anisotropy of its children, and the recursion continues until the minimum-mass constraint binds in low dimension (Appendix C).

Role of the merge stage. Because the split stage protects no cluster larger than about $2\tau_{\text{thresh}}$, the merge stage prevents the fragmentation of large clusters. A median split of a Gaussian with anisotropy r and mean per-dimension variance $\bar{\lambda}$ produces children whose means are $2\sqrt{2/\pi}\,\sqrt{\lambda_1} \approx 1.6\sqrt{r\bar{\lambda}}$ apart, each with per-dimension deviation close to $\sqrt{\bar{\lambda}}$ when $r \ll d$. The merge rule equation 2 therefore reunites them whenever $\alpha \gtrsim 0.8\sqrt{r}$, where r is the anisotropy of the component that was split, as estimated from its points. Spurious splits of round clusters are undone for $\alpha \gtrsim 0.8$ at the population level, whereas those of elongated clusters, or of small fragments whose sample anisotropy is inflated, require a larger α, which in turn risks merging genuinely distinct clusters. Anisotropy thus limits both stages: it makes spurious splits more likely and makes them harder to undo.

B.5 Approximations

The analysis makes three idealizations. First, it assumes hard assignments. At any parameter values, $\sum_m \gamma_{im} \log \mathcal{N}_m \le \max_m \log \mathcal{N}_m$, so the component-term gain obtained by EM does not exceed the hard-assignment gain, and equation 12 is a lower bound on the size and anisotropy at which splits are accepted. The gap can be substantial for a spurious split of a single Gaussian, where responsibilities near the cut are close to $\frac{1}{2}$ and EM draws the two components together; with EM capped at 15 iterations, the realized gain also depends on the initialization. Second, the factor $2/\pi$ is exact only for a median cut of a Gaussian parent: the children of a cut are truncated rather than Gaussian, so the factor is approximate beyond the first level, and a genuinely bimodal cluster separates further and is correctly split more readily. Third, the analysis uses population moments. With n_k not much larger than d, the anisotropy estimated from the component's own points exceeds its population value (Appendix C), so splits are accepted at smaller sizes than equation 12 predicts with the population r.

C Synthetic Validation of the Split Analysis

Setup. Each dataset is a single Gaussian cluster in $\mathbb{R}^d$ with covariance $\mathrm{diag}(\kappa, 1, \ldots, 1)$ and $\kappa = r(d-1)/(d-r)$, which gives population anisotropy exactly r in any dimension and so separates the effects of shape and dimension. We use $d \in \{8, 16, 64\}$, twelve values of r spaced logarithmically from 1 to $d/2$, and eighteen cluster sizes from 6 to 5000. Because the data contain one cluster, any accepted split is spurious. The split test follows Algorithm A: EM initialized at $\mu \pm \delta\sqrt{\lambda_1}\, v_1$, at most 15 iterations, the component-term likelihood of Appendix B, and acceptance when $\Delta_{\text{BIC}} > 0$ and

both children have effective mass at least $\tau_{\mathrm{thresh}} = 5$. For comparison we also evaluate the idealized split of the analysis: a hard cut through the mean orthogonal to v_1, without a mass constraint. Acceptance probabilities use 50 independent draws per configuration, and component counts before merging use 20 draws of the full recursive split stage. The onset in Figure 1 is the size above which at least half of the splits are accepted at every larger size, interpolated logarithmically between grid points.

Sample anisotropy. The closed form equation 3 is stated in terms of the population anisotropy, but the split test sees only the cluster's own points. For an isotropic cluster the leading sample eigenvalue concentrates near $(1 + \sqrt{d/n})^2$ while the mean eigenvalue stays near one (Johnstone, 2001), so the sample anisotropy is approximately $\hat{r} \approx (1 + \sqrt{d/n})^2$, well above one when n is comparable to d. Substituting $\hat{r}$ into the criterion gives a self-consistent onset of about 250 points for a round cluster at $d = 64$, in agreement with the idealized hard split (248 points), compared with 671 from the population value. Across all configurations, the criterion evaluated with the sample anisotropy predicts the outcome of the idealized split in 97%, 94% and 97% of draws at $d = 8$, 16 and 64, against 90%, 85% and 92% with the population anisotropy. The analysis is therefore accurate once r is read as the anisotropy of the points being tested.

Soft assignment and initialization. Soft EM raises the onset relative to the idealized hard split for near-isotropic clusters in high dimension, from 248 to about 450 points for a round cluster at $d = 64$, consistent with the lower-bound argument of Appendix B. For elongated clusters the two coincide. The onset is unchanged for $\delta \in \{0.5, 1, 2\}$ in every configuration, so the split factor is not a sensitive hyperparameter.

The split stage does not stop at the onset. Figure 2 reports the number of components produced by the split stage, before merging, for a single round cluster. Beyond the onset the count grows roughly in proportion to n. At $d = 8$ it follows the bound $n/(2\tau_{\mathrm{thresh}})$ imposed by the minimum-mass constraint, reaching about 470 components for $n = 5000$; at $d = 16$ it reaches about 320, and at $d = 64$ about 16. The recursion continues because the children of a split are not rounder than their parent: each accepted split halves the number of points, which inflates the sample anisotropy of the children, and in a representative run at $d = 8$ the median sample anisotropy rises from 1.06 at the root to about 2.7 eleven levels down. Termination is therefore governed by the minimum-mass constraint in low dimension and by the growth of the BIC penalty with d in high dimension, not by the cluster's shape.

Implications. Two consequences follow for ASM as a whole. First, in low dimension and at the sample sizes of a pooled dataset, a single true cluster reaches the merge stage as tens to hundreds of components, so the accuracy of the final count rests on the merge rule and on the choice of α; from Appendix B, the fragments of a cluster with anisotropy r are reunited only when $\alpha \gtrsim 0.8\sqrt{r}$. Second, the number of components grows with the number of points per cluster, so smaller per-client datasets produce fewer spurious components. This is the sense in which ASM is better suited to the federated regime of moderately sized clients than to the centralized limit.

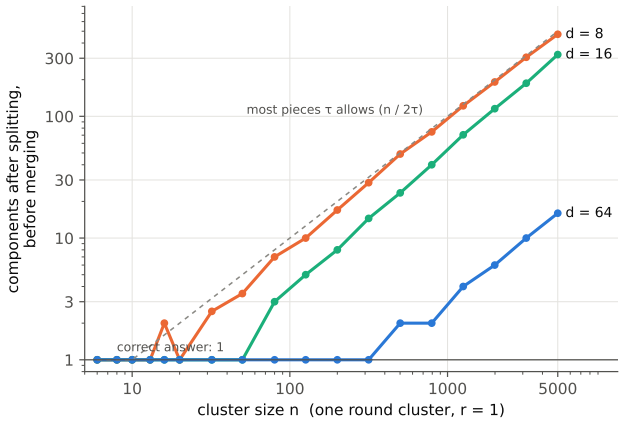

Figure 2: Number of components produced by the split stage of ASM-S, before merging, for a single round Gaussian cluster ($r = 1$) of n points; the correct count is one. Median over 20 draws, $\delta = 1$, $\tau_{\text{thresh}} = 5$. Dashed: the bound $n/(2\tau_{\text{thresh}})$ set by the minimum-mass constraint.

D Analysis of ASM Merge

Table 6: Merge ablation for ASM-S on pooled data over a range of merge thresholds α. Held-out ARI (mean over 10 seeds, 42–51) with the mean number of components $\hat{K}$ below it; "no merge" is the split tree itself. Bold: the α selected by silhouette. $^{\uparrow}/^{\downarrow}$: ARI significantly higher/lower than without the merge (Wilcoxon signed-rank, paired by seed, $p < 0.05$). At $\alpha = 0.25$ the merge never fires on any dataset (identical to no merge). *MNIST's selected $\alpha = 1.1$ is not a column: ARI 0.162, $\hat{K}$ 81.0.

		Merge threshold α						
Dataset	no merge	0.5	0.75	1.0	1.25	1.5	2.0	3.0
Waveform	0.083 $\hat{K}$ 28.8	0.083 $\hat{K}$ 28.8	0.083 $\hat{K}$ 28.8	0.083 $\hat{K}$ 28.8	0.208$^{\uparrow}$ $\hat{K}$ 11.1	0.303$^{\uparrow}$ $\hat{K}$ 6.2	**0.256**$^{\uparrow}$ $\hat{K}$ 3.0	0.050 $\hat{K}$ 1.4
Abalone	0.076 $\hat{K}$ 16.3	0.076 $\hat{K}$ 16.3	0.076 $\hat{K}$ 16.3	0.076 $\hat{K}$ 16.2	0.077 $\hat{K}$ 15.4	0.077 $\hat{K}$ 9.9	**0.087**$^{\uparrow}$ $\hat{K}$ 7.5	0.086$^{\uparrow}$ $\hat{K}$ 3.5
FrogA	0.225 $\hat{K}$ 15.5	0.225 $\hat{K}$ 15.5	0.225 $\hat{K}$ 15.5	0.222 $\hat{K}$ 15.2	0.219$^{\downarrow}$ $\hat{K}$ 14.6	0.215$^{\downarrow}$ $\hat{K}$ 14.0	**0.273**$^{\uparrow}$ $\hat{K}$ 10.6	0.497$^{\uparrow}$ $\hat{K}$ 6.7
FrogB	0.139 $\hat{K}$ 15.6	0.139 $\hat{K}$ 15.6	0.139 $\hat{K}$ 15.6	0.136 $\hat{K}$ 14.9	0.134 $\hat{K}$ 14.7	0.127$^{\downarrow}$ $\hat{K}$ 13.9	**0.153** $\hat{K}$ 10.8	0.275$^{\uparrow}$ $\hat{K}$ 6.2
MNIST*	0.090 $\hat{K}$ 150.9	0.090 $\hat{K}$ 150.9	0.090 $\hat{K}$ 150.9	0.112$^{\uparrow}$ $\hat{K}$ 121.3	0.200$^{\uparrow}$ $\hat{K}$ 27.2	0.136$^{\uparrow}$ $\hat{K}$ 16.0	0.002$^{\downarrow}$ $\hat{K}$ 1.2	0.000$^{\downarrow}$ $\hat{K}$ 1.0
FMNIST	0.070 $\hat{K}$ 152.5	0.070 $\hat{K}$ 152.5	0.070 $\hat{K}$ 152.5	0.070 $\hat{K}$ 152.5	0.070 $\hat{K}$ 152.5	0.070 $\hat{K}$ 152.5	0.070 $\hat{K}$ 152.5	0.071$^{\uparrow}$ $\hat{K}$ 149.9
EMNIST	0.180 $\hat{K}$ 193.2	0.180 $\hat{K}$ 193.2	0.180 $\hat{K}$ 193.2	0.180 $\hat{K}$ 193.1	0.189$^{\uparrow}$ $\hat{K}$ 176.7	0.230$^{\uparrow}$ $\hat{K}$ 76.6	0.057$^{\downarrow}$ $\hat{K}$ 21.6	0.000$^{\downarrow}$ $\hat{K}$ 1.0
CIFAR-10	0.120 $\hat{K}$ 112.5	0.120 $\hat{K}$ 112.5	0.120 $\hat{K}$ 112.5	0.120 $\hat{K}$ 112.5	0.121 $\hat{K}$ 112.1	0.122$^{\uparrow}$ $\hat{K}$ 110.8	**0.156**$^{\uparrow}$ $\hat{K}$ 87.7	0.164$^{\uparrow}$ $\hat{K}$ 30.9

E Our FedGEM implementation and the released code

Why a separate implementation. ASM is evaluated inside FedGEM, so every result depends on the aggregator being implemented as specified. Running the released code unmodified, we found three steps that we implement differently: two in which the code departs from the algorithm

Table 7: FedGEM with the true local cardinalities: personalised ARI ± 95% CI (t-distribution) and, below it, the mean global $\hat{K}$. *Published*: Table 1 of Ibrahim et al. (2026), CI from the reported SD with $n = 50$. *Released code*: the authors' unmodified FedGEM code run by us (8–10 runs; FMNIST with the script's $T = 20$; not run on CIFAR-10). *Ours*: our implementation, ten repetitions (seeds 42–51); last column its global ARI. Protocols also differ in partition, number of repetitions and feature standardisation.

		personalised ARI ($\hat{K}$)			
Dataset	K^*	published	released code	ours	ours, global ARI
Waveform	3	0.335 ± 0.022 $\hat{K}$ 4.4	0.318 ± 0.078 $\hat{K}$ 3.7	0.405 ± 0.063 $\hat{K}$ 2.0	0.339 ± 0.024
Abalone	8	0.138 ± 0.016 $\hat{K}$ 12.1	0.131 ± 0.032 $\hat{K}$ 15.2	0.174 ± 0.036 $\hat{K}$ 6.0	0.103 ± 0.007
FrogA	10	0.552 ± 0.037 $\hat{K}$ 23.9	0.626 ± 0.063 $\hat{K}$ 31.4	0.489 ± 0.055 $\hat{K}$ 7.9	0.601 ± 0.104
FrogB	8	0.468 ± 0.033 $\hat{K}$ 20.0	0.481 ± 0.128 $\hat{K}$ 22.9	0.318 ± 0.089 $\hat{K}$ 5.9	0.451 ± 0.093
MNIST	10	0.452 ± 0.014 $\hat{K}$ 13.6	0.481 ± 0.033 $\hat{K}$ 13.6	0.490 ± 0.014 $\hat{K}$ 9.4	0.446 ± 0.048
FMNIST	10	0.287 ± 0.016 $\hat{K}$ 17.6	0.195 ± 0.031 $\hat{K}$ 90.0	0.362 ± 0.009 $\hat{K}$ 8.0	0.252 ± 0.027
EMNIST	47	0.285 ± 0.006 $\hat{K}$ 58.7	0.322 ± 0.026 $\hat{K}$ 57.7	0.332 ± 0.004 $\hat{K}$ 49.5	0.257 ± 0.011
CIFAR-10	10	0.286 ± 0.009 $\hat{K}$ 37.7	–	0.546 ± 0.019 $\hat{K}$ 40.8	0.442 ± 0.017

of Ibrahim et al. (2026), and one in which paper and code agree but, as far as we can tell, contain a sign error. (i) The per-round uncertainty radius is meant to be found by bisection. In the released code the condition of the bisection loop is never satisfied on entry, so the radius always keeps its initial value $\|\hat{M} - \theta\|^2/2$. Our implementation runs the bisection. Run unmodified on MNIST, the released code reproduces the published global cardinality almost exactly ($\hat{K} = 13.6$ against the published 13.6), which indicates that the published numbers were obtained with this unrefined radius. (ii) When the uncertainty balls $B(\theta_1, r_1)$ and $B(\theta_2, r_2)$ of two clients overlap, the server moves both components to a point ν intended to lie in their intersection. Algorithm 2 of the paper and the released code step from θ_1 in the direction $\theta_1 - \theta_2$, away from the second ball; we step in the direction $\theta_2 - \theta_1$, toward it. Over 20,000 random overlapping pairs in 2 to 64 dimensions, our ν lies in both balls in every case and the published formula in 5.9% of cases, only when the second ball already contains θ_1 with room to spare. (iii) The released code stops early when the means change by less than 10^{-3}; we run a fixed $T = 10$ rounds (the released FMNIST script uses $T = 20$). As in the released code, components of the same client are never grouped together.

Metric and results. The released code reports the mean over clients of each client's ARI under its own components (personalised ARI). Our primary metric is the ARI of the global model on the pooled test set, and we report both. Table 7 compares, on the personalised metric, the published results, the released code as run by us (8–10 runs on seven datasets) and our implementation (Oracle-K_g, ten repetitions on all eight). The implementations differ in accuracy in both directions: ours is higher on Waveform, Abalone, FMNIST and CIFAR-10 and lower on FrogA and FrogB. Ours, however, estimates the global cardinality much more closely. Its $|\hat{K} - K^\star|$ is smaller than that of the published results on seven of the eight datasets, for example FrogA 7.9 against 23.9 with $K^\star = 10$ and EMNIST 49.5 against 58.7 with $K^\star = 47$. The exception is CIFAR-10, where both overshoot (40.8 against 37.7). It is also smaller than that of the released code on six of the seven datasets on which we ran it; the exception is Waveform. On FMNIST the released script, run with its own $T = 20$, reaches $\hat{K} = 90$; with $T = 10$ as in ours it gives 9.1. Because Oracle-K_g run through the same implementation, the comparisons of Section 5 do not depend on these differences.

Table 8: Datasets and federation settings. $\overline{K_g}$: mean number of classes per client.

Dataset	N	d	$K^\star$	G	train/test	$\overline{K_g}$	features
Waveform	5,000	21	3	5	70/30	2.0	raw, standardised
Abalone	4,177	7	8	5	70/30	4.5	raw (sex removed), standardised
FrogA	7,195	21	10	5	70/30	6.3	raw, standardised
FrogB	7,195	21	8	5	70/30	4.3	raw, standardised
MNIST	70,000	10	10	100	70/30	6.1	VAE embedding
FMNIST	70,000	64	10	100	70/30	6.1	VAE embedding
EMNIST	131,600	16	47	25	50/50	26.1	VAE embedding

F Implementation and protocol details

Data. The tabular features are standardised; as in the released loader of Ibrahim et al. (2026), the categorical sex attribute of Abalone is removed, and its age bins yield $K^\star = 8$. The image datasets are used as embeddings without standardisation:MNIST ($d = 10$) uses the released VAE embeddings; FMNIST ($d = 64$) and EMNIST ($d = 16$) use VAE embeddings that we trained with the architecture specified by Ibrahim et al. (2026); CIFAR-10 ($d = 64$) uses the released Barlow Twins embeddings passed through a second-stage VAE that we retrained, because the original was not released. FrogA and FrogB share one feature matrix and differ only in their labels. Every evaluation seed (42–51) draws a stratified train/test split: 70/30, and 50/50 for EMNIST. Every class is held by at least two clients.

Local EM and comparison methods.

- *ASM-I* (identity covariance) uses a dedicated EM implementation with fixed $\Sigma = \boldsymbol{I}$ and estimated weights.
- *ASM-S* (spherical covariance) uses scikit-learn's `GaussianMixture` (Pedregosa et al., 2011), with the perturbed means as initial means.
- *Merge rule.* All ASM runs, under both ASM-I and ASM-S, use the linear merge rule: two components are merged when $\|\mu_i - \mu_j\| \le \alpha(\sigma_i + \sigma_j)$, with $\sigma = \sqrt{\operatorname{tr}\Sigma/d}$.
- *DP-GMM:* scikit-learn `BayesianGaussianMixture` with spherical covariance, truncation level 200 and at most 200 iterations. Components below the weight threshold are removed, and their points are reassigned to the nearest remaining mean.
- *X-Means and G-Means:* the `pyclustering` implementations with at most 200 clusters.

Every Phase 1 method therefore has the same capacity of 200 components.

Hyperparameter grids.

Setting	Grid
ASM (ASM-I / ASM-S), real data (federated: selected per client; pooled: one configuration; 120 configurations)	$A \in \{20, 50, 80, 120, 200\}$, $\delta \in \{0.3, 0.5, 0.7, 1.0\}$, $\alpha \in \{0.25, 0.5, 0.75, 1.0, 1.5, 2.0\}$
ASM (ASM-I / ASM-S), MNIST (140 configurations)	as above, with $\alpha \in \{0.25, 0.5, 0.75, 0.9, 1.0, 1.1, 1.2\}$
DP-GMM (truncation level 200; 24 configurations)	concentration $\in \{0.001, 0.01, 0.1, 1, 10, 100\}$, weight threshold $\in \{0.001, 0.01, 0.03, 0.1\}$
υ_g, tabular data	Waveform $\{1, 5, 10\}$; Abalone $\{5000, 7000, 9000\}$; FrogA, FrogB $\{10^4, 10^5, 10^6\}$
υ_g, image embeddings	fixed: MNIST 2, FMNIST 300, EMNIST 2, CIFAR-10 10

With per-client selection, each client chooses its own configuration from the grid by the silhouette of its Phase 1 clustering on its own held-out points.

Time limits. Each FedGEM run used to select υ_g was limited to 150 s, and values whose runs timed out on both tuning seeds were excluded from selection. Evaluation runs were limited to 3,600 s. G-Means exceeded this limit on 5 of 10 seeds for CIFAR-10 on Regime A's partition; these were filled from an identical rerun. It also exceeded the limit on 1 seed for EMNIST on Regime B's partition, which is reported over the remaining 9. No other evaluation run timed out.

CIFAR-10 υ_g. The value of 10 differs from the paper's 20. The paper did not release itssecond-stage VAE, so we retrained it, and the retrained features have a different scale (a standard deviation of about 2.2 per dimension, against about 1 for the other embeddings). On these features the silhouette

differences across the υ_g grid were within noise. We therefore fixed υ_g at the value at which our implementation's global cardinality is closest to the published one.

Selection score. Each client's training data are split at random into fitting (80%) and validation (20%) sets. Every client fits Phase 1 with each configuration on its fitting set and keeps the configuration with the largest silhouette of its own validation points under its own components (ties:smaller $\hat{K}_g$, then grid order). No information is pooled across clients.

Pooled regime. With a single client holding all data, configurations are selected on a random subsample of at most 30,000 points split 80/20 (tuning seeds 1000, 1001). The score is the silhouette of the validation points, each assigned to the nearest component mean. Every method is then fitted on the training part of each evaluation split and scored by the ARI of the held-out test points assigned to the nearest component mean.

G Regime A and B

Figures 3 and 4 show what Phase 1 hands to Phase 2 on every client. Each row is one dataset and each column one method. The grey histogram gives the true local cardinalities K_g of all clients over the ten repetitions, and the blue histogram gives the estimates $\hat{K}_g$ of the same clients, the method is accurate when the two histograms overlap.

Train/test split and clients. For every evaluation seed, each dataset is first split into a training and a held-out test set, stratified by class. Only the training set is distributed to clients. The number of clients G is fixed per dataset: $G = 5$ for Waveform, Abalone, FrogA and FrogB, $G = 25$ for EMNIST and $G = 100$ for MNIST, FMNIST and CIFAR-10. A client's true local cardinality K_g is the number of distinct clusters present among the data points it receives.

Regime A.

1. *Class sets.* Each client draws a number of classes uniformly from $\{\lfloor 0.55K \rceil - j, \ldots, \lfloor 0.55K \rceil + j\}$ with $j = \max(1, \lfloor 0.25K \rceil)$, clipped to $[2, K-1]$, and then that many classes uniformly at random without replacement. Every client therefore holds a strict subset of the classes (for Waveform, with $K = 3$, every client holds exactly two).
2. *Coverage.* Any class held by fewer than two clients is added to randomly chosen clients that do not yet hold it and hold fewer than $K - 1$ classes, until every class has at least two owners.
3. *Allocation.* The training points of each class are shuffled and divided into equal parts among the clients that hold the class. Clients are disjoint and every training point is assigned.

A draw is accepted if every client holds between 2 and $K - 1$ classes and at least 100 points; otherwise it is redrawn (at most 200 times, keeping the draw with the largest minimum client size). Because a client receives an equal share of each of its classes, its size grows with its number of classes: the Spearman correlation between $N_g$and K_g is 0.98–0.99 on the image datasets and 0.85 on Abalone, and the number of points per class, N_g/K_g, varies little across clients (coefficient of variation 0.04 on the image datasets). The correlation is weaker on FrogA (0.52) and FrogB (0.36), whose classes differ strongly in size, and undefined on Waveform, where every $K_g = 2$. This partition is representative of practice, but a method that counts points could appear to count clusters.

Regime B. Regime B's partition is derived from the Regime A's partition of the same seed and removes the link between size and cardinality while keeping each client's class set, and hence its true K_g.

1. *Target sizes.* Let N_{med} be the median client size of Regime A. Each client draws a target size $T_g = N_{\text{med}}\, e^{u_g}$ with $u_g \sim \mathcal{U}[\log 0.5, \log 2]$, independently of K_g, i.e. log-uniformly over the fourfold range $[0.5N_{\text{med}}, 2N_{\text{med}}]$.
2. *Allocation.* Each client requests T_g/K_g points from each of its classes. When the requests for a class exceed its training points, all requests for that class are scaled down by the same



Figure 3: Local cardinalities under Regime A: true K_g (grey) and estimated $\hat{K}_g$ (blue) for every client over ten repetitions. Rows are datasets; columns are, from left to right, ASM-I, ASM-S, DP-GMM, X-Means and G-Means. Panel titles give the mean absolute error of $\hat{K}_g$ and the share of clients with $\hat{K}_g = 1$.



Figure 4: Local cardinalities under the Regime B, laid out as in Figure 3. Panel titles give the mean absolute error of $\hat{K}_g$ and the Spearman correlation ρ between $\hat{K}_g$ and K_g, with the value under the Regime A in brackets.

factor. Each client then receives $\max(2, \mathrm{round}(\cdot))$ points of each of its classes, drawn without replacement from the shuffled class pool, so clients remain disjoint; unlike Regime A, not every training point need be used.

3. *Fallback.* A client that would receive fewer than 60 points keeps its Regime A data.

Regime B's partition removes the size shortcut: on the image datasets the correlation between N_g and K_g is statistically equivalent to zero (-0.01 to 0.04), the coefficient of variation of N_g/K_g rises from 0.04 to 0.48–0.50, and the elasticity b in $\log N_g = a + b \log K_g$ falls from 1.00–1.03 to -0.03–0.05. A side effect is that clients with more classes now hold fewer points per class.

Method settings on the two partitions. On both partitions each client selects its own Phase 1 configuration on its own data (Appendix F); only the FedGEM setting υ_g is carried over from Regime A's tuning. Oracle-K_g is given the true class count of each client. All other protocol details (evaluation seeds 42–51, held-out test set, statistics) are as in Appendix F.

Table 9: Full federated pipeline, Regime B, per-client hyperparameter selection. Global ARI ± 95% CI half-width (t, n = 10 seeds 42–51) on the Regime B; second line $\hat{K}$, $|\Delta K|$ = mean —$\hat{K} - K^*$—. Mean row = unweighted mean over 8 datasets.

Dataset	ASM-S	DP-GMM	X-Means	G-Means	Oracle-K
Waveform	0.280 ± 0.022 $\hat{K}$ 4.4 · $\lvert\Delta K\rvert$ 1.4	0.252 ± 0.027 $\hat{K}$ 4.1 · $\lvert\Delta K\rvert$ 1.1	0.271 ± 0.032 $\hat{K}$ 5.2 · $\lvert\Delta K\rvert$ 2.2	0.138 ± 0.035 $\hat{K}$ 21.3 · $\lvert\Delta K\rvert$ 18.3	0.340 ± 0.011 $\hat{K}$ 2.0 · $\lvert\Delta K\rvert$ 1.0
Abalone	0.092 ± 0.010 $\hat{K}$ 3.8 · $\lvert\Delta K\rvert$ 4.2	0.097 ± 0.014 $\hat{K}$ 6.5 · $\lvert\Delta K\rvert$ 1.7	0.100 ± 0.008 $\hat{K}$ 14.4 · $\lvert\Delta K\rvert$ 6.4	0.071 ± 0.012 $\hat{K}$ 28.4 · $\lvert\Delta K\rvert$ 20.4	0.101 ± 0.007 $\hat{K}$ 6.0 · $\lvert\Delta K\rvert$ 2.0
FrogA	0.651 ± 0.123 $\hat{K}$ 9.1 · $\lvert\Delta K\rvert$ 2.3	0.384 ± 0.169 $\hat{K}$ 53.2 · $\lvert\Delta K\rvert$ 43.2	0.494 ± 0.169 $\hat{K}$ 24.9 · $\lvert\Delta K\rvert$ 14.9	0.304 ± 0.116 $\hat{K}$ 99.8 · $\lvert\Delta K\rvert$ 89.8	0.588 ± 0.138 $\hat{K}$ 7.9 · $\lvert\Delta K\rvert$ 2.1
FrogB	0.442 ± 0.083 $\hat{K}$ 10.4 · $\lvert\Delta K\rvert$ 3.0	0.255 ± 0.055 $\hat{K}$ 33.2 · $\lvert\Delta K\rvert$ 25.2	0.329 ± 0.094 $\hat{K}$ 20.9 · $\lvert\Delta K\rvert$ 12.9	0.190 ± 0.041 $\hat{K}$ 89.5 · $\lvert\Delta K\rvert$ 81.5	0.385 ± 0.096 $\hat{K}$ 5.9 · $\lvert\Delta K\rvert$ 2.1
MNIST	0.420 ± 0.032 $\hat{K}$ 14.7 · $\lvert\Delta K\rvert$ 4.7	0.408 ± 0.025 $\hat{K}$ 11.1 · $\lvert\Delta K\rvert$ 1.9	0.248 ± 0.051 $\hat{K}$ 4.6 · $\lvert\Delta K\rvert$ 5.4	0.273 ± 0.048 $\hat{K}$ 53.6 · $\lvert\Delta K\rvert$ 43.6	0.418 ± 0.057 $\hat{K}$ 8.7 · $\lvert\Delta K\rvert$ 1.3
FMNIST	0.274 ± 0.022 $\hat{K}$ 14.2 · $\lvert\Delta K\rvert$ 4.4	0.172 ± 0.027 $\hat{K}$ 114.3 · $\lvert\Delta K\rvert$ 104.3	0.283 ± 0.021 $\hat{K}$ 25.6 · $\lvert\Delta K\rvert$ 15.6	0.247 ± 0.024 $\hat{K}$ 46.6 · $\lvert\Delta K\rvert$ 36.6	0.250 ± 0.025 $\hat{K}$ 8.0 · $\lvert\Delta K\rvert$ 2.0
EMNIST	0.240 ± 0.011 $\hat{K}$ 44.7 · $\lvert\Delta K\rvert$ 7.1	0.258 ± 0.011 $\hat{K}$ 71.2 · $\lvert\Delta K\rvert$ 24.2	0.264 ± 0.021 $\hat{K}$ 47.0 · $\lvert\Delta K\rvert$ 7.8	0.149 ± 0.008 $\hat{K}$ 364.8 · $\lvert\Delta K\rvert$ 317.8 · n = 9	0.242 ± 0.007 $\hat{K}$ 46.4 · $\lvert\Delta K\rvert$ 1.2
CIFAR-10	0.379 ± 0.037 $\hat{K}$ 31.7 · $\lvert\Delta K\rvert$ 21.7	0.096 ± 0.008 $\hat{K}$ 199.4 · $\lvert\Delta K\rvert$ 189.4	0.403 ± 0.040 $\hat{K}$ 21.8 · $\lvert\Delta K\rvert$ 11.8	0.155 ± 0.013 $\hat{K}$ 111.3 · $\lvert\Delta K\rvert$ 101.3	0.415 ± 0.040 $\hat{K}$ 43.6 · $\lvert\Delta K\rvert$ 33.6

G.1 Points per cluster in Regime A and B

Regime B's partition is meant to remove the link between a client's size and its number of classes. We check this directly with a simple quantity, the number of points a client holds per class,

$$\text{points per cluster} \;=\; N_g/K_g.$$

If client size is proportional to cardinality, as the construction of Regime A's partition implies, N_g/K_g is nearly the same for every client; if size is unrelated to cardinality, it varies widely. We compute it for every client of every evaluation seed (42–51) on both partitions, using the partitions of the main experiments, and summarise its spread by the standard deviation (SD) and the coefficient of variation, $\mathrm{CV} = \mathrm{SD}/\mathrm{mean}$, which allows datasets with different class sizes to be compared.

Descriptive results. Table 10 and Figure 5 report the distribution over all clients and seeds. On the four image datasets Regime A's partition gives almost every client the same number of points per class: the CV is 0.04 (SD 2–4 points around means of 69–101), and the Spearman correlation between N_g and K_g is 0.98–0.99. In Regime B's the mean is essentially unchanged, but the spread grows about twelve fold (CV 0.48–0.50, SD 34–51), and the correlation between N_g and K_g vanishes (-0.01 to 0.04). The tabular datasets behave in the same direction but start from a larger spread in Regime A (CV 0.15–0.68), for the reason discussed below.

Statistical tests. Clients of the same seed share one partition and are therefore not independent, so every test below uses the seed as the unit of replication (ten seeds per dataset). Regime B's partition of a seed is built from Regime A's partition of the same seed, so the two partitions are paired by seed. We test two properties (Table 11).

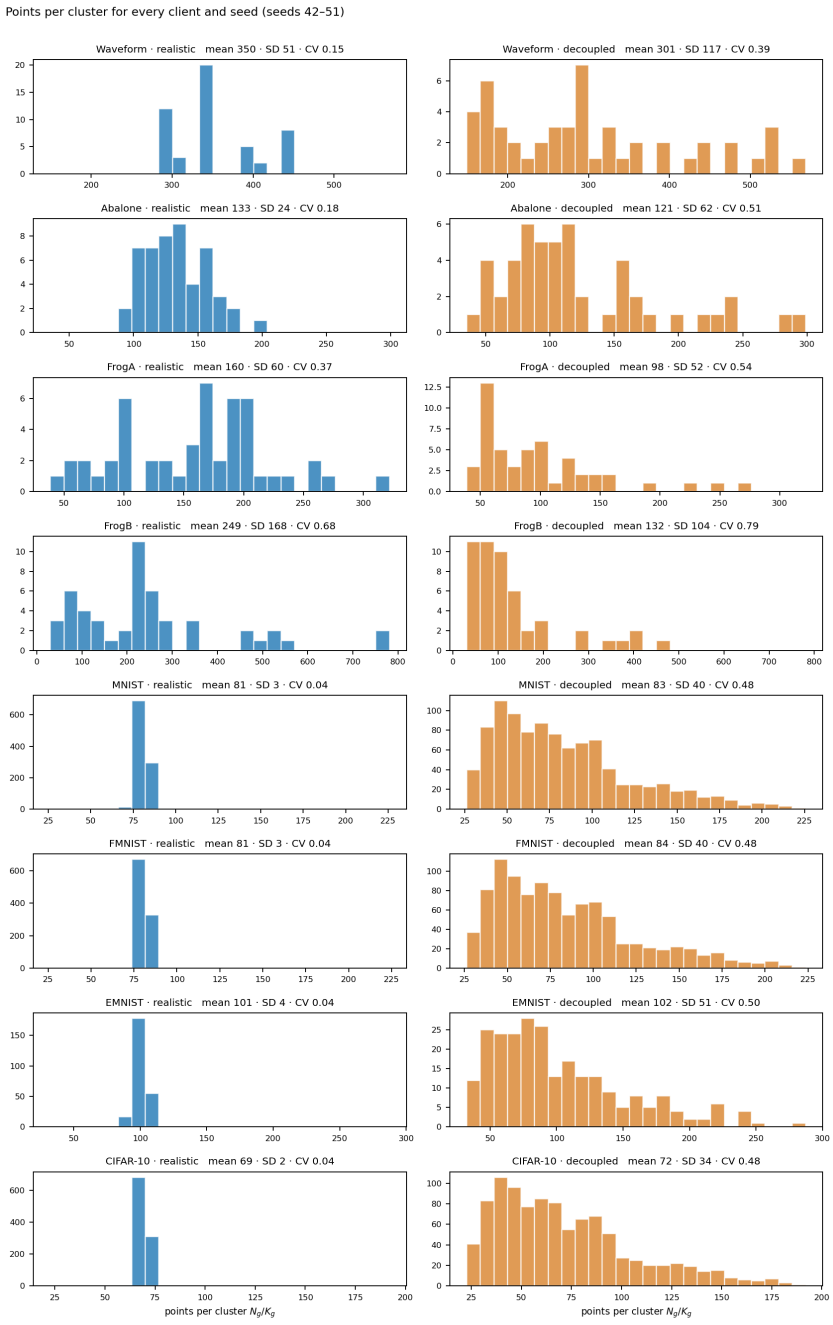

Figure 5: Points per cluster N_g/K_g for every client and seed (seeds 42–51), Regime A (left) and Regime B (right), with the mean, SD and CV of each distribution.

1. *Dispersion.* For each seed and partition we compute the CV of N_g/K_g over that seed's clients and test whether Regime B's CV exceeds the Regime A's partition one with a one-sided Wilcoxon signed-rank test over the ten seeds, Holm-corrected across the eight datasets. We also report the ratio of the two CVs over all clients, with a 95% confidence interval obtained by resampling whole seeds (10,000 draws).
2. *Proportionality.* We fit the elasticity b in $\log N_g = a + b \log K_g$ over all clients of a partition, with a 95% confidence interval from the same seed bootstrap. Client size proportional to cardinality corresponds to $b = 1$; size unrelated to cardinality corresponds to $b = 0$.

Table 10: Points per cluster N_g/K_g over every client and seed (seeds 42–51) of the realistic and decoupled partitions: mean, standard deviation, coefficient of variation (SD/mean) and Spearman correlation between client size N_g and cardinality K_g ("–": every Waveform client has $K_g = 2$). A near-constant N_g/K_g (low CV) means client size is proportional to cardinality.

		realistic				decoupled			
Dataset	cases	mean	SD	CV	$\rho(N_g, K_g)$	mean	SD	CV	$\rho(N_g, K_g)$
Waveform	50	350	51	0.15	–	301	117	0.39	–
Abalone	50	133	24	0.18	+0.85	121	62	0.51	+0.14
FrogA	50	160	60	0.37	+0.52	98	52	0.54	+0.08
FrogB	50	249	168	0.68	+0.36	132	104	0.79	-0.08
MNIST	1000	81	3	0.04	+0.98	83	40	0.48	-0.01
FMNIST	1000	81	3	0.04	+0.98	84	40	0.48	-0.01
EMNIST	250	101	4	0.04	+0.99	102	51	0.50	+0.04
CIFAR-10	1000	69	2	0.04	+0.98	72	34	0.48	-0.01

Table 11: Tests of the points-per-cluster comparison, with the seed as the unit of replication (10 seeds; the two partitions are paired by seed). CV: coefficient of variation of N_g/K_g within a seed (median over seeds). p: one-sided Wilcoxon signed-rank test of $\text{CV}_{\text{decoupled}} > \text{CV}_{\text{realistic}}$ over seeds, Holm-corrected across datasets. CV ratio: decoupled / realistic over all clients, 95% CI from resampling whole seeds. b: elasticity in $\log N_g = a + b \log K_g$ with a 95% seed-bootstrap CI; $b = 1$ means client size is proportional to cardinality, $b = 0$ that it is unrelated ("–": every Waveform client has $K_g = 2$).

Dataset	CV realistic	CV decoupled	seeds higher	p	CV ratio [95% CI]	b realistic [95% CI]	b decoupled [95% CI]
Waveform	0.11	0.42	10/10	0.008	2.66 [1.94, 4.08]	–	–
Abalone	0.09	0.46	10/10	0.008	2.85 [2.34, 3.74]	0.85 [0.68, 1.02]	0.11 [-0.06, 0.32]
FrogA	0.39	0.48	6/10	0.275	1.43 [1.04, 1.97]	1.33 [0.75, 1.89]	0.16 [-0.33, 0.59]
FrogB	0.63	0.58	5/10	0.500	1.17 [0.99, 1.53]	0.74 [0.12, 1.34]	-0.16 [-0.56, 0.30]
MNIST	0.03	0.48	10/10	0.008	11.96 [10.01, 15.74]	1.01 [0.99, 1.02]	-0.03 [-0.15, 0.09]
FMNIST	0.02	0.48	10/10	0.008	13.43 [10.95, 18.98]	1.00 [0.99, 1.02]	-0.03 [-0.15, 0.09]
EMNIST	0.03	0.50	10/10	0.008	11.17 [9.41, 14.80]	1.03 [1.01, 1.04]	0.05 [-0.03, 0.12]
CIFAR-10	0.02	0.48	10/10	0.008	13.42 [10.94, 18.80]	1.00 [0.99, 1.02]	-0.03 [-0.15, 0.09]

Results of the tests. On six of the eight datasets the dispersion of points per cluster is higher in Regime B' partition in every one of the ten seeds ($p = 0.008$ after Holm correction, the smallest value attainable with ten paired seeds). The CV ratio is 11–13 on the image datasets (95% intervals from 9.4 to 19.0) and about 3 on Waveform and Abalone (intervals 1.9–4.1 and 2.3–3.7). The elasticities confirm the mechanism: on the image datasets b is 1.00–1.03 in Regime A's partition, so client size is proportional to cardinality, and -0.03 to 0.05 in Regime B's, with every interval containing zero, so size carries no information about cardinality. Abalone follows the same pattern ($b = 0.85$, interval $[0.68, 1.02]$, against 0.11, interval $[-0.06, 0.32]$); on Waveform every client holds two classes, so b is undefined.

FrogA and FrogB are the exceptions: the increase in CV is not significant ($p = 0.28$ and 0.50). Their classes differ in size by up to a factor of sixty (48 to 2,905 training points), so the number of points a client holds per class already depends strongly on which classes it holds, and Regime A's partition is only weakly coupled to begin with (CV 0.39 and 0.63, correlations 0.52 and 0.36). Regime B's partition still removes the remaining link: Regime B's elasticities are 0.16 and -0.16, with intervals containing zero.

G.2 Robustness of cardinality prediction to the partition

In Regime A's partition, a client's size grows with the number of classes it holds (Appendix G.1), so a Phase 1 method could estimate K_g from N_g instead of from the structure of the data. Regime B's partition removes this link while keeping every client's class set. If a method's estimate of the number of clusters does not depend on client size, its accuracy should be the same on both partitions. If it does depend on client size, its accuracy should change. We test this directly on the quantity the pipeline is meant to deliver, the final number of clusters $\hat{K}$, and on the local estimates $\hat{K}_g$ it is built from.

Table 12: Robustness of cardinality prediction to the partition (realistic → decoupled; the two partitions are paired by seed and client, seeds 42–51). *Global*: relative error of the final count, $|\hat{K} - K^*|/K^*$; D = decoupled − realistic, mean over the 80 dataset–seed pairs with a 90% CI; "equivalent": the CI lies within ± 0.2 (TOST); "sig. diff.": datasets with a significant difference (two-sided Wilcoxon over seeds, Holm across datasets, $p < 0.05$); "equiv.": datasets on which the partitions are equivalent. *Local*, mean over datasets: relative error of the clients' estimates $|\hat{K}_g - K_g|/K_g$, and the Spearman correlation of $\hat{K}_g$ with the true K_g and with the client size N_g (computed within each seed). Oracle-K receives the true K_g and has no local estimates.

	global: final $\hat{K}$				local: clients' $\hat{K}_g$		
Method	D [90% CI]	equivalent	sig. diff.	equiv.	rel. error	$\rho(\hat{K}_g, K_g)$	$\rho(\hat{K}_g, N_g)$
ASM-I	+0.048 [-0.029, +0.125]	yes	0/8	5/8	0.51 → 0.52	+0.10 → -0.05	+0.07 → +0.13
ASM-S	-0.081 [-0.161, -0.002]	yes	0/8	1/8	0.47 → 0.51	+0.11 → -0.03	+0.08 → +0.14
DP-GMM	-0.025 [-0.513, +0.462]	no	0/8	1/8	1.73 → 1.90	+0.02 → -0.08	+0.11 → +0.10
X-Means	-0.192 [-0.314, -0.071]	no	3/8	1/8	1.50 → 1.18	+0.28 → -0.10	+0.32 → +0.58
G-Means	-1.357 [-2.226, -0.488]	no	4/8	0/8	7.12 → 5.57	+0.52 → +0.05	+0.52 → +0.52
Oracle-K	+0.035 [-0.013, +0.084]	yes	0/8	7/8	–	–	–

Design. All numbers come from the runs of the main experiments (evaluation seeds 42–51, eight datasets). Regime B's partition of a seed is constructed from Regime A's partition of the same seed, so the two partitions are paired by seed. Clients within a seed are not independent, so the seed is the unit of replication throughout.

1. *Global count.* For each method, dataset and seed we compute the relative error of the final count, $e = |\hat{K} - K^*|/K^*$, and the paired difference $D = e_{\text{regime-b}} - e_{\text{regime-a}}$.
   - *Difference test:* a two-sided Wilcoxon signed-rank test over the ten seeds on each dataset, Holm-corrected across the eight datasets within each method.
   - *Equivalence test:* two one-sided tests (TOST). The partitions count as equivalent when the 90% confidence interval of the mean of D lies inside ± 0.2, that is, within 20% of the true number of clusters.
   - *Pooled tests:* both tests are repeated on all 80 dataset–seed pairs.
2. *Local estimates.* For each seed and partition we compute three quantities over that seed's clients:
   - the mean relative error $|\hat{K}_g - K_g|/K_g$;

   We compare the two partitions with Wilcoxon signed-rank tests over the seeds. On Waveform every client holds two classes, so the correlation with K_g is undefined there.

We compare ASM-I and ASM-S with DP-GMM, X-Means and G-Means, and with Oracle-K. Oracle-K receives the true K_g, so any change in its error is due only to the Phase 2 aggregation.

Results for the global count. Table 12 summarises the tests.

ASM. Pooled over datasets, the error of ASM's final count is statistically equivalent on the two partitions. No dataset shows a significant difference for either variant after Holm correction. Its mean relative local error changes little (ASM-I 0.51 to 0.52, ASM-S 0.47 to 0.51), and it is equivalent within ± 0.1 on four and three of the eight datasets respectively.For ASM-S the error is in fact slightly lower on Regime B, most visibly on MNIST (0.85 to 0.47) and FMNIST (0.70 to 0.44). Removing the size information therefore does not hurt it.

Oracle-K. It behaves the same way: pooled $D = +0.04$, CI $[-0.01, 0.08]$, and it is equivalent on seven of the eight datasets. This confirms that the Phase 2 aggregation is itself insensitive to the partition.

Size-based baselines. X-Means and G-Means are not robust: X-Means changes significantly on three datasets, and its pooled difference is significant ($D = -0.19$, CI $[-0.31, -0.07]$, $p = 0.03$). G-Means changes significantly on four datasets: its error rises on CIFAR-10 and EMNIST and falls on FrogA and FrogB. Because the direction differs between datasets, the pooled Wilcoxon test is not significant ($p = 0.19$), but the mean shift is large ($D = -1.36$) and equivalence fails.

DP-GMM. It shows no significant change, but its errors are so large and variable (relative errors of 10 to 19 on FMNIST and CIFAR-10) that equivalence cannot be established either (CI $[-0.51, 0.46]$).

H RUNTIME COMPLEXITY

Table 13: Dominant computational cost of the Phase-1 methods as implemented. R: split rounds; I: EM, variational or Lloyd iterations; T: DP truncation level.

Method	dominant cost of one run on N points in d dimensions	what multiplies it
ASM-I, ASM-S	$O\big(R\,[\,Nd^2 + INd\,]\big)$ + merge $O(M\,\hat{K}Nd)$	R split rounds ($\approx \log_2 \hat{K}$), $I \le 15$ EM steps, M merge passes
DP-GMM	$O\big(I\,T\,Nd\big)$	truncation $T = 200$ in every iteration, $I \le 200$
X-Means	$O\big(R\,I\,\hat{K}Nd\big)$	a full $\hat{K}$-means refit after every split round
G-Means	$O\big(R\,[\,I\,\hat{K}Nd + N \log N\,]\big)$	as X-Means, plus an Anderson–Darling test per cluster

A split test of ASM fits one and two components to the points of a single component, so a round of tests over all current components touches every point once, at cost $O(Nd^2)$ for the leading eigenvector that sets the initial split direction plus $O(INd)$ for at most $I = 15$ EM steps; because accepted splits are applied simultaneously, the number of rounds grows like $\log_2 \hat{K}$ for balanced trees, for both ASM-I and ASM-S. No step of ASM costs a factor of $\hat{K}$ in the number of points processed. DP-GMM pays its truncation level $T = 200$ in every variational iteration regardless of how many components survive, and X-Means and G-Means refit all $\hat{K}$ means after every round of splits. The merge is quadratic in $\hat{K}$ and matters only when splitting overshoots, as in the pooled setting (Table 2).