PREFERENTIAL FEDERATED LEARNING BEYOND LABEL CALIBRATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalized federated learning (pFL) assumes that clients differ, but usually fixes what each client keeps private before measuring how they differ. When clients are human annotators, disagreement is often treated as noise or as a difference in strictness that a per-client offset and scale can absorb. We isolate a component that no such calibration can capture: *preferential concept shift*, in which clients differ in their preferences, that is, in how they weigh the properties of an item, so that they label the same items but order some of them differently. A shared model then misorders some of these items for at least one client, however much data it observes. We capture this shift with a *context vector*, a small client state that tilts each client's labelling rule toward its preferences and is kept locally. Across two real-world datasets at opposite extremes of how items are shared between clients, we find that preferential concept shift is present in both, but that far fewer of its directions can be learned from the labels each client holds than significance testing detects. A shuffled-context control separates personalization gains from those due to added model capacity. Because the context is personal, it can also identify the client, so we ask which transmitted payload does. Moreover, we show that a contextual privacy audit of the client state links a client's sessions on both datasets, whereas the model update does so only when a client's sessions update the same shared parameters, as with tied item loadings. Keeping the state local closes the stronger channel at no cost to accuracy.

1 INTRODUCTION

Annotators often disagree when labelling subjective content, and much of this disagreement reflects real differences in judgement rather than noise (Aroyo & Welty, 2015; Uma et al., 2021; Davani et al., 2022). In federated learning, each annotator can act as a *client*: its labels stay local, and a server trains a shared model from the clients' updates. This matters when the labels are themselves private, e.g., ratings of one's own life-log photographs or feedback on an assistant's replies. Annotation datasets that keep rater identifiers are the public proxy for such clients. A single shared model, however, cannot serve clients who judge the same items differently. Personalized federated learning (pFL) therefore gives each client extra parameters, such as a private output head (Collins et al., 2021) or a whole private model (Li et al., 2021), or a client embedding that the server holds (Shamsian et al., 2021). FLEAD (Rodríguez-Barroso et al., 2024) applies this idea to annotator disagreement. Yet these methods fix what each client keeps private before asking what the clients actually disagree about. If clients differ only in strictness, so that they order all items alike, a level and a scale per client suffice. If they differ in their *preferences*, that is, in which items they label higher, no shared model, even with a per-client level and scale, can represent them: averaging silently overrules the minority. We call this *preferential concept shift*. For example, take two chatbot conversations, one containing a racial slur and one a gender stereotype. Two annotators may both rate either conversation as more harmful than a benign one, yet the first rates the slur as more harmful than the stereotype, and the second rates the stereotype as more harmful than the slur.

Three ways in which clients disagree. To make this precise, we describe each client k by its *labelling rule*: the label it gives, on average, to an item. An item is described by its properties (for a conversation, the kind of harm it involves, such as a health question or a racial insult), and the rule weighs them, $y = \beta_k + s_k^\top x$ (Section 3 writes this rule on the log-odds scale as the client's *score*).

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.

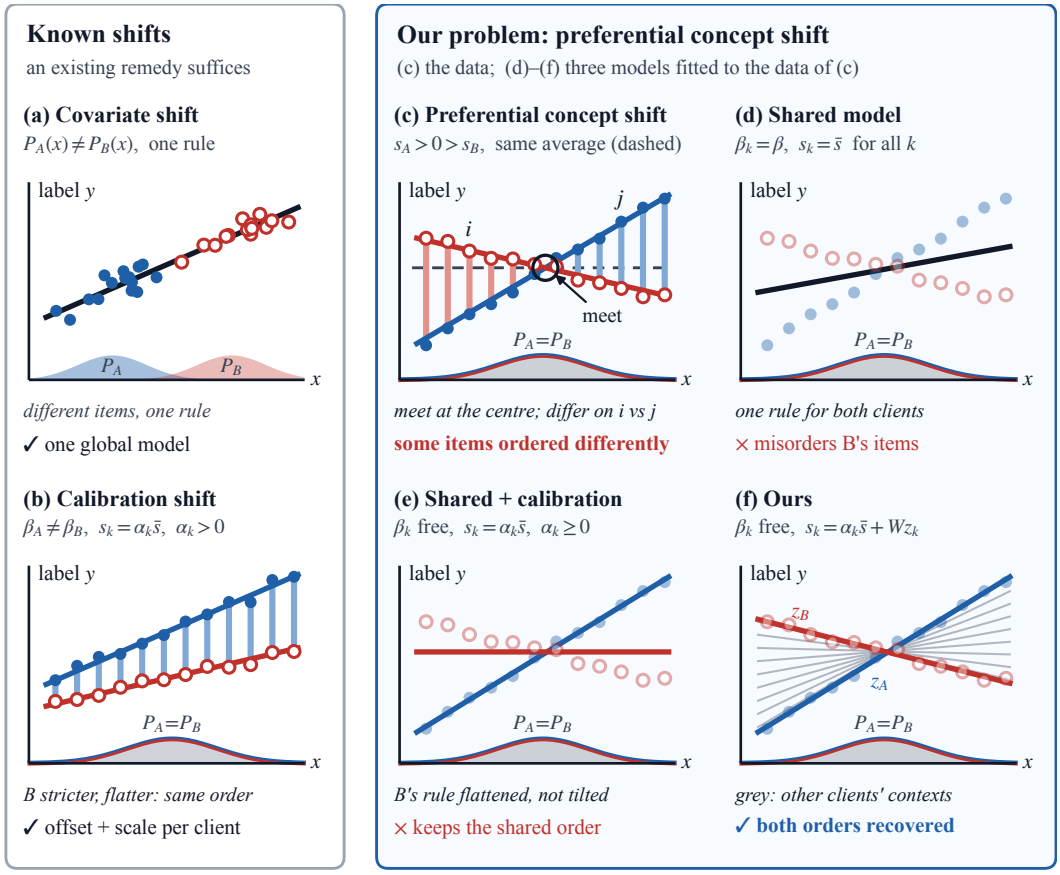

Figure 1: Clients A (filled) and B (open); x is the item property on which they differ, and vertical segments join the two labels of one item. Left: two known kinds of disagreement, each handled by an existing remedy. Right: preferential concept shift (c), and three models fitted to its data (d–f).

The level β_k sets how often client k gives a positive label, and the weights s_k set which properties it attends to; averaging over clients gives the population weights $\bar{s}$. Figure 1 shows one property of x at a time. Two clients A and B can then disagree in three ways. Under *covariate shift* (a), they see different items but label them by the same rule, so one shared model fits both. Under *calibration shift* (b), they label the same items, but one client is stricter or less decisive: its weights are a rescaled copy of the population's, $s_k = \alpha_k \bar{s}$ with $\alpha_k > 0$. Its rule is lower or flatter, yet both clients still rank every pair of items in the same order, so a per-client offset and scale, a per-client recalibration of the shared rule, remove the difference. Both cases, and their remedies, are well studied (left box).

Preferential concept shift. Under preferential concept shift (c), the clients differ in their *preferences*: they weigh the properties of an item differently. Along x, client A's label rises while client B's falls. This does not make the two clients opposites. They have the same average label, they meet near the centre, where they give items the same label, and on properties where their preferences agree they order items alike. They disagree only on the order of *some* pairs of items, those that differ mainly along x: A labels item j higher than item i, while B labels i higher than j. The disagreement is not about how strict a client is but about *which* items it labels higher, that is, about its preferences. The dataset-shift literature calls any change in the labelling rule concept shift (Moreno-Torres et al., 2012; Kairouz et al., 2021), which merges (b) and (c). We separate them because they need different remedies; Definition 3.1 states the condition formally.

Why a context vector. The right box of Figure 1 fits three models to the data of (c). A shared model (d) has one rule, so on every pair of items where the clients disagree it ranks the pair wrongly for one of them, however much data it sees: training settles the disagreement in favour of whichever client dominates the data. Adding a per-client offset and scale (e) lets each client shift or stretch the shared rule, but the scale must stay positive, since calibration keeps the order of items, so for client B it can only flatten the rule. Neither model can change the *direction* of a client's weights, which is where its preferences lie. Our model adds this control through a small *context vector* $z_k \in \mathbb{R}^d$

and sets $s_k = \alpha_k \bar{s} + W z_k$ (f). The population weights $\bar{s}$ and the map W, whose directions we call *context axes*, are shared and learned by the federation; the level β_k, scale α_k, and context z_k, together the *client state*, are learned and kept locally. With $z_k = 0$ the model reduces to calibration (e); a nonzero context tilts the client's rule, changing the order only of items that differ along the context axes. In (f), each grey line is the rule of another client with a different context. In short, the level shifts a client's rule, the scale stretches it, and the context tilts it.

Research questions. Measuring preferential concept shift in real data raises four questions. **(RQ1)** *Do clients differ beyond calibration, and do they order items differently?* **(RQ2)** *How many directions of this shift can the labels a client holds support?* **(RQ3)** *Is a gain from the client state personal, or only added model capacity?* **(RQ4)** *Which transmitted payload, the client state or the model update, links a client's sessions?* We answer them with one procedure applied to two datasets at opposite extremes of how items are shared between clients: in DICES-350 (Aroyo et al., 2023) every rater labels every item, while in R3 (Xu et al., 2021) no two subjects label the same image.

Contributions.

- **A test that separates preference from strictness (RQ1, Section 3).** *We split every client's score exactly into calibration (level, scale) and a concept component, define preferential concept shift as a partial disagreement in item order, and derive an ordering statistic that is exactly zero under any per-client calibration. On both datasets, clients differ beyond calibration and order items differently.*
- **A client state sized by what the labels support (RQ2).** *We define the context vector as the best rank-d summary of the concept components and separate the dimension a significance test detects from the dimension held-out prediction supports. Far fewer directions are learnable than detectable, and the learnable number grows with the labels each client holds, so each client keeps only a level, a scale and a few context coordinates.*
- **An exact control for personalization (RQ3).** *A shuffled-context permutation test on one trained model, with zeroed contexts and a refit on permuted labels, ties the gain to the client rather than to added capacity, for existing and new clients alike. Inside one federated model, adding a level, a scale and then the context improves or matches NLL, accuracy and macro-F1 at every step, within one standard deviation of the best personalized baseline.*
- **A channel-level privacy audit (RQ4).** *One linkage attack applied to both transmitted channels shows that the client state links a client's sessions on both datasets, while the model update does so only when tied item loadings make a client's sessions update the same shared parameters. Keeping the state local closes the stronger channel at no cost in accuracy and leaves the update to secure aggregation.*

2 Related work

Existing work. Perspectivist work treats annotator disagreement as signal rather than noise (Aroyo & Welty, 2015; Uma et al., 2021). Annotator models capture how strict or reliable an annotator is (Dawid & Skene, 1979; Hovy et al., 2013; Xu et al., 2021), or give each annotator a head (Davani et al., 2022), an embedding (Deng et al., 2023) or a group-aware model (Gordon et al., 2022; Fleisig et al., 2023; Prabhakaran et al., 2024); pluralistic alignment models the same kind of heterogeneity in preferences over pairs of responses (Sorensen et al., 2024; Poddar et al., 2024), and the role of demographics remains contested (Xu et al., 2026; Orlikowski et al., 2023; 2025). The dataset-shift taxonomy groups every change in $P_k(y \mid x)$ under concept shift (Moreno-Torres et al., 2012), distinct from label shift, which also changes $P_k(x)$ (Lipton et al., 2018; Saerens et al., 2002), and from concept drift over time (Gama et al., 2014); measurement research knows the level and the scale as rater severity and central tendency (Myford & Wolfe, 2003). In federated learning, FLEAD (Rodríguez-Barroso et al., 2024) treats each annotator as a client and trains one global model, as FedAvg (McMahan et al., 2017) and FedProx (Li et al., 2020) do; clustered methods learn one model per group of clients (Ghosh et al., 2020; Jothimurugesan et al., 2023); and pFL methods keep a private head (Collins et al., 2021), a client embedding (Shamsian et al., 2021; Scott et al., 2024; Zakerinia et al., 2026) or a whole model (Li et al., 2021; Dinh et al., 2020; Fallah et al., 2020). Attacks on federated learning reconstruct training data or infer membership and properties (Zhu et al., 2019; Geiping et al., 2020; Shokri et al., 2017; Melis et al., 2019; Lam et al., 2021; Du et al., 2025).

What is new. Our estimator belongs to a familiar family: a logistic factor model with a per-client intercept and slope, as in multidimensional item response theory (Reckase, 2009) and matrix factorization with user biases (Salakhutdinov & Mnih, 2008; Koren et al., 2009); the level and scale are a per-client Platt recalibration (Platt, 1999) of the shared score. The contribution lies in what we ask of this family: *an order-based criterion that separates the heterogeneity recalibration can absorb, rater severity and central tendency, from the preferential concept shift it cannot, with a statistic that is exactly zero under recalibration; the gap between detectable and learnable directions, which sizes the private state from the labels a client holds; an exact control that ties the gain to the client; and an audit of which transmitted payload links a client, rather than what data it reveals*. FLEAD's global model is the shared model of Figure 1(d), and clustered methods assume a few groups, whereas preferences vary continuously along shared context axes. Methods in the same function class, such as client embeddings and FedRep, reach the same accuracy once their dimension matches ours (Table 1), as expected; the split is what makes the private component interpretable and testable.

3 METHODOLOGY

We learn one model with two parts: an *item side*, shared and learned by the federation, and a small *client state*, learned and kept locally by each client (Figure 2). The model is built in four steps, each derived from the one before: (1) compare every client with the population to split its scores into a *calibration* part (how strict the client is) and a *concept* part (what it prefers); (2) find the directions of the item space along which the concept parts of clients vary, the *context axes*; (3) place every client on these axes, which gives its *context vector*; (4) estimate all of these quantities with one federated model.

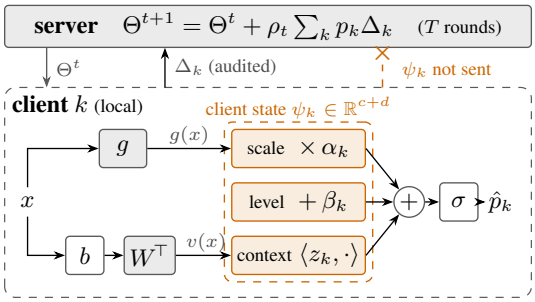

Figure 2: **One client in one round.** Grey: the shared item side $\Theta = (g, W)$, received each round. Orange: the client state $\psi_k = (\beta_k, \alpha_k, z_k)$, fitted and kept locally. Only the update Δ_k is sent; it depends on the client's context (Lemma C.6), so the contextual privacy audit tests it.

Setup. Clients $k = 1, \dots, K$ label items x from an item space $\mathcal{X}$. Client k holds $\mathcal{D}_k = \{(x_j, y_{kj}) : j \in \mathcal{J}_k\}$, the $n_k = |\mathcal{J}_k|$ items it labelled, drawn from its item distribution $P_k(x)$, with labels drawn from its own law $P_k(y \mid x)$. We describe client k by its *score* $\eta_k(x) = \operatorname{logit} P_k(y = 1 \mid x)$, the log-odds that it labels item x positive; for ordinal labels, η_k is the location of a cumulative-logit model with shared cut-points. Let $p_k = n_k / \sum_l n_l$ be client k's share of the labels and $P = \sum_k p_k P_k(x)$ the pooled item distribution. Scores are functions in $L^2(P)$, and averages over items are taken under P: $\mathbb{E}_P[f] = \mathbb{E}_{x\sim P}[f(x)]$, with inner product $\langle f, h \rangle_P = \mathbb{E}_P[fh]$ and norm $\|f\|_P^2 = \langle f, f \rangle_P$; with J shared items these are plain averages over the J items. The design is *crossed* if all clients label the same items and *disjoint* if no two clients share an item.

Step 1: calibration and concept components. The *population score* $g = \sum_k p_k \eta_k$ is the average client's score, assumed non-constant under P. To separate strictness from preference, we regress each client's score on the population score, with items as data points: $\alpha_k = \frac{\operatorname{Cov}_P(\eta_k, g)}{\operatorname{Var}_P(g)}$, $\beta_k = \mathbb{E}_P[\eta_k] - \alpha_k \mathbb{E}_P[g]$, $\gamma_k = \eta_k - \beta_k - \alpha_k g$. This splits every score exactly into level, scale and concept, $\eta_k = \beta_k + \alpha_k g + \gamma_k$. The level β_k and scale α_k are the client's *calibration*: how much more often it labels items positive and how sharply it follows the population; together they are a per-client Platt recalibration (Platt, 1999) of the population score. The *concept component* γ_k is the regression residual, what strictness cannot explain. As a least-squares residual it has zero mean over items and no correlation with g; and because the scores average to g, the calibrations average to $(0, 1)$ and the concept components to $\sum_k p_k \gamma_k = 0$. The split is not a model: it holds for any scores. Affine calibration uses $c = 2$ numbers per client; a richer increasing calibration, such as client-specific cut-points for ordinal labels, raises c without changing anything below.

Definition 3.1 (Client heterogeneity)**.** Clients exhibit (i) *covariate shift* if $P_k(x) \neq P_l(x)$ for some k, l; (ii) *calibration shift* if $(\beta_k, \alpha_k) \neq (0, 1)$ for some k, with $\alpha_k > 0$; and (iii) *concept shift* if $\|\gamma_k\|_P > 0$ for some k. Concept shift is *preferential* if some clients $k \neq l$ order some items

differently: there is a set S of item pairs (x, x'), with positive probability when x and x' are drawn independently from P, such that $\big(\eta_k(x) - \eta_k(x')\big)\big(\eta_l(x) - \eta_l(x')\big) < 0$ for all $(x, x') \in S$.

In words: client k labels x higher than x' while client l labels x' higher than x, for a non-negligible share of item pairs. With $\gamma_k = 0$ and $\alpha_k > 0$, a client's score is increasing in g, so calibration never changes the order of two items (Remark A.1). Among clients with positive scale, every preferential concept shift therefore lies in the concept components; a client with $\alpha_k \leq 0$ would invert the population order through its scale alone. All DICES scales and 39 of the 40 R3 scales are positive; the ordering null keeps the one negative scale, so the ordering test measures disagreement beyond it (Remark A.1). The converse does not hold: a γ_k that only bends a client's scores without changing any order is concept shift but not preferential. Preferential concept shift is Figure 1(c); App. A relates (ii)–(iii) to label shift, rater severity and concept drift.

Step 2: context axes. All concept components are functions on the same item space, so we ask along which directions of that space they vary most across clients. Their covariance is $C = \sum_k p_k \gamma_k \gamma_k^\top$: with J shared items, a $J \times J$ matrix; in general, the operator on $L^2(P)$ that maps f to $\sum_k p_k \langle \gamma_k, f \rangle_P \gamma_k$. It is symmetric and positive semidefinite, with rank $r \leq K - 1$ because $\sum_k p_k \gamma_k = 0$. Its eigenvectors $\phi_1, \phi_2, \ldots$, orthonormal with eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq 0$, are the *context axes*: $\phi_i(x)$ is the position of item x on axis i, and λ_i measures how much clients disagree along it. Each axis is orthogonal to the population score, so it describes *what* clients prefer, not how strict they are. On a crossed design, the singular values that the spectrum test of RQ1 compares with its null estimate these disagreements on the label scale (App. B.4).

Step 3: the context vector. A client's position on an axis is the average, over items, of its concept component times the items' positions on that axis: $z_{k,i} = \langle \gamma_k, \phi_i \rangle_P = \mathbb{E}_P\big[\gamma_k(x)\, \phi_i(x)\big]$. It is positive if the client scores items above its calibration where they sit at the positive end of the axis, negative in the opposite case, and zero if its preferences are unrelated to the axis. Keeping the first d axes, with $v(x) = (\phi_1(x), \ldots, \phi_d(x))$, gives $\gamma_k(x) \approx \langle z_k, v(x) \rangle$: the client's preference is summarized by the *context vector* $z_k \in \mathbb{R}^d$. This summary is the best possible of its size:

Definition 3.2 (Context vector)**.** A *d-dimensional context* is a map $v : \mathcal{X} \to \mathbb{R}^d$, shared by all clients, together with client vectors $z_k \in \mathbb{R}^d$, minimizing $\mathcal{E}_d(v, z) = \sum_k p_k \|\gamma_k - \langle z_k, v \rangle\|_P^2$.

Proposition 3.3 (Existence, rank, gauge)**.** (a) *$v = (\phi_1, \ldots, \phi_d)$ with $z_{k,i} = \langle \gamma_k, \phi_i \rangle_P$ minimizes $\mathcal{E}_d$, and* $\min \mathcal{E}_d = \sum_{i>d} \lambda_i$. (b) *The smallest d with* $\min \mathcal{E}_d = 0$ *is r.* (c) *If $\lambda_d > \lambda_{d+1}$, only d and the span of v are identified: every other minimizer is $(A^{-\top} v, A z_k)$ with A invertible.*

The error of keeping d axes is the disagreement left on the discarded ones, and r axes reproduce every γ_k exactly. Because z_k is free in sign, a context can change the order of items where calibration cannot. By (c), the sign and units of z_k are arbitrary, so contexts are compared through principal angles between their subspaces (App. A).

Step 4: the model. Client k scores item x as $\eta_k(x) = \underbrace{\beta_k + \alpha_k\, g(x)}_{\text{calibration}} + \underbrace{\langle z_k, v(x) \rangle}_{\text{context}}$, $\quad v(x) = W^\top b(x)$, and predicts $\sigma(\eta_k(x))$, with σ the logistic function. Since the context axes are unknown, they are learned: $b(x)$ is an item representation, and W maps it to the item's coordinates on the axes, so the columns of W are the context axes expressed in that representation. The calibration model is the d=0 fit of the same objective, so every gain of the context is measured beyond calibration. The fitted axes are not constrained to be orthogonal to 1 and g; the ordering test uses only the fitted scores η_k and is unaffected by this choice of coordinates. When $g(x) = \bar{s}^\top b(x)$, client k's weights on $b(x)$ are $s_k = \alpha_k \bar{s} + W z_k$, the form of Section 1. On crossed data $b(x)$ is the item indicator (*free loadings*, one embedding per item), so any v in Definition 3.2 is attainable. On disjoint data free loadings are not identified (Proposition B.1), so $b(x)$ is a fixed encoder or covariate vector (*tied loadings*), which also extends to items no other client has labelled.

Federated training. The item side $\Theta = (g, W)$ enters every client's loss; the client state $\psi_k = (\beta_k, \alpha_k, z_k) \in \mathbb{R}^{c+d}$ enters only client k's. Training solves $\min_\Theta \sum_k p_k \min_{\psi_k} \mathcal{L}_k(\Theta, \psi_k)$, where $\mathcal{L}_k = \widehat{F}_k + \frac{\mu}{2} \|\psi_k - \psi^0\|^2$ adds to $\widehat{F}_k$, client k's empirical negative log-likelihood, a ridge penalty of strength μ towards the population state $\psi^0 = (0, 1, \mathbf{0})$ (App. B). In each of T rounds t, the server

sends Θ^t; client k fits ψ_k with Θ^t fixed, takes local steps on Θ with ψ_k fixed to reach Θ_k, and returns the update $\Delta_k = \Theta_k - \Theta^t$, with the scalar losses and curvature summaries that the server step uses; the server sets $\Theta^{t+1} = \Theta^t + \rho_t \sum_k p_k \Delta_k$, with the step ρ_t chosen by backtracking on the federated objective (App. E.1). The federation thus learns what no single client can, the population score and the context axes, while each client keeps the $c + d$ numbers that place it among them. The client state is never sent, so it adds nothing to the uplink.

Choosing d and adding clients. We fit $d = 0, \ldots, d_{\max}$ and keep the smallest d whose validation loss lies within one paired standard error of the best. This *learnable* dimension $d^\star$ can be far below the *detectable* dimension d_{spec}, the number of leading singular values of the calibration model's residual matrix that exceed their parametric null (App. F.1), which estimates r. A new client fits ψ_k from m labels with Θ frozen, under an empirical-Bayes prior centred on the training clients' mean state (App. E).

What each research question rests on. Each design choice follows from one result, stated and proved in App. C. **(RQ1)** With disjoint supports, concept shift is not identifiable without structure (Lemma C.1), so on R3 we tie loadings and test on the cells of the crossed experimental design. Calibration with positive scales never changes the order of two items (Remark A.1), so a statistic that measures ordering disagreement is exactly zero under calibration and tests preferential concept shift directly. **(RQ2)** Asymptotically, a state of $c + d$ numbers beats both a shared model and private models over a range of client sizes when the best rank-d context removes a large enough share of the shared model's error relative to the size of the state (Theorem C.3); we use this for the shape of the trade-off and choose d by held-out prediction. **(RQ3)** If the pairing of clients with contexts carries no information, the gain over a shuffled pairing has mean zero and admits an exact permutation test (Proposition C.4); a gain counts as personalization only if the own context beats every shuffled pairing and differs from the zeroed context, on every split. The shuffle tests that the gain belongs to the client, the zeroed context that the context reaches the output, and a refit on permuted labels, with the same parameters, that the gain does not come from capacity. **(RQ4)** Under a Gaussian model, the chance of linking a client grows at most linearly in the number of coordinates it releases (Proposition C.5), so the small size of the state bounds what it can reveal but does not make it safe; under gradient steps the update's context block lies along z_k (Lemma C.6), so the contextual privacy audit tests both channels.

4 Experiments

Setup. **DICES-350** (Aroyo et al., 2023) is fully crossed: 104 raters each label all 350 adversarial conversations as safe or unsafe. **R3** (Xu et al., 2021) is fully disjoint: 40 subjects rate how well they remember photographs from their own life-logs, binarized at $\text{Mem}_s \geq 5$; each photograph is described by its design variables, and the spectrum test uses the crossed design cells. *Seen items* split each client's ratings $60/20/20$; *items unseen by the client* (DICES) split the items, so a test item was labelled only by other clients; *new clients* are held out whole and supply their first m labels. Gains are reductions in test negative log-likelihood (NLL, bits per rating) against the *calibration* model, which keeps a per-client level and scale. RQ(1,2,3) fit this objective centrally, which the federated model matches within 0.01 bits (Table 15); the federated comparison and RQ4 use the federated model. All experiments share five frozen splits, one model family and one code base. Protocols, per-split results and every number behind the figures are in App. F.

RQ1: clients differ beyond calibration, and they order items differently. A difference in strictness alone can create several interaction directions through the logistic link, so we test against a null that contains both the level and the scale of every client. We fit the calibration model, simulate labels from it, refit it on every draw, and compare the singular values of the residual matrix with their null distribution. On both datasets the leading directions lie far outside the null (Figure 3a); on synthetic data generated from the calibration model the test rejects at most 2 of 10 datasets (App. F.1). Clients therefore disagree beyond strictness.

Rejecting calibration does not yet show that clients *order* items differently, because an order-preserving distortion of the shared rule also leaves residual structure. Under calibration with positive scales, any two clients rank every pair of items alike, so we measure ordering directly: the average size of the pairwise disagreements $\max\{0, -(\eta_{ka} - \eta_{kb})(\eta_{la} - \eta_{lb})\}$ between clients' fitted rules, against

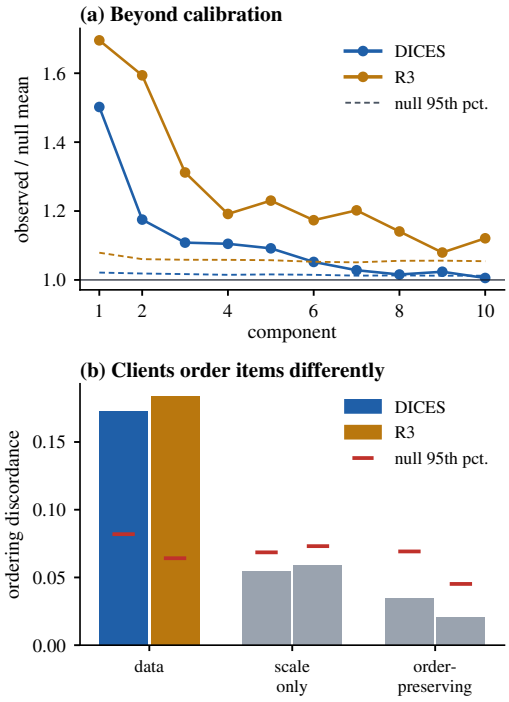

Figure 3: (a) Residual spectrum over a null with per-client level *and* scale, refitted on every draw. (b) Ordering disagreement of the fitted clients against the same null, on the data and on two synthetic worlds with no ordering differences.

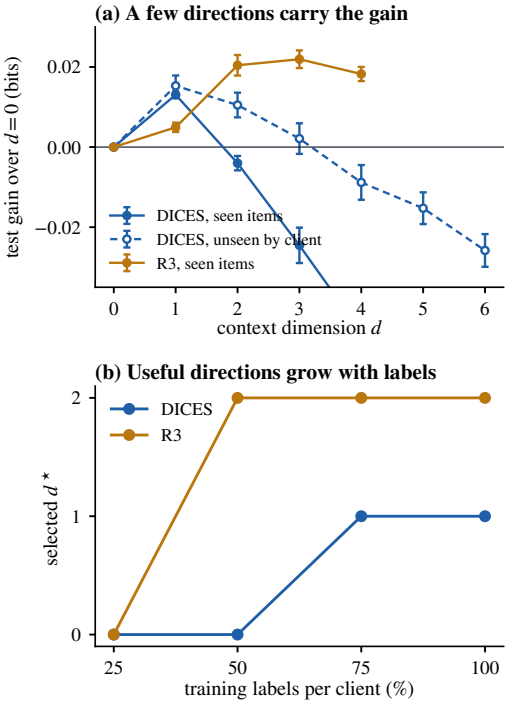

Figure 4: (a) Test gain over calibration by context dimension (mean $\pm$ SE, five splits). (b) Dimension selected when clients keep part of their training labels.

the same null. The data exceed the null several times over on both datasets, while a world with only scale differences and a world with an order-preserving distortion both stay inside it (Figure 3b); the second world is the direct check that a nonlinear but order-preserving rule does not pass as preference. The disagreement is partial, as in Definition 3.1: two clients agree on most orders and differ on some. On R3 the leading direction is consistent with a forgetting rate: after Benjamini–Hochberg correction, 12 of 40 subjects are significantly less likely to remember a photograph the longer its retention interval, while 2 are significantly more likely. Beyond its leading directions the R3 spectrum sits on a plateau above the null, as photographs within a design cell differing in content would produce (Remark B.2), so on R3 we interpret the leading directions, which the ordering test and the controls of RQ3 support, rather than a count.

RQ2: a few directions carry the predictive gain, and their number grows with the labels. The significance test detects many directions on both datasets (App. F.1), though on R3 part of them reflect content within design cells; a federated client needs to know how many improve its predictions. We fit the context model with increasing dimension d and select $d^\star$ on validation with a one-standard-error rule on the paired split-by-split difference. One direction is selected on DICES, both on seen items and on items unseen by the client, and two on R3; each gains on all five splits, and further DICES directions add no gain on seen items (Figure 4a). The selected subspace is stable across splits, far above random subspaces. The client state is therefore small, $c + d^\star = 3$ numbers on DICES and 4 on R3.

The gap between detected and useful directions is a property of the labels each client holds. When clients keep only part of their training labels, fewer directions are selected, down to none with a quarter of the labels on both datasets; the useful dimension grows as clients hold more labels (Figure 4b). A state sized by a significance test would carry directions that a client cannot estimate; sizing it on held-out prediction gives a state that grows with the client's data.

RQ3: the gain is personal, not added capacity. A context adds parameters, so a gain could come from capacity rather than from the client. We separate the two on a single trained checkpoint per split, so that only the pairing of clients with contexts changes: each client uses its

Table 1: Federated comparison, mean ± s.d. over three splits; new clients: DICES m=100, R3 m=200. NLL in bits; F1 is macro-F1. *State*: numbers each client keeps locally. Bold: best in column; underlined: within one s.d. of the best. [a]Index of one of three shared models.

		Seen items			New clients		
Method	State	NLL ↓	Acc. ↑	Macro-F1 ↑	NLL ↓	Acc. ↑	Macro-F1 ↑
DICES-350							
FedAvg	0	0.792 ±0.017	0.733 ±0.006	0.640 ±0.004	0.773 ±0.046	0.743 ±0.025	0.654 ±0.021
+ level	1	0.660 ±0.008	0.781 ±0.005	0.732 ±0.005	0.645 ±0.023	0.788 ±0.012	0.737 ±0.006
+ level + scale	2	0.645 ±0.010	0.786 ±0.005	0.742 ±0.006	0.631 ±0.024	0.790 ±0.011	0.743 ±0.004
Ours (+ context)	3	0.629 ±0.010	0.792 ±0.003	**0.751** ±0.003	0.615 ±0.025	0.796 ±0.009	0.752 ±0.006
Client embedding	3	**0.628** ±0.010	**0.793** ±0.003	0.750 ±0.003	**0.613** ±0.024	**0.798** ±0.012	**0.754** ±0.007
FedRep	3	0.629 ±0.010	**0.793** ±0.005	0.750 ±0.006	**0.613** ±0.025	0.797 ±0.013	0.752 ±0.005
IFCA	1[a]	0.709 ±0.016	0.766 ±0.007	0.714 ±0.010	0.690 ±0.035	0.768 ±0.019	0.717 ±0.010
Ditto	351	0.664 ±0.008	0.780 ±0.005	0.728 ±0.006	0.656 ±0.022	0.786 ±0.012	0.728 ±0.007
R3							
FedAvg	0	0.968 ±0.005	0.591 ±0.007	0.590 ±0.007	0.991 ±0.011	0.562 ±0.008	0.559 ±0.007
+ level	1	0.886 ±0.011	0.658 ±0.012	0.657 ±0.012	0.911 ±0.042	0.635 ±0.034	0.620 ±0.029
+ level + scale	2	0.885 ±0.012	0.660 ±0.015	0.660 ±0.015	0.905 ±0.043	0.640 ±0.031	0.623 ±0.026
Ours (+ context)	4	0.871 ±0.013	0.671 ±0.015	0.670 ±0.015	0.892 ±0.044	0.663 ±0.031	0.641 ±0.028
Client embedding	3	0.873 ±0.011	0.670 ±0.019	0.670 ±0.019	0.894 ±0.041	0.663 ±0.029	0.643 ±0.022
FedRep	3–4	0.872 ±0.008	0.673 ±0.016	0.673 ±0.016	0.890 ±0.042	0.666 ±0.037	0.645 ±0.038
IFCA	1[a]	0.898 ±0.001	0.653 ±0.006	0.653 ±0.006	0.943 ±0.037	0.622 ±0.037	0.608 ±0.043
Ditto	7	**0.860** ±0.013	**0.679** ±0.014	**0.678** ±0.014	**0.860** ±0.036	**0.683** ±0.030	**0.664** ±0.024

own context, the context is set to zero, or the contexts are permuted across clients 500 times. On every split of both datasets the own context gains, the zeroed context does not, and even the best of the 500 permutations loses (Figure 5a). A refit after permuting each client's labels across its items keeps every parameter but breaks the link between items and labels, and its mean gain is negative on both datasets, so the gain comes from the client's labels rather than from capacity. The context is estimated from a client's own labels and is of no use to anyone else. The same holds for clients the federation has never seen. A new client's state is read from its first m labels with the shared model frozen, shrunk towards the training clients' mean state in proportion to how much each coordinate varies across them (the empirical-Bayes readout of App. E). The own context improves on a calibration-only readout from $m = 10$ on both datasets; the gain grows with m on R3 and levels off from $m = 50$ on DICES, whereas another new client's context is worse at every $m \geq 5$ (Figure 5b). In our sample, demographics explain no more of the context than chance. The context is therefore behavioural: it is read from what a client labels, not from who the client is. A new client should contribute a few dozen labels, not a demographic profile.

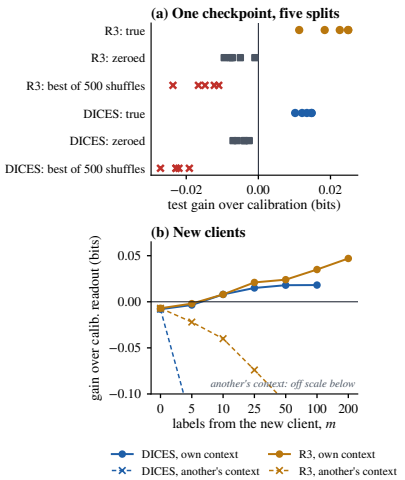

Figure 5: (a) One trained checkpoint per split: own, zeroed, and best of 500 shuffled contexts. (b) New clients: gain over a calibration readout from the same m labels.

Federated comparison: each part of the client state earns its place. We now train the same model federatedly and compare it with other ways of personalizing. All federated arms share one model family, item representation, optimizer and budget of 100 rounds, and differ only in what a client keeps. The nested ablation isolates the mechanism inside the actual federated model (Table 1): adding a level, then a scale, then the context improves or matches NLL, accuracy and macro-F1 at every step, on both datasets and for both seen items and new clients, and on one federated checkpoint, shuffling the contexts across clients turns the context's gain into a loss. For new clients, ours improves on FedAvg and on its own shuffled state by $m = 5$, the smallest support tested, on both datasets. Against other ways of personalizing, ours is within one standard deviation of the best method in every column of both datasets while keeping only 3–4 numbers per client. The client embedding in the style of Deng et al. (2023) and FedRep keep states of similar size and perform alike, as expected, since at matched dimension they span nearly the same functions; only ours separates strictness from preference, so its context can be tested and controlled (RQ1–RQ3). Clustered FL (IFCA) trails on both datasets. The methods that lead on R3, Ditto keep a full personal model, which is affordable

there because each subject contributes about 265 ratings on six design variables, and which DICES, with one parameter per item, punishes. Every arm passes a quality gate (better than the base rate and the majority class), and the federated models come within 0.01 bits of the central optimum of the same objective. The deployment rule is therefore the one the analysis predicts: keep a level, a scale and a small context locally, and size the context on held-out prediction.

RQ4: the client state is the channel to keep local. Our contextual privacy audit asks whether an honest-but-curious server can link two pseudonymous payloads of the same client, for example from two sessions. Each client's training ratings are split into two halves, each payload is computed from each half on the trained federated model, and the server pairs the halves by nearest neighbour under three matchers, against a permutation null. The client state links clients far above chance on both datasets and under all three matchers (Figure 6), which matters wherever a server holds client embeddings. The full federated update links fewer: none beyond chance on DICES, where per-item loadings make a rater's two sessions update different parameters, and some R3 subjects, where tied loadings make them update the same ones. Keeping the state local therefore closes the stronger channel, at no cost in accuracy because the model is unchanged, whereas per-release differential privacy on the state removes its gain at every privacy level we tested. One of 30 attribute tests exceeds its minimum detectable effect, the age of R3 subjects from the payload that combines the state and the update, while the state alone exceeds no threshold; the update is therefore the channel that secure aggregation should protect.

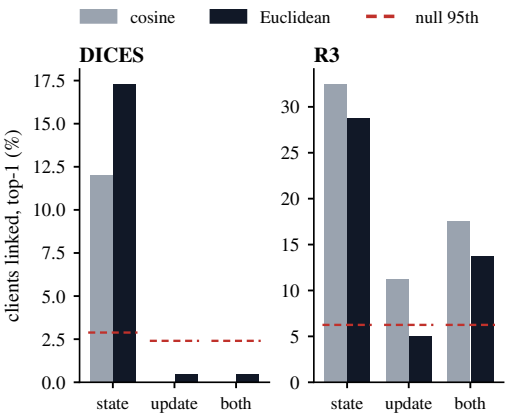

Figure 6: Share of clients whose two pseudonymous payloads are linked (top-1), by payload and matcher (standardized cosine and Euclidean; raw cosine in Table 18); dashed: permutation-null 95th percentile.

5 Conclusion and Limitations

Clients who label subjective content differ in strictness and in preference, and only the second changes which items they label higher. Splitting each client's score into calibration and a concept component makes this difference testable: on a fully crossed and a fully disjoint dataset, clients order items differently beyond what calibration explains (RQ1). A few directions of this shift carry the predictive gain, and their number grows with the labels a client holds (RQ2); the gain is tied to the client, for existing and new clients alike (RQ3); and the client state, more than the update, is the channel that links a client's sessions (RQ4). The deployment rule follows: keep a level, a scale and a small context locally, size the context on held-out prediction, and protect the update with secure aggregation when tied item loadings make a client's sessions update the same parameters. More broadly, our results suggest sizing the private component of personalized federated learning by what held-out prediction supports rather than by significance or by architecture.

Limitations. Our evidence comes from two datasets chosen at opposite extremes of item sharing, with 104 and 40 clients; other designs and larger federations remain to be measured. The analysis covers binary labels (ordinal labels enter through a cumulative-logit score with shared cut-points), a context that is linear in a per-item or fixed item representation, and full client participation. On R3 the item side is six design variables, so photograph content they cannot represent raises the residual spectrum (Remark B.2); we therefore interpret its leading directions rather than a count. The joint fit is non-convex, so the learnable dimension is a property of our estimator at these data sizes as well as of the data (App. D). The privacy audit assumes an honest-but-curious server that links payloads rather than reconstructing data; it examines the update but not the scalar losses and curvature summaries the server step uses, and secure aggregation is assumed rather than implemented. Extending the split to learned item encoders and partial participation, and bounding the update channel under secure aggregation, are natural next steps.

REFERENCES

Lora Aroyo and Chris Welty. Truth is a lie: Crowd truth and the seven myths of human annotation. *AI Magazine*, 36(1):15–24, 2015.

Lora Aroyo, Alex S. Taylor, Mark Díaz, Christopher M. Homan, Alicia Parrish, Gregory Serapio-García, Vinodkumar Prabhakaran, and Ding Wang. DICES dataset: Diversity in conversational AI evaluation for safety. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2023.

Liam Collins, Hamed Hassani, Aryan Mokhtari, and Sanjay Shakkottai. Exploiting shared representations for personalized federated learning. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Aida Mostafazadeh Davani, Mark Díaz, and Vinodkumar Prabhakaran. Dealing with disagreements: Looking beyond the majority vote in subjective annotations. *Transactions of the Association for Computational Linguistics*, 10:92–110, 2022.

A. P. Dawid and A. M. Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

Naihao Deng, Xinliang Frederick Zhang, Siyang Liu, Winston Wu, Lu Wang, and Rada Mihalcea. You are what you annotate: Towards better models through annotator representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Canh T. Dinh, Nguyen H. Tran, and Tuan Dung Nguyen. Personalized federated learning with Moreau envelopes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Jiacheng Du, Jiahui Hu, Zhibo Wang, Peng Sun, Neil Zhenqiang Gong, Kui Ren, and Chun Chen. SoK: On gradient leakage in federated learning. In *Proceedings of the 34th USENIX Security Symposium*, 2025.

Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Eve Fleisig, Rediet Abebe, and Dan Klein. When the majority is wrong: Modeling annotator disagreement for subjective tasks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37, 2014.

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients – how easy is it to break privacy in federated learning? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Mitchell L. Gordon, Michelle S. Lam, Joon Sung Park, Kayur Patel, Jeffrey T. Hancock, Tatsunori Hashimoto, and Michael S. Bernstein. Jury learning: Integrating dissenting voices into machine learning models. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 2022.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard Hovy. Learning whom to trust with MACE. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pp. 1120–1130, 2013.

Ellango Jothimurugesan, Kevin Hsieh, Jianyu Wang, Gauri Joshi, and Phillip B. Gibbons. Federated learning under distributed concept drift. In *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Proceedings of Machine Learning Research, 2023.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, et al. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210, 2021.

Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

Maximilian Lam, Gu-Yeon Wei, David Brooks, Vijay Janapa Reddi, and Michael Mitzenmacher. Gradient disaggregation: Breaking privacy in federated learning by reconstructing the user participant matrix. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and Systems (MLSys)*, 2020.

Tian Li, Shengyuan Hu, Ahmad Beirami, and Virginia Smith. Ditto: Fair and robust federated learning through personalization. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Zachary Lipton, Yu-Xiang Wang, and Alexander Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3122–3130, 2018.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (S&P)*, 2019.

Jose G. Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V. Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

Carol M. Myford and Edward W. Wolfe. Detecting and measuring rater effects using many-facet Rasch measurement: Part I. *Journal of Applied Measurement*, 4(4):386–422, 2003.

Matthias Orlikowski, Paul Röttger, Philipp Cimiano, and Dirk Hovy. The ecological fallacy in annotation: Modeling human label variation goes beyond sociodemographics. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1017–1029, 2023.

Matthias Orlikowski, Jiaxin Pei, Paul Röttger, Philipp Cimiano, David Jurgens, and Dirk Hovy. Beyond demographics: Fine-tuning large language models to predict individuals' subjective text perceptions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2092–2111, 2025.

John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*. MIT Press, 1999.

Sriyash Poddar, Yanming Wan, Hamish Ivison, Abhishek Gupta, and Natasha Jaques. Personalizing reinforcement learning from human feedback with variational preference learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Vinodkumar Prabhakaran, Christopher Homan, Lora Aroyo, et al. GRASP: A disagreement analysis framework to assess group associations in perspectives. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2024.

Mark D. Reckase. *Multidimensional Item Response Theory*. Springer, 2009.

Nuria Rodríguez-Barroso, Eugenio Martínez Cámara, Jose Camacho Collados, M. Victoria Luzón, and Francisco Herrera. Federated learning for exploiting annotators' disagreements in natural language processing. *Transactions of the Association for Computational Linguistics*, 12, 2024.

Marco Saerens, Patrice Latinne, and Christine Decaestecker. Adjusting the outputs of a classifier to new a priori probabilities: A simple procedure. *Neural Computation*, 14(1):21–41, 2002.

Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2008.

Jonathan Scott, Hossein Zakerinia, and Christoph H. Lampert. PeFLL: Personalized federated learning by learning to learn. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024.

Aviv Shamsian, Aviv Navon, Ethan Fetaya, and Gal Chechik. Personalized federated learning using hypernetworks. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, 2021.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (S&P)*, 2017.

Taylor Sorensen et al. Position: A roadmap to pluralistic alignment. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Alexandra N. Uma, Tommaso Fornaciari, Dirk Hovy, Silviu Paun, Barbara Plank, and Massimo Poesio. Learning from disagreement: A survey. *Journal of Artificial Intelligence Research*, 72: 1385–1470, 2021.

Qianli Xu, Fen Fang, Ana Garcia del Molino, Vigneshwaran Subbaraju, and Joo-Hwee Lim. Predicting event memorability from contextual visual semantics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Yinuo Xu, Veronica Derricks, Allison Earl, and David Jurgens. Modeling annotator disagreement with demographic-aware experts and synthetic perspectives. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026.

Hossein Zakerinia, Jonathan Scott, and Christoph H. Lampert. Federated learning with unlabeled clients: Personalization can happen in low dimensions. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

A Formulation

A.1 Properties of the decomposition

The coefficients of Step 1 in §3, $\alpha_k = \mathrm{Cov}_P(\eta_k, g)/\mathrm{Var}_P(g)$ and $\beta_k = \mathbb{E}_P[\eta_k] - \alpha_k \mathbb{E}_P[g]$, are those of the orthogonal projection of η_k onto $\mathrm{span}\{1, g\}$ in $L^2(P)$, and γ_k is its residual; as a least-squares residual, γ_k has zero mean and is uncorrelated with g. The projection is linear and maps g to itself with coefficients $(0, 1)$. Applying it to $g = \sum_k p_k \eta_k$ therefore gives $\sum_k p_k \beta_k = 0$ and $\sum_k p_k \alpha_k = 1$, and the residuals satisfy $\sum_k p_k \gamma_k = g - g = 0$; consequently the contexts also average to zero, $\sum_k p_k z_{k,i} = \langle \sum_k p_k \gamma_k, \phi_i \rangle_P = 0$. The split is unique whenever g is non-constant under P, so that 1 and g are linearly independent. The average client is therefore $(\beta, \alpha, \gamma) = (0, 1, 0)$, which is why $\psi^0 = (0, 1, \mathbf{0})$ serves as the population state. With affine calibration the state has $c = 2$ calibration coordinates; a richer increasing calibration changes c but none of the results below.

A.2 Calibration never changes an order

Remark A.1 (Order). If $\gamma_k = \gamma_l = 0$ and $\alpha_k, \alpha_l > 0$, then for every pair of items

$$\{\eta_k(x) - \eta_k(x')\}\{\eta_l(x) - \eta_l(x')\} = \alpha_k \alpha_l \{g(x) - g(x')\}^2 \;\geq\; 0.$$

Calibration shift with a positive scale therefore never changes the order of two items, and preferential concept shift (Definition 3.1) requires $\gamma \neq 0$ for at least one of the two clients. The converse does

not hold: an increasing but non-affine distortion of g has $\gamma_k \neq 0$ and changes the order of no pair. Concept shift is thus necessary, not sufficient, for clients to order items differently. This is why RQ1 has two tests: the spectrum test detects $\gamma \neq 0$, and the ordering statistic of App. F.2, which is exactly zero under calibration with positive scales, detects differences in order. A negative scale, $\alpha_k < 0$, would reverse the order of every pair at once; Definition 3.1 therefore takes $\alpha_k > 0$. In the fitted models, all 104 DICES scales are positive at every context dimension, the smallest being 0.37 at the selected $d^\star = 1$. On R3, 39 of the 40 subjects have positive scales; one subject's scale is negative (-0.19 in the calibration model, -0.38 at the selected $d^\star = 2$), so that subject's memory runs against the population score across the design. The ordering null is simulated from the fitted calibration model, including this negative scale, so the reversal it causes is part of the null, and the ordering test measures only disagreement beyond it.

A.3 Proof of Proposition 3.3

(a) For fixed v with orthonormal coordinates in $L^2(P)$, the minimiser over z_k is the projection $z_{k,i} = \langle \gamma_k, v_i \rangle_P$, and $\mathcal{E}_d = \sum_k p_k \|\gamma_k\|^2 - \sum_{i \le d} \langle v_i, C v_i \rangle_P$. Orthonormalising v does not change the span of $\{\langle z_k, v \rangle\}$, so it suffices to maximise $\sum_{i \le d} \langle v_i, C v_i \rangle_P$ over orthonormal v. By the Ky Fan principle for the compact self-adjoint covariance C, the maximum is $\sum_{i \le d} \lambda_i$, attained at $v = (\phi_1, \dots, \phi_d)$; since $\sum_k p_k \|\gamma_k\|^2 = \operatorname{tr} C = \sum_i \lambda_i$, the minimum is $\sum_{i > d} \lambda_i$. *(b)* The range of C lies in $\operatorname{span}\{\gamma_1, \dots, \gamma_K\}$, which has dimension at most $K - 1$ because $\sum_k p_k \gamma_k = 0$; hence $\mathcal{E}_d = 0$ exactly when $d \ge r$. *(c)* $\langle A z_k, A^{-\top} v \rangle = z_k^\top A^\top A^{-\top} v = \langle z_k, v \rangle$, so all such pairs attain the same error. Conversely, when $\lambda_d > \lambda_{d+1}$ every minimiser has $\operatorname{span}(v) = \operatorname{span}(\phi_1, \dots, \phi_d)$, and two bases of this span differ by an invertible A. Requiring v orthonormal restricts A to be orthogonal, and ordering by distinct λ_i leaves only the signs of the coordinates. □

Because of this $\mathrm{GL}(d)$ invariance, fitted contexts are never compared coordinate by coordinate: all comparisons across splits use principal angles between context subspaces, which do not depend on the gauge.

B Method details

This section gives the algorithms behind §3 and the results on new clients; formal statements of the guarantees are in App. C.

Table 2: Notation. Shared objects are learned by the federation; the client state is kept locally by each client.

Symbol	Meaning	Held by
η_k, g	client score (log-odds); population score $g = \sum_k p_k \eta_k$	— / shared
β_0	global intercept, absorbed into g	shared
$\beta_k,\ \alpha_k,\ \gamma_k$	level, scale and concept component of η_k (Step 1)	—
$C,\ \lambda_i,\ r$	covariance of the concept components, its eigenvalues and rank	—
ϕ_i	context axes: eigenvectors of C, directions of the item space (Step 2)	—
$b(x)$	item representation (item indicator on DICES, design basis on R3)	shared
$W, v(x)$	map to the context axes; item coordinates $v(x) = W^\top b(x) \in \mathbb{R}^d$	shared
$\bar{s}$	population weights on $b(x)$ when $g(x) = \bar{s}^\top b(x)$	shared
Θ	shared parameters (g, W)	shared
z_k	context vector in $\mathbb{R}^d$, $z_{k,i} = \langle \gamma_k, \phi_i \rangle_P$ (Step 3)	client
ψ_k	client state $(\beta_k, \alpha_k, z_k) \in \mathbb{R}^{c+d}$; population state $\psi^0 = (0, 1, \mathbf{0})$	client
$c,\ d$	calibration coordinates ($c = 2$: level, scale); context dimension	—
$F_k,\ \widehat{F}_k$	population and empirical risk of client k	—
$\mathcal{L}_k$	penalised loss $\widehat{F}_k + \frac{\mu}{2} \|\psi_k - \psi^0\|^2$	client
Δ_k	update sent to the server, $\Theta_k - \Theta^t$	sent
p_k	aggregation weight $n_k / \sum_l n_l$	shared
$d^\star,\ d_{\mathrm{spec}}$	learnable and detectable dimension	—
m	labelled ratings supplied by a new client	—
$\mu,\ \kappa,\ \rho_t$	ridge penalty; local step size; server step	—
$\chi,\ D_{\mathrm{loc}}$	estimation constant; dimension of a per-client model (Theorem C.3)	—

B.1 Fitting the model and choosing its dimension

Algorithm 1 fits the model of §3 for each context dimension. The $d = 0$ fit is the calibration model, and every larger d is fitted by the same penalised objective, so every gain at $d \geq 1$ is measured beyond calibration. The detectable dimension d_{spec} counts the leading singular values of the residual matrix that exceed their calibration null (App. B.4); only the learnable dimension $d^\star$ sizes the deployed state.

Algorithm 1 Fitting the context model and selecting its dimension (train and validation data)

Require: labelled ratings; frozen splits; item representation $b(x)$; dimensions $\{0, \dots, d_{\max}\}$; ridge grid for μ
Ensure: map W, dimension $d^\star$, contexts $\{z_k\}$, stability score
1: fit the calibration model ($d = 0$): shared g and every (β_k, α_k), penalised
2: choose μ on validation NLL of the calibration model
3: **for** $d = 1, \dots, d_{\max}$ **do**
4: initialise Z, W from the truncated SVD of the calibration model's client $\times$ item residuals
5: fit $(g, W, \{\beta_k, \alpha_k, z_k\})$ jointly by L-BFGS on the penalised likelihood
6: $\text{NLL}_d \leftarrow$ validation loss on each split
7: **end for**
8: $d^\star \leftarrow$ smallest d whose validation NLL exceeds the best by less than one standard error of the paired split-by-split difference
9: **stability:** principal-angle cosines between the contexts fitted on different splits
10: **return** W, $d^\star$, $\{z_k\}$, stability

Proposition B.1 (Identification of the fitted model)**.** *The context term is unchanged under $(z_k, W) \mapsto (Az_k, WA^{-1})$ for every $A \in \text{GL}(d)$, so d and the column space of W are identified but the coordinates of z_k are not. With free loadings and a single client per item, neither z_k nor $v(x_j)$ is identified for any $d \geq 1$, and the data carry no information about d.*

The first part is the sample counterpart of Proposition 3.3(c), and is why contexts are compared through principal angles. The second part is why a disjoint dataset needs tied loadings, as R3 has.

Remark B.2 (A weak item side inflates concept shift). Item structure that $b(x)$ cannot represent is indistinguishable from a client-specific difference, and part of it is absorbed into z_k. A weak item representation therefore makes spurious concept shift *easier* to find. This is why DICES, where every item is shared, uses free loadings, which represent every item exactly, and why the RQ1 tests compare against a null fitted with the same item side.

B.2 Federated training

Algorithm 2 implements the training loop of Figure 2. With Θ fixed, step (a) is a strongly convex problem in $c + d$ variables for a logistic output with a ridge penalty, solved in a few Newton steps. With ψ_k fixed, step (b) computes the shared update at the client's own state, so the client's concept shift shapes Θ without the state being sent. The uplink per client and round is the update of $|\Theta|$ values, the curvature contributions the server aggregates into H^{t+1}, and the scalar losses used for backtracking; the client state adds nothing to it. Lemma C.6 shows what the update nevertheless carries, and the audit of §4 examines the update. Every federated arm of §4 uses this loop with the same optimiser, rounds and local steps, and differs only in which terms a client keeps (App. E.1).

The curvature estimate H^t is block-diagonal: one $(1 + d) \times (1 + d)$ block per item for free loadings, and a full matrix over the design basis on R3. The server step ρ_t is chosen by backtracking on the scalar losses the clients report for the candidate model.

B.3 Serving a new client

A client that joins after training estimates its state from m labelled ratings with Θ fixed. Two results describe how the gain depends on m.

Proposition B.3 ($1/m$ law and crossover)**.** *Let a new client's context be estimated from m ratings by $\hat{z}(S_m)$, asymptotically unbiased with covariance Σ_z/m, and let $H_z \succeq 0$ be the Hessian of the*

Algorithm 2 Federated training with a local client state

Require: rounds T; local steps E; step size κ; ridge μ; weights p_k
Ensure: shared Θ^T; client states $\{\psi_k\}$ kept locally
server initialises Θ^0 and a curvature estimate H^0; each client sets $\psi_k \leftarrow \psi^0$
for $t = 0, \dots, T-1$ **do**
  server sends Θ^t and H^t to the clients
  for all clients k **in parallel do**
    (a) personalise with Θ^t fixed: $\psi_k \leftarrow \arg\min_\psi \widehat{F}_k(\Theta^t, \psi) + \frac{\mu}{2}\|\psi - \psi^0\|^2$
    (b) shared step with ψ_k fixed: E steps $\Theta \leftarrow \Theta - \kappa\,(H^t)^{-1}\nabla_\Theta \widehat{F}_k(\Theta, \psi_k)$ from Θ^t, giving Θ_k
    (c) send $\Delta_k \leftarrow \Theta_k - \Theta^t$ ▷ ψ_k is never sent
  end for
  $\Theta^{t+1} \leftarrow \Theta^t + \rho_t \sum_k p_k \Delta_k$, with the largest $\rho_t \in \{1, \frac{1}{2}, \dots\}$ that lowers the reported losses
  update H^{t+1} from the clients' curvature at Θ^{t+1}
end for

client's risk in z at its optimum. The gain over the shared model is

$$G(m) = G_\infty - \frac{\operatorname{tr}(H_z \Sigma_z)}{2m} + o\left(\frac{1}{m}\right), \qquad m^* = \frac{\operatorname{tr}(H_z \Sigma_z)}{2G_\infty},$$

where G_∞ is the gain with the true context and m^* the support size at which the gain changes sign.

The crossover grows with the number of estimated coordinates through $\operatorname{tr}(H_z\Sigma_z)$, but also with the curvature and noise per coordinate, so it cannot be read off d alone and has to be measured. Against FedAvg the measured crossover lies at or below $m = 5$, the smallest support tested, on both datasets (App. F.5).

Proposition B.4 (Shrinkage cannot do worse than the population prior)**.** *Let contexts vary across clients as $z_k \sim \mathcal{N}(\bar{z}, \tau^2 I_d)$, and let the support estimate satisfy $\hat{z} \mid z_k \sim \mathcal{N}(z_k, (\sigma^2/m) I_d)$. The posterior mean $\tilde{z}_m = \bar{z} + \omega_m(\hat{z} - \bar{z})$, with $\omega_m = m\tau^2/(m\tau^2 + \sigma^2)$, has excess risk under the quadratic approximation*

$$\Delta R(m) = \tfrac{1}{2} \operatorname{tr}(H_z)\, \frac{\tau^2 \sigma^2}{m\tau^2 + \sigma^2},$$

which equals the risk of the population prior at $m = 0$, is non-increasing in m, and tends to zero as $m \to \infty$.

Remark B.5 (Why shrinkage is needed). An unshrunk readout ($\omega_m = 1$) can be worse than the population prior at small m, because it uses a noisy estimate at full weight. The same happens with a fixed weak ridge when a client's support labels are nearly all of one class: its level and scale diverge. Under the assumptions of Proposition B.4, shrinking each coordinate in proportion to its spread across clients removes both failures.

Algorithm 3 is the readout used for every new client in §4. It is the posterior mode under a Gaussian prior with mean and per-coordinate variance estimated from the training clients, which is the likelihood counterpart of the posterior mean of Proposition B.4.

Algorithm 3 Reading out the state of a new client (empirical Bayes)

Require: trained Θ; training states $\{\psi_k\}$ with weights p_k; support set S_m; query set Q
$\bar{\psi} \leftarrow \sum_k p_k \psi_k$; $\hat{\tau}_j^2 \leftarrow \sum_k p_k (\psi_{k,j} - \bar{\psi}_j)^2$ for each coordinate j
if $m = 0$ **then**
  $\tilde{\psi} \leftarrow \bar{\psi}$ ▷ population prior
else
  $\tilde{\psi} \leftarrow \arg\min_\psi \widehat{F}_{S_m}(\Theta, \psi) + \sum_j (\psi_j - \bar{\psi}_j)^2/(2\hat{\tau}_j^2)$ ▷ damped Newton, Θ fixed
end if
return $\eta(x) = \tilde{\beta} + \tilde{\alpha}\, g(x) + \langle \tilde{z}, W^\top b(x) \rangle$ for $x \in Q$

B.4 NULLS AND CONTROLS

Nulls. Each null breaks exactly one thing and is refitted on every draw.

- *Calibration null (spectrum test).* Labels are simulated from the fitted calibration model $\beta_0 + \beta_k + \alpha_k g_j$, which keeps every item effect, every client level and scale and the mean–variance relation of the output, and sets the concept component to zero; the calibration model is refitted on each of 500 draws and the singular values of the residual matrix are recomputed. A level-only null ($\alpha_k = 1$) is reported alongside for comparison (App. F.1).
- *Ordering null.* Labels are simulated from the same fitted calibration model, the context model with the observed d is refitted on each of 200 draws, and the ordering statistic is recomputed (App. F.2). Under the null the true ordering disagreement is zero, so the null distribution measures what estimation noise alone produces.
- *Shuffled context.* Contexts are permuted across clients with every other parameter of one trained model kept (Proposition C.4).

Control protocol. On one trained checkpoint per split, predictions are made in three modes.

- TRUE: each client uses the context estimated from its own labels.
- SHUFFLED: contexts are permuted across clients (500 permutations); the model, the level and scale of every client, and the distribution of contexts are unchanged, and only the client–context correspondence is broken.
- ZEROED: the context is set to $\mathbf{0}$ with every other parameter kept. This measures what the model is worth without a context, and checks the wiring: if TRUE and ZEROED gave identical predictions, the context would never reach the output.

A gain counts as personalisation only when TRUE beats every SHUFFLED pairing, so that the permutation p-value is at its floor, and differs from ZEROED, on every split.

C GUARANTEES: FORMAL STATEMENTS AND PROOFS

Logarithms are natural.

C.1 RQ1: IDENTIFIABILITY

Lemma C.1 (Identifiability under disjoint support)**.** *If* $\mathrm{supp}(P_k) \cap \mathrm{supp}(P_l) = \emptyset$ *for all* $k \neq l$*, concept shift is not identifiable from observed labels without structural restrictions on the hypothesis class.*

Proof. The observed data are draws of (k, x, y) with $x \sim P_k$ and $y \sim P_k(y \mid x)$, so their law depends on η_k only through its values on $\mathrm{supp}(P_k)$. Define one rule $\eta^*(x) = \eta_k(x)$ for $x \in \mathrm{supp}(P_k)$, which is well defined because the supports are disjoint. The federation in which every client labels by η^* has $\gamma_k = 0$ for all k and induces the same law of (k, x, y) as the original one. The two models are observationally equivalent, so any test of $\gamma \neq 0$ has power equal to its size, unless the hypothesis class excludes η^*, for example by tying loadings to a basis or by testing on the cells of a crossed design. □ □

When no item is labelled twice, client-specific rules and one shared rule applied to different items produce the same data (Figure 1a). On R3 we therefore test for concept shift on the cells of the crossed experimental design and tie item loadings to the design basis. Distinct photographs can still come from overlapping populations; the lemma concerns what the observed labels identify, and the design cells are what make R3 testable.

C.2 RQ2: WHEN A SMALL STATE BEATS BOTH ALTERNATIVES

Definition C.2 ((d, ε)-structured heterogeneity)**.** A federation is (d, ε)-structured if there are shared parameters Θ and client states $\psi_k \in \mathbb{R}^{c+d}$ with $\sum_k p_k[F_k(\Theta, \psi_k) - F_k(w_k^*)] \leq \varepsilon$, where w_k^* is client k's own optimum in a model class of dimension D_{loc}. Let ε_d be the smallest such ε, and $\Gamma = F(w^*) - \sum_k p_k F_k(w_k^*)$ the heterogeneity gap.

Theorem C.3 (Dominance window)**.** *Assume each client's problem is strongly convex and smooth in its own parameters, the shared parameters are estimated with vanishing error, and fitting s client parameters from m ratings costs $\chi s/m + o(1/m)$ in excess risk. Ignoring $o(1/m)$ terms, a context model of dimension d beats the shared model when $m > \chi(c+d)/(\Gamma - \varepsilon_d)$ and beats per-client models of dimension D_{loc} when $m < \chi(D_{\mathrm{loc}} - c - d)/\varepsilon_d$. The window of m in which it beats both is non-empty if and only if*

$$\frac{\varepsilon_d}{\Gamma} < 1 - \frac{c+d}{D_{\mathrm{loc}}}.$$

Proof. The excess risk of an estimator is $\sum_k p_k[F_k(\hat{\vartheta}_k) - F_k(w_k^*)]$. The shared model has excess risk tending to Γ. The context model has approximation error ε_d and estimation error $\chi(c+d)/m$. Per-client models have zero approximation error and estimation error $\chi D_{\mathrm{loc}}/m$. The context model beats the shared one when $\varepsilon_d + \chi(c+d)/m < \Gamma$, and the per-client models when $\varepsilon_d + \chi(c+d)/m < \chi D_{\mathrm{loc}}/m$. Both bounds on m are compatible if and only if $(c+d)\varepsilon_d < (D_{\mathrm{loc}} - c - d)(\Gamma - \varepsilon_d)$, that is, $D_{\mathrm{loc}}\varepsilon_d < (D_{\mathrm{loc}} - c - d)\Gamma$. □ □

The window is wide when the context explains most of the gap ($\varepsilon_d \ll \Gamma$) and the per-client model is large relative to the state ($D_{\mathrm{loc}} \gg c + d$). On DICES a per-client model has one parameter per item plus an intercept ($D_{\mathrm{loc}} = 351$) against a state of $c + d = 3$, and local and Ditto models trail the context model. On R3 a per-client model over the six design variables has $D_{\mathrm{loc}} = 7$ against $c + d = 4$, so the window is narrow, and with about 265 ratings per subject full per-client models fit best, as §4 reports.

C.3 RQ3: VALIDITY OF THE SHUFFLED-CONTEXT CONTROL

Proposition C.4 (Validity of the shuffled-context control)**.** *Fix a trained model and its contexts $\hat{z}_1, \ldots, \hat{z}_K$, and let π be a uniformly random permutation of the clients, drawn independently of the evaluation data. If the pairing of contexts with clients carries no information about the evaluation data, then $T \triangleq \hat{R}(\textsc{shuffled}) - \hat{R}(\textsc{true})$ has mean zero, and its permutation distribution gives an exact test.*

Proof. Let D_k^{ev} be client k's evaluation data. Under the null, conditional on the trained model and on the multiset $\{\hat{z}_k\}$, the evaluation data are independent of which context is attached to which client, so pairing each client with its own context is one of $K!$ equally likely pairings. Let $\hat{R}(\pi) = \sum_k p_k \hat{R}_k(\hat{z}_{\pi(k)})$. For π uniform and independent of the data, $\hat{R}(\mathrm{id})$ and $\hat{R}(\pi)$ have the same distribution, so T has mean zero. With B independent permutations, the rank of $\hat{R}(\mathrm{id})$ among $\{\hat{R}(\mathrm{id}), \hat{R}(\pi_1), \ldots, \hat{R}(\pi_B)\}$ is uniform, and $(1 + \#\{b : \hat{R}(\pi_b) \le \hat{R}(\mathrm{id})\})/(B+1)$ is an exact p-value. The model is the same in every mode and only the pairing changes, so the test is unaffected by parameter count, optimiser, splits and the marginal distribution of contexts. □ □

C.4 RQ4: WHAT LEAVES THE CLIENT

Proposition C.5 (Linkage bound)**.** *Let a released payload and a server's auxiliary profile for client k be $q_k = z_k + \xi_k$ and $a_k = z_k + \zeta_k$, with $z_k \sim \mathcal{N}(0, I_s)$ and $\xi_k, \zeta_k \sim \mathcal{N}(0, \nu^2 I_s)$ independent, so that matching coordinates have correlation $\rho = (1+\nu^2)^{-1}$. Any rule that identifies a uniformly chosen client among K from its payload and all K profiles succeeds with probability at most*

$$\frac{s\, I(\rho) + \log 2}{\log K}, \qquad I(\rho) = -\tfrac{1}{2}\log(1-\rho^2).$$

Proof. Let k be uniform and the attacker observe $O = (q, a_1, \ldots, a_K)$ with $q = q_k$. The profiles are independent of k, so $I(k; O) = I(k; q \mid a_{1:K}) = H(q \mid a_{1:K}) - H(q \mid a_{1:K}, k)$. Given k, the payload depends on the profiles only through a_k, so $H(q \mid a_{1:K}, k) = H(q \mid a_k)$. Conditioning reduces entropy and $q \sim \mathcal{N}(0, (1+\nu^2) I_s)$ whatever k is, so $H(q \mid a_{1:K}) \le H(q)$. Hence $I(k; O) \le I(q; a_k) = s\, I(\rho)$, and Fano's inequality gives $\Pr(\hat{k} \ne k) \ge 1 - (I(k;O) + \log 2)/\log K$. □ □

Here s is the number of released coordinates: $s = d$ for the context alone and $s = c + d$ for the full state. The bound grows with s, but it does not make a small state safe: $I(\rho)$ grows without limit as coordinates become reliable, so a few well-estimated numbers can identify many clients, which is what the audit of §4 finds for the state. Adding noise lowers ρ but costs every client utility. The bound concerns the released state only; the update is audited directly, because of the following lemma.

Lemma C.6 (Rank-one update)**.** *Suppose W is part of Θ, ψ_k is held fixed during the shared step, and the shared step takes plain gradient steps on W. Then the W-block of Δ_k equals $u_k z_k^\top$ for some vector u_k, so its row space is spanned by z_k whenever $u_k \neq 0$.*

Proof. The context term is $b(x)^\top W z_k$. With residual $e = \partial \ell / \partial \eta$, the gradient of client k's empirical loss in W is $\big(\sum_i e_{ki} b(x_i)\big) z_k^\top$. The residuals change between steps but z_k does not, so every step adds a matrix of the form $(\cdot)\, z_k^\top$, and their sum is $u_k z_k^\top$ with $u_k = -\kappa \sum_{\text{steps}} \sum_i e_{ki} b(x_i)$. □ □

Algorithm 2 preconditions the shared step with a curvature estimate that couples the item score and the context coordinates, so its update is not exactly rank one. The lemma is why the update is a channel worth auditing; the audit of §4 then measures the implemented update directly.

C.5 Proofs for App. B

Proposition B.1. *Invariance.* $\langle A z_k, (W A^{-1})^\top b(x) \rangle = z_k^\top A^\top A^{-\top} W^\top b(x) = \langle z_k, W^\top b(x) \rangle$, so the likelihood is unchanged. *What is identified.* Stack contexts as $Z \in \mathbb{R}^{K \times d}$ and item representations as B, one row per item. The likelihood depends on the context term only through $M = Z W^\top B^\top$. If Z, W and B have full column rank, then $\operatorname{rank} M = d$ and the row space of M is the image under B of the column space of W; since B is injective, that column space is identified. *Free loadings, one client per item.* Let $k(j)$ be the only client labelling item j. The context term enters only through $c_j = \langle z_{k(j)}, v_j \rangle$, which is unchanged by $z_k \mapsto A_k z_k$, $v_j \mapsto A_{k(j)}^{-\top} v_j$ for client-specific A_k; no common gauge removes this family, and any $\{c_j\}$ is attained with $d = 1$ ($z_k = 1$, $v_j = c_j$). □

Proposition B.3. Expanding the client's risk $R(z)$ around its minimiser z^* with Hessian H_z, the linear term vanishes, and with $\mathbb{E}\hat{z} = z^* + o(m^{-1/2})$ and $\operatorname{Cov} \hat{z} = \Sigma_z / m$, $\mathbb{E} R(\hat{z}) = R(z^*) + \operatorname{tr}(H_z \Sigma_z)/(2m) + o(1/m)$. With R_0 the shared model's risk and $G_\infty = R_0 - R(z^*)$, $G(m) = R_0 - \mathbb{E} R(\hat{z})$ gives the expansion in Proposition B.3; setting its leading terms to zero gives m^*. □

Proposition B.4. The posterior is Gaussian with mean $\tilde{z}_m$ and covariance $v_m I_d$, where $v_m = (\tau^{-2} + m\sigma^{-2})^{-1} = \tau^2 \sigma^2 / (m\tau^2 + \sigma^2)$. Under the quadratic approximation the excess risk of $\tilde{z}$ is $\frac{1}{2} \operatorname{tr}(H_z \, \mathbb{E}[(\tilde{z} - z)(\tilde{z} - z)^\top]) = \frac{1}{2} \operatorname{tr}(H_z) v_m$, which is decreasing in m, equals $\frac{1}{2} \operatorname{tr}(H_z) \tau^2$ at $m = 0$ (the risk of predicting with $\bar{z}$), and tends to zero. The unshrunk $\hat{z}$ has risk $\frac{1}{2} \operatorname{tr}(H_z) \sigma^2 / m$, which exceeds the prior's whenever $m < \sigma^2 / \tau^2$; this is Remark B.5. □

D Assumptions and scope

Lemma C.1, Remark A.1 and Propositions 3.3, C.4 and C.5 are exact under their stated assumptions. Theorem C.3 is asymptotic. Its convexity assumption concerns each client's own parameters and holds in our models: with Θ fixed, the penalised logistic loss is strongly convex in ψ_k. It does not cover the shared fit. Jointly in (W, z_k) the context term is bilinear and the objective non-convex, so a fit may stop at a local optimum whose approximation error exceeds ε_d. We therefore use the theorem for the shape of the trade-off, not for numerical predictions, and treat the learnable dimension as a property of our estimator at the data sizes studied as well as of the data. Table 3 lists each assumption and its status.

E Experimental protocol

One model family. Every experiment and every federated arm uses $\eta_{kj} = \beta_0 + \beta_k + \alpha_k g_j + \langle z_k, v_j \rangle$ with $g = Bs$ and $V = BW$ (§3, where β_0 is absorbed into g). On DICES, B is the item indicator (free loadings, 350 items). On R3, B holds the six design variables of each rated photograph: photo

Table 3: Assumptions behind the theoretical results, and how the fitted models relate to them.

Result	Assumption	Status in the fitted models
Prop. 3.3	g non-constant; $\lambda_d > \lambda_{d+1}$ for uniqueness in (c)	Population statement; the eigengap is checked empirically through subspace stability across splits.
Rem. A.1	$\alpha_k > 0$	Checked on the fitted scales; underlies the ordering test.
Lem. C.1	Disjoint supports	Exact; the observed R3 data have one subject per photograph.
Prop. B.1	Full column rank of Z, W and B	Holds generically for d below the number of clients and the item-side dimension; dimension selection keeps d small.
Thm. C.3	Strong convexity and smoothness in client parameters	Holds for the per-client logistic subproblem with fixed shared parameters; the joint bilinear fit is non-convex and solved from an SVD initialisation.
	Vanishing error in shared parameters	Approximate; shared parameters are fitted on finite data.
	Estimation error $\chi s/m$	Asymptotic; penalised fits at small m can deviate.
Prop. B.3	Asymptotically unbiased readout with covariance Σ_z/m	Approximate for the empirical-Bayes readout, which is biased towards the prior at small m.
	Independent support draws	Supports are nested across m with a fixed query set, so estimates at different m are correlated.
Prop. B.4	Gaussian prior and noise; quadratic risk	Approximate; motivates the empirical-Bayes readout of Algorithm 3, which is used for every new client.
Prop. C.4	No pairing information under the null	The permutation is applied at evaluation on one trained checkpoint, so only the pairing changes.
Prop. C.5	Gaussian coordinates and noise; independent coordinates	A modelling assumption; ρ is not estimated, so the bound is used qualitatively. The audited state includes the level and scale, whose coordinates need not be independent of the context.
Lem. C.6	Plain gradient steps on W with ψ_k fixed	Motivates the audit; the implemented step is preconditioned, so the update is audited directly.

review (± 1), standardized retention interval and its centred square, standardized scan session, and centred indicators for faces and people. The arms differ only in which terms a client keeps: shared ($\beta_0 + g_j$), level ($+\beta_k$), calibration ($+\beta_k + \alpha_k g_j$), and ours ($+$ a context of dimension d). The objective is the mean negative log-likelihood plus ridge penalties expressed as pseudo-counts on the shared parameters (1) and on the client state (chosen from $\{0.3, 1, 3\}$ on validation).

Data. Both datasets go through the same preprocessing in every experiment: on DICES the 19 raters flagged by the dataset authors are removed and `Unsure` is mapped to safe; on R3 the target is $\mathbf{1}[\mathrm{Mem}_s \geq 5]$ and the design cells used for the RQ1 tests are photo review $\times$ interval bin $\times$ faces $\times$ people, kept when at least 30 subjects occupy them.

Splits and selection. Five random splits are frozen to disk and shared by every experiment; federated runs use the first three. Every split divides the data $60/20/20$ into training, validation and test sets: training data fit the model, validation data choose the ridge strength and the context dimension, and test data give every reported number. *Seen items*: each client's own ratings are divided $60/20/20$, so each test rating is one the client held out, on an item that other clients may have labelled in training. *Items unseen by the client* (DICES): the 350 items are divided $60/20/20$, so the client never labelled its test items in training; a held-out item's parameters are fitted on half of the clients and scored on the other half, then the halves swap, so no rating enters its own prediction. *New clients*: 30 DICES raters or 8 R3 subjects are held out whole per split, with nested support sets of their first m ratings and a fixed query set. The context dimension $d^\star$ is the smallest d whose validation NLL exceeds the best one by less than one standard error of the *paired* split-by-split difference. Pairing matters: splits differ in difficulty (different items are held out), and an unpaired standard error counts that variation as noise.

Metrics. Every reported number is computed on test data. *NLL* is the mean negative log-likelihood per rating in bits (the natural-log value divided by $\ln 2$), averaged over all test ratings; lower is better, and a *gain* is the reduction in test NLL against the calibration model of the same split, so a positive gain favours the model being compared. *Accuracy* and *macro-F1* threshold the predicted probability at 0.5; macro-F1 averages the F1 scores of the two classes, so the minority class counts as much as the majority class. *AUC* is the area under the ROC curve of the predicted probabilities. *Harmed* is the share of clients whose test NLL is higher than under FedAvg, and *worst decile* is the mean test

NLL of the tenth of clients with the highest test NLL. Means and standard deviations are taken across splits. Statistics used in a single analysis (the ordering discordance, principal-angle cosines, top-1 linkage, balanced accuracy and the minimum detectable effect(MDE)) are defined where they are reported (App. F.2, F.3 and F.6).

New clients. A new client's state is fitted from its m support labels with the shared model frozen, by damped Newton steps on the penalized likelihood. The prior is empirical Bayes (App. B.3): its mean is the training clients' label-weighted mean state, and coordinate i is penalized with strength $1/(2\tau_i^2)$, where τ_i^2 is the spread of that coordinate across the training clients. Coordinates that barely vary across clients, such as the scale, are shrunk hard; the level and the context are left freer. A fixed weak ridge instead lets a client whose support labels are almost all positive reach extreme states; on DICES this raised the query NLL of two held-out raters to about 20 bits.

E.1 Federated setup

Protocol. Every arm runs 100 rounds with all clients participating and aggregation weights $p_k = n_k / \sum_l n_l$. In each round a client (i) fits its state with the shared parameters fixed (Newton), (ii) takes two local steps on the shared parameters, preconditioned by the server's curvature estimate from the previous round (per-item blocks for free loadings, a full matrix for the six-dimensional design basis), and (iii) returns the update, with the scalar losses and curvature contributions the server step uses. The server averages the updates and, using the scalar losses the clients report for the candidate model, keeps the largest step that lowers the federated objective (backtracking). The client state is never sent. The same optimizer, rounds and local steps are used by every arm.

Baselines. All baselines are implemented on the same model family, so they differ from ours only in what a client keeps.

- *FedAvg*: shared model only, trained with the same preconditioned optimizer as every arm, so it differs from standard FedAvg (McMahan et al., 2017) in the optimizer, not in the objective.
- *Client embedding*, in the style of Deng et al. (2023): a free client vector acting through the shared item loadings, with a level but without the calibration split; dimension from $\{2, 4, 8\}$.
- *FedRep* (Collins et al., 2021): a personal linear head and level on a shared representation, without the shared score; head dimension from $\{1, 2, d^\star+1, 4\}$.
- *IFCA* (Ghosh et al., 2020): $C \in \{2, 3\}$ shared models; each client trains and is served by the one with the lowest loss on its own data.
- *Ditto* (Li et al., 2021): a personal model minimizing the client's loss plus $\lambda\|\theta_k - \theta_{\text{global}}\|^2$ at the final FedAvg model, $\lambda \in \{1, 10, 100\}$.

Hyper-parameters are chosen per split on validation NLL from grids of comparable size; new clients reuse the seen-items choice. Every arm passes a gate before it is reported: test NLL below the base rate and accuracy at least that of the majority class. All arms pass. As a convergence check, the federated shared, calibration and context models are compared with the central optimum of the same objective (Table 15).

F Full results

F.1 RQ1: disagreement beyond calibration

Test. For a fitted null model we form the client $\times$ item matrix of residuals $y - \hat{p}$ (on R3, summed over the ratings in a design cell and divided by the square root of their number) and compute its singular values. Labels are simulated from the fitted null, the null model is *refitted* on every draw, and the statistic is recomputed (500 draws, so $p \geq 1/501 \approx 0.002$). The level-only null keeps an item effect and a client level; the calibration null adds the client scale. d_{spec} counts leading components above their null 95th percentile.

Calibration of the test. Synthetic datasets drawn from the fitted calibration model (*scale only*), or from it with an order-preserving bend of the shared score (*monotone bend*), are tested with 100 null

Table 4: Interaction spectrum against a level-only and a calibration null, refitted on every draw. The ratio is the observed singular value over its null mean.

Dataset	Null	Comp.	Obs./null	z	p	d_{spec}
DICES-350	level only	σ_1	1.57×	41.9	$\leq$0.002	10
DICES-350	level only	σ_2	1.26×	24.5	$\leq$0.002	
DICES-350	calibration	σ_1	1.50×	38.6	$\leq$0.002	9
DICES-350	calibration	σ_2	1.18×	16.1	$\leq$0.002	
R3	level only	σ_1	1.70×	17.3	$\leq$0.002	19
R3	level only	σ_2	1.65×	18.6	$\leq$0.002	
R3	calibration	σ_1	1.70×	15.8	$\leq$0.002	17
R3	calibration	σ_2	1.59×	16.9	$\leq$0.002	

draws each (Table 5). With ten replicates per world the rejection rates are coarse (at most 2 of 10), against observed calibration-null z-scores of 38.6 (DICES) and 15.8 (R3).

Table 5: Rejection rate of the calibration-null test at level 0.05 on synthetic data with no disagreement beyond calibration.

Dataset	Synthetic world	Datasets	Rejection rate
DICES-350	scale only	10	0.2
DICES-350	monotone bend	10	0.0
R3	scale only	10	0.0
R3	monotone bend	10	0.1

Reading the R3 spectrum. On R3 the first three components exceed their null mean by 70%, 59% and 31%; from the fourth on, the spectrum stays 10–20% above the null out to the twentieth component, so d_{spec} counts nearly every component. A plateau of this kind is what photographs that differ in content within a design cell would produce, since the binomial null treats the ratings in a cell as exchangeable. We therefore interpret the leading R3 components and do not report a detectable dimension for R3 in the main text.

F.2 RQ1: ORDERING DIFFERENCES

Statistic. For the fitted context model, $D = \binom{K}{2}^{-1} \sum_{k<l} \overline{\max\{0, -\Delta_k \Delta_l\}}$, where $\Delta_k = \eta_{ka} - \eta_{kb}$ and the bar averages over 3,000 random item pairs: pairs of the 350 DICES items, and pairs among 400 sampled R3 photographs, which every subject can be scored on through the design basis. Under calibration with positive scales $D = 0$ exactly. The null simulates from the fitted calibration model and refits the context model with the same d (200 draws). We use the magnitude rather than the count of disagreements, because estimation noise creates many disagreements of negligible size; the count is reported alongside (Table 6, last column).

Table 6: Ordering discordance of the fitted clients' rules against the calibration null, on the data and on two synthetic worlds without ordering differences.

Dataset	World	Discordance	Null mean	Null 95th	p	Pairs ordered diff.
DICES-350	observed	0.173	0.053	0.082	0.005	15.2%
DICES-350	scale only	0.054	0.050	0.069	0.244	10.7%
DICES-350	monotone bend	0.034	0.049	0.069	0.976	8.7%
R3	observed	0.184	0.043	0.064	0.005	30.6%
R3	scale only	0.059	0.052	0.073	0.268	26.6%
R3	monotone bend	0.021	0.032	0.045	0.902	14.6%

R3 forgetting slopes. A per-subject logistic regression of the binary target on the standardized retention interval (ridge 1 on the slope) gives slopes with Wald intervals. The mean slope is -0.26; 14 of 40 intervals lie below zero (12 after Benjamini–Hochberg correction) and 2 above zero (both after correction). Eight point estimates are positive, but only two are distinguishable from zero, so we report two.

Table 7: Dimension ladders, five splits. Gains are test NLL relative to $d = 0$ (the calibration model) with standard errors across splits; bold: selected $d^\star$.

Dataset	Protocol	d	Val. NLL	Test NLL	Gain $\pm$ SE	+ splits	AUC
DICES-350	seen items	0	0.6438	0.6437	—	0/5	0.846
DICES-350	seen items	**1**	0.6313	0.6307	$+0.0131 \pm 0.0009$	5/5	0.854
DICES-350	seen items	2	0.6438	0.6477	-0.0040 ± 0.0018	0/5	0.849
DICES-350	seen items	3	0.6676	0.6683	-0.0245 ± 0.0044	0/5	0.843
DICES-350	seen items	4	0.6933	0.6899	-0.0462 ± 0.0056	0/5	0.838
DICES-350	seen items	5	0.7292	0.7279	-0.0842 ± 0.0052	0/5	0.829
DICES-350	seen items	6	0.7632	0.7671	-0.1234 ± 0.0028	0/5	0.821
DICES-350	unseen by client	0	0.6463	0.6408	—	0/5	0.844
DICES-350	unseen by client	**1**	0.6327	0.6255	$+0.0153 \pm 0.0026$	5/5	0.853
DICES-350	unseen by client	2	0.6373	0.6303	$+0.0105 \pm 0.0031$	5/5	0.852
DICES-350	unseen by client	3	0.6434	0.6387	$+0.0021 \pm 0.0038$	3/5	0.850
DICES-350	unseen by client	4	0.6488	0.6496	-0.0088 ± 0.0043	1/5	0.846
DICES-350	unseen by client	5	0.6605	0.6561	-0.0153 ± 0.0040	0/5	0.845
DICES-350	unseen by client	6	0.6697	0.6666	-0.0258 ± 0.0041	0/5	0.842
R3	seen items	0	0.8779	0.8824	—	0/5	0.724
R3	seen items	1	0.8662	0.8774	$+0.0049 \pm 0.0012$	5/5	0.730
R3	seen items	**2**	0.8540	0.8619	$+0.0204 \pm 0.0026$	5/5	0.744
R3	seen items	3	0.8542	0.8604	$+0.0219 \pm 0.0022$	5/5	0.746
R3	seen items	4	0.8576	0.8641	$+0.0182 \pm 0.0018$	5/5	0.744

F.3 RQ2: DIMENSION, STABILITY AND LABEL BUDGET

On items unseen by the client, $d = 2$ still gains (0.0105 bits on all five splits) but less than $d = 1$, and the paired rule selects $d = 1$. On R3, $d = 3$ gains slightly more on test (0.0219 against 0.0204) but not on validation, where $d = 2$ is best.

Table 8: Stability of the selected context subspace: mean cosine of the principal angles between the context of split 0 and of each other split, against random subspaces of the same size.

Dataset	$d^\star$	Mean cosine vs split 0	Range	Random (95th pct.)
DICES-350	1	0.931	0.927–0.933	0.084 (0.203)
R3	2	0.876	0.855–0.896	0.187 (0.309)

Table 9: Label budget: the dimension selected when each client keeps a fraction of its training ratings (validation and test sets unchanged), and its test gain.

Labels per client	25%	50%	75%	100%
DICES-350: $d^\star$	0	0	1	1
gain at $d^\star$ (bits)	—	—	+0.0069	+0.0131
R3: $d^\star$	0	2	2	2
gain at $d^\star$ (bits)	—	+0.0124	+0.0186	+0.0204

With a quarter of the labels, a one-dimensional DICES context loses 0.061 bits against calibration, and at half it still loses 0.010; the direction becomes useful between half and three quarters of the labels.

F.4 RQ3: PERSONALIZATION CONTROLS, NEW CLIENTS AND DEMOGRAPHICS

On R3 one of ten permutations gains 0.005 bits, less than a quarter of the true gain; the mean permuted gain is negative on both datasets.

At $m = 0$ each readout returns its own model's mean training state: the calibration readout uses the calibration model ($d = 0$), the other rows the context model, whose shared score is fitted jointly with the contexts and so predicts slightly worse at the average client, as the zeroed context does in Table 10.

F.5 FEDERATED COMPARISON

Context over calibration, per split. On one federated checkpoint per split, the true contexts gain over the federated calibration model on every split (0.0165 bits on DICES and 0.0145 on R3 on

Table 10: Controls on one trained checkpoint per split (mean over five splits; gains in bits over the calibration model of the same split). p is the share of 500 context permutations doing at least as well as the true pairing.

Dataset	True	Zeroed	Shuffled (mean)	Best of 500 shuffles	p
DICES-350	+0.0131	−0.0046	−0.0467	−0.0228	$\leq$0.002
R3	+0.0204	−0.0061	−0.0595	−0.0157	$\leq$0.002

Table 11: Label permutation: each client's training labels are permuted across its items and both the calibration and the context model are refitted; gain of the context over calibration (bits, split 0).

Dataset	True labels	Permuted (mean)	Permuted (max)	Permutations
DICES-350	+0.0135	−0.0318	−0.0262	10
R3	+0.0226	−0.0098	+0.0054	10

average), while contexts permuted across clients lose (-0.026 and -0.037) and zeroed contexts lose slightly (-0.002 and -0.014); $p \leq 0.002$ on every split.

F.6 RQ4: CONTEXTUAL PRIVACY AUDIT

Threat model and payloads. An honest-but-curious server tries to link two pseudonymous payloads of the same client; it does not recover a real-world identity. Each client's training ratings are split at random into two halves. The *state* payload is the client's (β_k, α_k, z_k) fitted on a half with the trained shared model fixed; the *update* payload is the client's full local update of the trained federated model computed on a half (701 coordinates on DICES, 19 on R3); *both* concatenates the two after standardizing each block. Matchers: cosine on raw coordinates, cosine on standardized coordinates, and Euclidean distance on standardized coordinates, which also uses magnitude. Linkage is scored in both directions against a permutation null (2,000 permutations).

On DICES a client's update is computed on the items of one half, and with free item loadings it changes only those items, so a client's two half-updates share almost no coordinates; this is why update linkage sits at or below chance there, not because the update carries no information. On R3 the loadings are tied to the design basis, so both half-updates change the same coordinates. The state out-links the update under every matcher on both datasets.

One of 30 tests exceeds its MDE (R3 age from the combined payload), and four more exceed the permutation threshold but not the MDE, all of them involving the update. No test on the state alone exceeds either.

AI USE STATEMENT

Large Language Models (LLMs) were used exclusively to polish the writing tone, improve clarity, and assist formatting.

ETHICS STATEMENT

Data & Consent. This work uses two existing datasets (DICES-350 and R3) under their licenses and data-use agreements. We collected no new human data. For R3, we rely solely on released design variables. **Demographics & Privacy.** Rater metadata is analyzed strictly in aggregate to evaluate if demographics explain preferences; we strongly caution against using these attributes for individual profiling. Our privacy audits (linkage and attribute-inference attacks) are standard diagnostic tools, used here only to motivate our privacy-preserving choices: keeping the client state local and using secure aggregation.

REPRODUCIBILITY STATEMENT

Complete formal proofs, methodology details, data preprocessing steps, and hyperparameter search grids are provided in the Appendices. To ensure strict comparability, all experiments use standardized data splits frozen to disk. Both datasets are accessible under their original release terms.

Table 12: New DICES raters: query NLL (bits) by the number m of support labels, central model, three splits. All readouts use the empirical-Bayes prior of App. E.

Readout	m=0	m=5	m=10	m=25	m=50	m=100
own context (readout)	0.785	0.708	0.666	0.635	0.620	0.613
calibration readout	0.777	0.704	0.674	0.650	0.638	0.631
another client's context	0.785	0.836	0.865	0.916	0.924	0.950
population prior	0.785	0.785	0.785	0.785	0.785	0.785
metadata prior	0.803	0.803	0.803	0.803	0.803	0.803

Table 13: New R3 subjects: query NLL (bits) by m, central model, three splits.

Readout	m=0	m=5	m=10	m=25	m=50	m=100	m=200
own context (readout)	1.002	0.952	0.926	0.905	0.893	0.879	0.860
calibration readout	0.995	0.950	0.934	0.926	0.917	0.914	0.907
another client's context	1.002	0.972	0.974	1.000	1.026	1.038	1.044
population prior	1.002	1.002	1.002	1.002	1.002	1.002	1.002
metadata prior	1.022	1.022	1.022	1.022	1.022	1.022	1.022

Table 14: Association of the context with client metadata: R^2 of a regression of the context on the metadata (categorical attributes as indicators, numeric ones standardized) against 2,000 permutations of clients.

Dataset	R^2	Null mean	Null 95th	p	Predictors / clients
DICES-350	0.105	0.088	0.160	0.29	9 / 104
R3	0.142	0.205	0.331	0.82	8 / 40

Table 15: Seen items: AUC, share of clients worse off than under FedAvg, and mean NLL of the worst decile of clients; bottom: central optimum of the same objective (convergence check). Means of NLL, accuracy and macro-F1 are in Table 1.

	DICES-350			**R3**		
Method	AUC	Harmed	Worst dec.	AUC	Harmed	Worst dec.
FedAvg	0.740	—	1.362	0.620	—	1.055
+ level	0.838	13.8%	0.985	0.720	22.5%	1.075
+ level + scale	0.844	10.3%	0.970	0.722	22.5%	1.055
Ours (+ context)	0.853	10.3%	0.959	0.736	13.3%	1.040
Client embedding	0.854	9.6%	0.944	0.733	12.5%	1.054
FedRep	0.853	10.3%	0.953	0.735	15.0%	1.048
IFCA	0.810	35.3%	1.091	0.712	25.8%	1.089
Ditto	0.838	15.1%	0.977	0.744	14.2%	1.030
central shared		NLL 0.789			NLL 0.968	
central calibration		NLL 0.645			NLL 0.885	
central context		NLL 0.628			NLL 0.861	

Table 16: New DICES raters, federated: query NLL (bits) by m, three splits.

Method	m=0	m=5	m=10	m=25	m=50	m=100
FedAvg	0.773	0.773	0.773	0.773	0.773	0.773
+ level	0.769	0.706	0.678	0.659	0.649	0.645
+ level + scale	0.773	0.703	0.674	0.650	0.639	0.631
Ours (+ context)	0.779	0.701	0.664	0.637	0.621	0.615
Client embedding	0.774	0.701	0.665	0.636	0.620	0.613
FedRep	0.776	0.700	0.664	0.637	0.621	0.613
IFCA	0.793	0.828	0.735	0.707	0.693	0.690
Ditto	0.773	0.724	0.697	0.674	0.662	0.656
Ours, *shuffled* state	0.779	0.840	0.870	0.895	0.912	0.934

Table 17: New R3 subjects, federated: query NLL (bits) by m, three splits.

Method	m=0	m=5	m=10	m=25	m=50	m=100	m=200
FedAvg	0.991	0.991	0.991	0.991	0.991	0.991	0.991
+ level	0.991	0.947	0.934	0.928	0.919	0.916	0.912
+ level + scale	0.995	0.948	0.933	0.925	0.915	0.912	0.906
Ours (+ context)	0.994	0.947	0.931	0.925	0.913	0.906	0.892
Client embedding	0.997	0.949	0.933	0.926	0.910	0.905	0.894
FedRep	0.997	0.951	0.935	0.924	0.910	0.904	0.891
IFCA	1.062	0.997	0.960	0.984	0.958	0.948	0.943
Ditto	0.991	0.985	0.989	0.922	0.905	0.883	0.860
Ours, *shuffled* state	0.994	1.035	1.045	1.040	1.023	1.022	1.031

Table 18: Top-1 linkage (% of clients). *: above the permutation null at $p < 0.05$.

Dataset	Payload	Coords	Cosine	Cosine (std.)	Euclidean (std.)	Null 95th	Chance
DICES-350	state	3	13.0*	12.0*	17.3*	2.9	1.0
DICES-350	update	701	0.0	0.0	0.5	2.4	1.0
DICES-350	both	704	6.2*	0.0	0.5	2.4	1.0
R3	state	4	18.8*	32.5*	28.7*	6.2	2.5
R3	update	19	12.5*	11.2*	5.0	6.2	2.5
R3	both	23	30.0*	17.5*	13.8*	6.2	2.5

Table 19: Attribute inference from each payload (balanced accuracy, cross-validated). Threshold: 95th percentile under label permutation; MDE: smallest accuracy detected with 80% power. Bold: exceeds the MDE; †: exceeds only the threshold. R3 attributes keep classes with at least three subjects.

Dataset	Attribute	Payload	Bal. acc.	Chance	Threshold	MDE
DICES-350	gender	state	0.420	0.500	0.579	0.625
DICES-350	gender	update	0.433	0.500	0.585	0.637
DICES-350	gender	both	0.490	0.500	0.609	0.662
DICES-350	race	state	0.239	0.200	0.263	0.320
DICES-350	race	update	0.277†	0.200	0.271	0.320
DICES-350	race	both	0.228	0.200	0.263	0.320
DICES-350	age	state	0.333	0.333	0.407	0.450
DICES-350	age	update	0.390	0.333	0.410	0.467
DICES-350	age	both	0.371	0.333	0.414	0.467
DICES-350	education	state	0.337	0.333	0.369	0.433
DICES-350	education	update	0.352	0.333	0.389	0.450
DICES-350	education	both	0.378	0.333	0.383	0.450
R3	Gender	state	0.483	0.500	0.574	0.650
R3	Gender	update	0.665†	0.500	0.630	0.713
R3	Gender	both	0.658†	0.500	0.641	0.713
R3	Age	state	0.283	0.250	0.417	0.550
R3	Age	update	0.467†	0.250	0.417	0.569
R3	Age	both	**0.733**	0.250	0.450	0.569
R3	Handness	state	0.500	0.500	0.500	0.637
R3	Handness	update	0.459	0.500	0.640	0.750
R3	Handness	both	0.459	0.500	0.653	0.738
R3	Race	state	0.500	0.500	0.500	0.625
R3	Race	update	0.500	0.500	0.625	0.738
R3	Race	both	0.500	0.500	0.625	0.738
R3	Education	state	0.483	0.500	0.575	0.662
R3	Education	update	0.512	0.500	0.626	0.713
R3	Education	both	0.604	0.500	0.671	0.750
R3	Language	state	0.617	0.500	0.642	0.713
R3	Language	update	0.489	0.500	0.667	0.738
R3	Language	both	0.539	0.500	0.641	0.700

Table 20: Per-release Gaussian differential privacy on the state ($\delta = 10^{-5}$, clipping at the 95th percentile of state norms, split 0): utility is the test gain of the noisy state over the same model with the context set to zero, and linkage is computed on the defended payload.

	DICES-350		**R3**	
ϵ	Utility (bits)	Linked	Utility (bits)	Linked
none	+0.0185	17.3%	+0.0380	28.7%
10.0	−0.7820	2.9%	−1.6835	1.2%
5.0	−2.1137	1.0%	−3.6974	1.2%
1.0	−8.5164	0.0%	−8.6964	1.2%