WHEN FRAGMENTS RECUR: COMPOSITIONAL RECALL UNDER DISTRIBUTED CONCEPT DRIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

In long-running federated systems, clients may revise different parts of their class-to-label mappings at different times, for example when hospitals adopt updated coding conventions for different disease groups or moderation systems update policies for different content categories. A client can then face a complete labelling function that has never appeared before, although each of its local components has already been learned by other clients or in earlier rounds. Existing methods for recurring concept drift store and retrieve complete models, so this distributed historical knowledge cannot be recombined. We call this setting *fragment-recurrent concept drift*. On constructed CIFAR-100, GTSRB and 20 Newsgroups benchmarks with frozen encoders and current labels available before prediction, whole-head memory trails a memory that assembles rows by class state by 23–32 percentage points at switches to unseen combinations, even with true identities given. We propose FragMem, a server-side compositional memory with historical state slots shared across clients and rounds: current class-conditional feature statistics select one slot per label, the selected classifier rows form the client's model, and the server aggregates locally trained updates by label and slot, so clients with different concepts improve shared class-level knowledge. FragMem comes within 0.3 points of the class-state oracle; its advantage over whole-concept memory grows with the demand for unseen combinations and shrinks when states are hard to identify or current-batch adaptation suffices. The results support persistent federated memory that recombines knowledge from recurring concept fragments.

1 INTRODUCTION

Federated systems often span organisations whose labelling practices differ and evolve independently (Plank, 2022; Wei et al., 2022). Consider hospitals that jointly train a clinical coding model: each record's underlying condition is filed under the hospital's current diagnosis code, coding conventions are revised over time, and a revision can affect one group of conditions at a time. During staged or institution-specific roll-outs, a hospital may combine a mapping for cardiovascular diagnoses that another hospital used earlier with a mapping for endocrine diagnoses that a third hospital used at another time. Its hospital-wide labelling function has then never appeared before, although each local component has already been learned somewhere in the federation. A similar pattern can arise in distributed content moderation, where policies for categories such as harmful content, nudity or misinformation are revised separately.

This creates a distinctive reuse problem: the knowledge a client needs already exists, but it is spread over complete models learned by different clients at different times. We call the setting *fragment-recurrent concept drift*: a *fragment* is a class-to-label mapping for a group of classes, the complete mapping is a *concept*, and fragments recur across clients and rounds in new combinations, so BK fragments form K^B concepts (Figure 1a). Only the mapping changes, so this is real concept drift in $P(Y \mid X)$ (Chen et al., 2024) arising at different clients and times (Jothimurugesan et al., 2023). Our experiments construct such recurrence rather than document deployed systems. *How should federated memory organise knowledge learned across clients and rounds for reuse in new combinations?*

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.

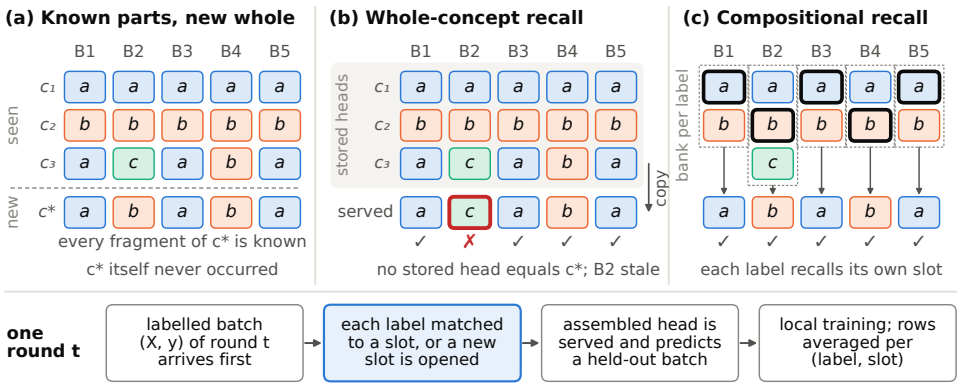

Figure 1: **Reusing familiar fragments in a new combination.** (a) Each block of labels (B1–B5) follows a fragment (a, b, c). The client's new concept c^* combines fragments already seen in the federation. (b) Whole-concept memory has no head for c^*. The concept-identity reference copies the client's previous head, which is stale on B2 (CIFAR-100, H2: 39.8%). (c) FragMem retains shared historical slots per label (drawn per block) and retrieves their states independently (63.1%; Table 1). Bottom: the labelled batch selects rows before prediction on a held-out batch.

Existing federated methods address two parts of this problem separately. Concept-level methods retain complete models across rounds (Ghosh et al., 2020; Jothimurugesan et al., 2023; Guo et al., 2024; Fenoglio et al., 2025), but recalling one stored model cannot combine components from different historical models. Class-level aggregation, exemplified by FedCCFA (Chen et al., 2024), combines classifier rows across clients within a round, without retaining a bank of historical states per label. The missing capability is to accumulate class-level knowledge across clients *and* rounds, then retrieve it independently of the complete concepts in which it was learned.

We compare two memories given the true identities, with frozen encoders and current labels available before prediction. One stores complete heads by concept; the other stores rows by class state. At switches to unseen combinations on CIFAR-100, GTSRB and 20 Newsgroups features, the whole-head reference trails by 23–32 percentage points. Removing identity errors therefore leaves a substantial gap. This reference initialises a new concept from the client's previous head; it does not select the best stored head or bound all adaptation methods (Section 5.2).

FragMem provides this persistent server-side memory through two modules (Figure 1c). *Class-state memory addressing* uses current per-label feature statistics to select historical slots, creating a slot when no state matches. The descriptors follow class-wise model specifications (Wu et al., 2023; Liu et al., 2025; Zhang et al., 2023). *Compositional recall and consolidation* assembles the selected classifier rows, then aggregates local updates by label and slot. Clients with different complete concepts can improve the same stored row, which remains available in later rounds.

FragMem comes within 0.3 points of the class-identity oracle on the recombination benchmark, as does per-class loss matching over the same memory; we further test where limited recombination, identification errors and strong current-batch adaptation reduce the gains. Our contributions are:

- **A setting for compositional reuse in FL.** A controlled generator varies which blocks change, how often their combination is new and whether other groups recently held the target fragments (Sections 3 and 5).
- **A persistent federated memory.** FragMem combines per-label retrieval with aggregation into shared state slots that accumulate knowledge across clients and rounds (Section 4).
- **Evidence on when composition helps.** Identity-given comparisons and memory controls connect the gains to recall granularity and reuse demand; further experiments identify where these gains shrink or reverse (Section 5).

2 RELATED WORK

Concept-level memory in FL. Clustered and multi-model FL assigns clients to one of several models (IFCA, CFL, FeSEM, FedEM, FedRC; Ghosh et al., 2020; Sattler et al., 2021; Long et al., 2023; Marfoq et al., 2021; Guo et al., 2024). Drift methods detect changes and create, merge or re-weight models (FedDrift, Flash, Adaptive-FedAvg, FLUX, Fielding, FedDAA, FedDCA; Jothimurugesan et al., 2023; Panchal et al., 2023; Canonaco et al., 2021; Fenoglio et al., 2025; Li et al., 2026a; Peng & Tang, 2025; Fang et al., 2026); FEROMA weights models by data-profile similarity (Li et al., 2026b). These approaches organise reuse around complete models per cluster, concept or client; we retrieve local states separately from the complete historical models that contain them.

Class-level aggregation. FedCCFA (Chen et al., 2024) clusters and averages classifier rows among the current round's clients, without a bank of historical states per label. Class-wise federated averaging (Lee & Choi, 2025) targets label heterogeneity. Personalised FL adapts models per client (Tan et al., 2023; Li et al., 2021; Deng et al., 2020; Oh et al., 2022; Tan et al., 2022), including closed-form classifiers from class statistics (Fanì et al., 2024). FragMem retains shared label-state slots after their contributing clients move to other concepts.

Recurring concepts and composition. Stream learning reuses classifiers for recurring contexts (Katakis et al., 2010; Gama & Kosina, 2014; Gama et al., 2014; Lu et al., 2019), fingerprints concepts (Halstead et al., 2021), and studies partial drift and recurrence (Webb et al., 2016; Aguiar & Cano, 2024; Gunasekara et al., 2024). Continual learners infer tasks (Wortsman et al., 2020) and compose modules (Mendez & Eaton, 2021), including in FL (Yoon et al., 2021; Dong et al., 2022). We study the reuse of recurring concept fragments across clients and rounds; our construction changes $P(Y \mid X)$ with $P(X)$ fixed, unlike label shift, which changes $P(Y)$ with $P(X \mid Y)$ fixed (Lipton et al., 2018).

Model specifications. Learnware and RKME specifications identify reusable models from data statistics under one shared labelling function (Zhou, 2016; Zhou & Tan, 2024; Wu et al., 2023); class-wise specifications match models across label spaces (Guo et al., 2023; Liu et al., 2025), model labels select small models per label (Zhang et al., 2025), and Model Spider uses per-class feature centres (Zhang et al., 2023). FragMem's fingerprint follows this line; our contribution is to use such descriptors to address a persistent federated memory, consolidate updates by label and state, and evaluate compositional recall under drift.

3 PROBLEM FORMULATION

3.1 FEDERATED CONCEPT DRIFT

A frozen encoder ϕ maps inputs to $\mathbb{R}^d$; a sample of true class $z \in \mathcal{Z}$ receives label $y = g_{k,t}(z)$ under client k's *class-to-label mapping* $g_{k,t} : \mathcal{Z} \to \mathcal{Y}$ at round t, $|\mathcal{Y}| = C$; heads are trained on ϕ. Class proportions $P_k(z)$ and features $P(\phi(x) \mid z)$ are fixed, so varying $g_{k,t}$ changes $P_{k,t}(Y \mid X)$ with $P_k(X)$ fixed (Chen et al., 2024; Webb et al., 2016). A *concept* is a complete mapping g; the *state* of label c is the set of true classes it carries, $s_c = g^{-1}(c)$.

3.2 FRAGMENT RECURRENCE

The classes are partitioned into B blocks. Within block b the mapping is one of K *fragments*, so a concept is a tuple $(f_1, \ldots, f_B)$ of fragments and $g_{k,t}$ is the concept client k holds at round t. A concept is *seen* once some client has held it as a whole; a switch is *unseen* if its target concept is not seen although each of its fragments has been held before. *Fragment-recurrent concept drift* is drift in which fragments recur in new combinations. In a federation the fragments a client needs may have been learned by different clients at different rounds, so neither its own history nor its current-round peers need hold them together. Unlike recurring concept drift, the whole concept need not return; unlike partial drift, the changed part may already be known.

3.3 LIMITS OF WHOLE-CONCEPT RECALL

Let $S \subseteq \mathcal{G}$ be the concepts seen so far, with $\mathcal{G}$ the valid mappings (one label per true class). A concept-level memory stores one whole head per $s \in S$; a class-level memory stores one row per (label, state) seen. Which concepts can each serve with rows trained for the current states?

Proposition 1 (Recall-set separation)**.** *A learner that serves one stored whole head can serve a head trained for the current concept s^* only if $s^* \in \mathcal{R}_{\text{whole}} = S$. A learner that assembles one stored row per label can, under correct per-label identification, do so for every $s^* \in \mathcal{R}_{\text{comp}} = \mathcal{G} \cap \prod_{c=1}^{C} \{s_c : s \in S\}$. Here $S \subseteq \mathcal{R}_{\text{comp}}$, strictly whenever a valid mapping combines states of different seen concepts, and $\mathcal{R}_{\text{comp}}$ can contain K^B concepts while $|S| = K$.*

The proof is immediate (Appendix A). For $s^* \notin S$ a whole-head learner has no correct head to identify: every stored head is stale on at least one label, and better concept identification cannot change that, whereas a class-level memory reaches s^* once its fragments have occurred. The proposition concerns which heads are *recallable*, not their accuracy: rows learned in different combinations are only approximately compatible, and strategies that retrain or assemble heads from the current batch are not restricted (Appendix A gives an accuracy form under a row-compatibility assumption). This separation motivates a memory that keeps reusable states below the whole-concept level and recomposes them for each client.

3.4 EVALUATION PROTOCOL

In each round (Figure 1, bottom) client k first receives a labelled batch (X_t, y_t), which a method may use to decide which model to serve. The served model then predicts a held-out evaluation batch, and only then does the client train on (X_t, y_t). Recall is thus not zero-shot, but apart from fine-tuning baselines and per-round refits the served rows were trained on earlier rounds only.

4 FRAGMEM: STATE-INDEXED FEDERATED MEMORY

FragMem maintains a server-side memory shared across clients and rounds. For each label c, a bank $\mathcal{M}_c = \{(\xi_{c,j}, r_{c,j})\}_{j=1}^{J_c}$ pairs state descriptors $\xi_{c,j}$ with classifier rows $r_{c,j} = (w_{c,j}, b_{c,j})$. Clients with different complete concepts can share slots for matching label states. *Class-state memory addressing* selects each client's slots; *compositional recall and consolidation* assembles its model and aggregates updates into the shared memory (Figure 2).

4.1 CLASS-STATE MEMORY ADDRESSING

Several states per label. A label's state, $s_c = g^{-1}(c)$, changes with the class-to-label mapping. The server retains several slots per label without knowing the class blocks. Each descriptor $\xi_{c,j}$, or *fingerprint*, contains the feature sum, count $N_{c,j}$ and within-label sum of squares accumulated from assigned clients and rounds; $M_{c,j}$ is their mean.

Matching a state. Class-conditional feature statistics indicate which state a label currently represents. Client k uploads the mean m_c, count n_c and within-label sum of squares for each label in its current batch ($C(d{+}2)$ floats), plus split-half statistics. For every slot the server computes

$$z_c(j) = \frac{\|m_c - M_{c,j}\|^2}{v_c\,(1/n_c + 1/N_{c,j})}, \tag{1}$$

where v_c is the within-label variance trace. Under one state z_c is roughly a scaled χ^2 with mean near 1; label c takes the slot $a_{k,c}^t$ of smallest z_c.

Creating a slot. If the minimum exceeds a threshold τ, the server creates a slot initialised from the nearest slot's row. To calibrate τ without true state identities, each client randomly splits its samples of each label into two halves. Both halves share a state, so their values of Equation (1) provide a null sample. We fit a scaled χ^2 by matching moments (Satterthwaite, 1946) and use its $1 - \alpha/(CT)$ quantile, with $\alpha = 0.05$. This targets a nominal spurious-slot rate of about α per client

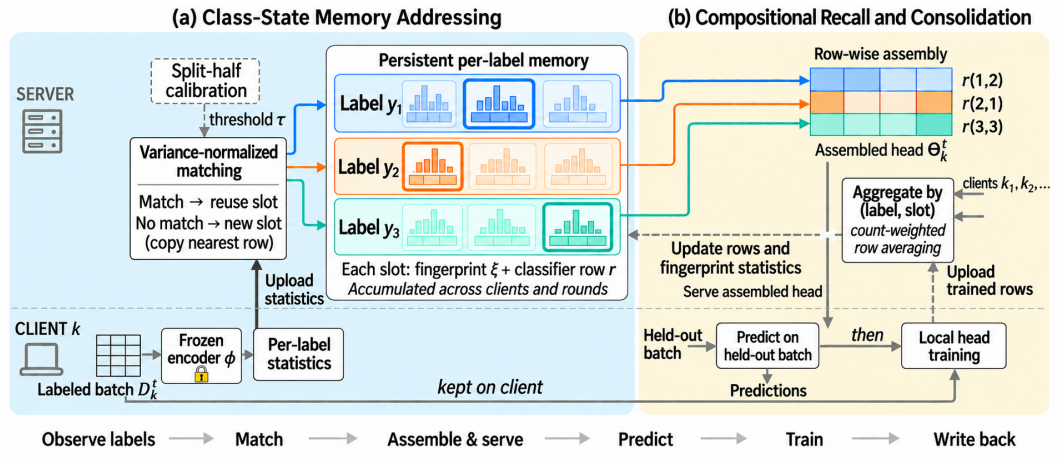

Figure 2: **Framework of FragMem in one round.** (a) Class-state memory addressing matches per-label statistics of the client's labelled batch to persistent server-side slots, creating a slot when no state matches; each slot stores a feature-statistics fingerprint and a classifier row. (b) Compositional recall and consolidation assembles the selected rows into a client-specific head; after held-out prediction and local training, the server averages the returned rows per (label, slot) and accumulates their statistics. Slots remain available in later rounds; the encoder stays frozen.

over a run; heavier tails can raise the actual rate, so a new slot is an estimated new state without an exact error-control guarantee. Each stored row can subsequently be retrieved independently of the complete concept in which it was learned.

4.2 Compositional Recall and Consolidation

Recall across the federation's history. The server assembles client k's head as $\Theta_k^t = \mathrm{Stack}_{c=1}^{C}\, r_{c,a_{k,c}^t}$, one row per label. Its rows can come from different clients, rounds and concepts, and recalling a stored state does not require another client to hold it in the current round.

Consolidate across clients and rounds. The client predicts its held-out batch, trains the whole assembled head on its labelled batch, and uploads rows and statistics. For each (label, slot), the server averages returned rows by label count and accumulates their statistics. Clients whose other label states differ can update the same slot, which remains available in later rounds. *Consolidation* denotes this averaging and accumulation, with no additional anti-forgetting regulariser. Rows learned in different combinations are only approximately compatible (Section 3.3).

Variants and control. *Per-class loss matching*, our main class-level baseline, keeps the banks and novelty rule but picks each label's slot by cross-entropy on its samples (Section 5.3); the *sequential mean test* re-tests a slot's recent batches (Section 5.7); "FragMem, concept units" runs the same code with one unit per concept and three secondary settings. Appendix C specifies them.

5 Experiments

5.1 Setup

Datasets. Three feature pools with 20 classes and 120 images or documents per class: CIFAR-100 at its 20 superclasses (Krizhevsky, 2009) and 20 GTSRB classes (Stallkamp et al., 2012), encoded by an ImageNet ResNet-18 (He et al., 2016; Deng et al., 2009), and 20 Newsgroups (Lang, 1995) with TF-IDF; all are reduced to $d = 128$ (Appendix B). Section 5.7 adds a CIFAR-100 fine-class pool for many-to-one mappings, GoEmotions with sentence embeddings and DINOv2 features.

Table 1: **Switch-round accuracy (%)**; H1: whole concepts recur (negative control); H2: recombination; baselines in Appendix D. Mean $\pm$ 95% t half-width, seeds 44–48. "Rd. t": the served model has read the round-t labelled batch (✓), has not (✗), or is given the true state ($\star$). Bytes: relative to FedAvg on H2 (range over pools). Bold: highest non-oracle mean (not a significance test).

		CIFAR-100		GTSRB		20 Newsgroups		
Method	Rd. t	H1	H2	H1	H2	H1	H2	Bytes
Diagnosis: identity given								
Class-identity oracle	$\star$	63.7 ± 0.3	63.2 ± 0.3	87.2 ± 0.7	86.9 ± 0.3	65.0 ± 0.3	64.5 ± 0.2	1.52
Concept-identity reference	$\star$	63.2 ± 0.4	39.8 ± 2.0	85.8 ± 0.7	54.9 ± 2.8	65.1 ± 0.2	38.4 ± 2.3	1.52
Diagnosis: same memory, concept or class units								
FragMem, concept units	✓	62.4 ± 0.5	39.8 ± 2.0	85.6 ± 0.7	54.9 ± 2.8	61.6 ± 0.9	38.4 ± 2.2	2.03
Loss matching (class units)	✓	62.8 ± 1.5	62.4 ± 1.1	84.4 ± 2.2	83.1 ± 1.5	64.3 ± 1.5	63.7 ± 1.1	4.45–4.61
FragMem (ours)	✓	$\mathbf{63.6 \pm 0.4}$	$\mathbf{63.1 \pm 0.3}$	86.8 ± 0.7	86.6 ± 0.6	64.9 ± 0.3	64.4 ± 0.2	2.03
FragMem + sequential mean test	✓	63.5 ± 0.4	63.1 ± 0.3	$\mathbf{86.9 \pm 0.7}$	$\mathbf{86.7 \pm 0.5}$	$\mathbf{64.9 \pm 0.3}$	$\mathbf{64.4 \pm 0.2}$	2.03
Existing methods: no drift handling								
FedAvg	✗	17.8 ± 2.0	22.9 ± 1.1	22.5 ± 1.6	30.7 ± 1.0	21.3 ± 1.9	24.7 ± 0.9	1.00
Local only	✗	16.1 ± 1.6	38.4 ± 2.1	19.9 ± 2.2	53.8 ± 3.0	18.3 ± 1.8	36.4 ± 2.4	0
Existing methods: concept level								
IFCA	✓	49.6 ± 3.9	39.9 ± 1.9	65.7 ± 5.6	52.5 ± 4.9	46.2 ± 6.4	38.2 ± 2.1	61.92
FedRC	✓	45.6 ± 6.1	38.0 ± 2.5	65.5 ± 5.6	49.9 ± 4.8	50.2 ± 8.0	35.4 ± 2.2	60.80
FedDrift	✗	12.7 ± 1.0	32.4 ± 3.3	15.2 ± 2.1	46.9 ± 4.9	16.4 ± 2.5	29.9 ± 5.6	1.00
FLUX	✗	16.6 ± 2.1	39.9 ± 1.9	20.3 ± 2.5	55.0 ± 2.7	19.7 ± 2.5	38.4 ± 2.2	1.17
FEROMA (τ=$1/K$)	✓	19.3 ± 2.4	34.6 ± 1.4	26.2 ± 3.1	54.8 ± 1.7	22.1 ± 2.2	33.0 ± 1.6	1.17
Existing methods: class level, same round								
FedCCFA	✗	15.9 ± 1.7	40.8 ± 1.6	19.2 ± 2.0	56.4 ± 1.4	19.0 ± 1.8	39.0 ± 1.6	1.50

Federation and statistics. 20 clients in 4 groups, $T = 100$ rounds with a 20-round burn-in, $B = 5$ blocks of 4 labels, $K = 6$ fragments per block, 400 labelled and 400 held-out evaluation samples per client-round, and a linear head. Seeds 44–48 confirm; seeds 42–43 developed the identification rule. The seed is the statistical unit (means with 95% t half-widths, df $= 4$; paired differences with 95% t intervals; no multiplicity correction).

Compared methods. Two identity-given learners, the *class-identity oracle* and the *concept-identity reference*, isolate the memory unit at perfect identification (Section 5.2); two controls share FragMem's pipeline, *FragMem with concept units* (one slot per whole concept) and *per-class loss matching* (class units, cross-entropy matcher; Section 4). Existing methods cover no drift handling (FedAvg, local training), concept-level clustering and drift handling (IFCA, FedRC, FedDrift, FLUX, FEROMA), class-level aggregation within a round (FedCCFA) and, in Appendix D, further personalised, clustered and drift-aware methods, fine-tuning and no-memory per-round refits. Baselines follow their papers or official code; IFCA and FedRC receive a favourable cluster count.

Benchmark. Fragments are permutations within a block, the identity and $K - 1$ derangements chosen greedily to maximise their smallest Hamming distance. Groups share a concept sequence and switch independently with probability 0.3 per round. A *pure* burn-in shows one concept per fragment index, so every fragment is seen; a *mixed* burn-in adds, in its second half, a random walk that changes q blocks per step. Afterwards each switch changes q blocks and, with probability ρ, targets an unseen combination; an *exclusive* rule prefers target fragments that no other group held in the previous round. **H1** ($q = B$, $\rho = 0$, pure burn-in) is the negative control: whole concepts recur. **H2** ($q = 1$, $\rho = 1$, mixed burn-in) is recombination: every switch changes one block and lands on an unseen combination almost never held by another group in the previous round. The main metric is *switch-round accuracy*, the mean evaluation accuracy over the (client, round) cells whose concept has just changed; we also report the next two rounds and stable rounds. Methods that serve before reading the batch are marked in Table 1.

5.2 Main results: identification or recall?

Identity given. Two learners receive the true state and otherwise follow FragMem's pipeline. The *class-identity oracle* keeps rows per (label, state); the *concept-identity reference* keeps one head per concept and, at an unstored concept, copies the client's previous slot, stale on the changed blocks. On H2 the reference trails the oracle by 23.4, 32.0 and 26.1 points (Table 1), on H1 by -0.1 to 1.4: with identification error removed, whole-head recall still trails by 23–32 points at unseen

Table 2: **(a) Ablation** (CIFAR-100): variant minus FragMem at the switch round, points (mean $\pm$ 95% t half-width, seeds 44–48; Appendix C). **(b) Robustness** (H2): FragMem minus the concept-identity reference (the class-identity oracle is at most 0.35 points above FragMem; "–": not run; Table 5).

(a)

Variant − FragMem	H1	H2
No memory (best refit)	-8.61 ± 0.39	-8.25 ± 0.39
Concept units	-1.15 ± 0.32	-23.31 ± 2.06
Block units (known partition)	-0.41 ± 0.14	-0.99 ± 0.24
Previous-slot init.	-0.19 ± 0.08	-0.05 ± 0.05
+ slot merging	$+0.01 \pm 0.05$	$+0.01 \pm 0.03$
Concept-unit control's 3 settings	-0.10 ± 0.07	$+0.02 \pm 0.11$
Fixed z=3, pooled variance	-0.06 ± 0.23	$+0.01 \pm 0.08$

(b)

Setting	CIFAR-100	GTSRB	20 Newsgroups
Switch rate 0.1	$+14.7 \pm 1.5$	$+19.9 \pm 2.0$	$+16.4 \pm 1.3$
Switch rate 0.5	$+26.3 \pm 1.3$	$+35.7 \pm 1.3$	$+28.9 \pm 1.1$
K=3 fragments per block	$+26.4 \pm 2.1$	$+35.9 \pm 2.0$	$+28.9 \pm 1.2$
48 clients	$+23.4 \pm 2.2$	$+31.7 \pm 2.7$	–
100 clients	$+23.2 \pm 2.3$	$+31.8 \pm 2.8$	–
Semantic blocks	$+21.9 \pm 2.7$	–	–

combinations. The reference is neither the proposition's best stored head nor an upper bound for every concept-level strategy; it isolates how memory is organised at fixed identification. Concept-level methods that read the batch before serving (FragMem with concept units, IFCA, FedRC) stay near the reference on H2, at 38.0–39.9, 49.9–54.9 and 35.4–38.4%: learned identification does not change what one stored head can serve (Section 3.3).

FragMem approaches the class oracle. FragMem comes within 0.12, 0.29 and 0.16 points of the class-identity oracle. Against the same pipeline with concept units (three secondary settings differ, Section 4; they matter by at most 0.10 points, Table 2a), it gains $+23.3\,[21.3,\ 25.4]$, $+31.7\,[29.0,\ 34.5]$ and $+26.0\,[23.9,\ 28.1]$ points (5/5 seeds), and per-class loss matching, a different matcher over the same memory, comes within 0.6–3.5 points of FragMem (Section 5.3). Within this protocol the gap is thus not a matter of identification but of the memory unit.

Existing methods. FedCCFA aggregates per class but serves before reading the round's batch; at switches it is within 0.7 points of local training on H1 and 2.3–2.6 above it on H2. Where timing matters less, FragMem at the switch leads FedCCFA *one round later* by 17.9, 23.8 and 21.6 points and in stable rounds by 12.1, 15.3 and 16.6 (H2, 5/5 seeds), consistent with FedCCFA's lack of a cross-round bank but not isolating it: raising the co-presence of the target fragment in the previous round narrows the gap by only 2.2 of about 22 points (Appendix B). FragMem also leads FEROMA, which reads current-round profiles (33.0–54.8% on H2), and each further baseline by more than 24 points on H2 (Appendix D; FeSEM H2 incomplete).

Negative control. When whole concepts recur (H1), FragMem minus the concept reference is $+0.40\,[0.30,\ 0.50]$, $+0.98\,[0.35,\ 1.60]$ and $-0.23\,[-0.47,\ 0.02]$ points. The difference stays within ± 1 point on CIFAR-100 and 20NG but **not on GTSRB**, where FragMem is better; the excess is concentrated on labels whose (label, class) pair is shared across concepts, so that their class-level slots pool data (shared labels gain $+2.27$ points, unshared lose 1.29; difference $+3.55\,[2.32,\ 4.79]$): an association consistent with data sharing, not a controlled test.

5.3 Ablation study

Persistent memory, not a better per-round fit. The best per-round refit of the head without memory (Appendix C) trails FragMem by $8.25\,[7.86,\ 8.64]$, $4.44\,[3.86,\ 5.02]$ and $7.49\,[7.24,\ 7.73]$ points on H2 (Table 2a; 8.61 on CIFAR-100 H1): the advantage survives the tested refits.

Unit of memory. Concept units instead of class units cost 23.3 points on H2 and 1.15 on H1 (CIFAR-100, Table 2a; other pools in Table 1), whereas known blocks as units come within 0.41 (H1) and 0.99 (H2) points of labels. What matters is a unit below the whole concept, not the class itself; class units need no knowledge of the blocks.

Matcher. FragMem minus per-class loss matching is $+0.71\,[-0.45,\ 1.86]$, $+3.54\,[2.18,\ 4.90]$ and $+0.65\,[-0.56,\ 1.86]$ on H2; with previous-slot initialisation the differences are $+0.66\,[-0.50,\ 1.82]$, $+3.49$ and $+0.60$. We claim no general accuracy advantage; that two matchers over the same memory come this close points to the memory unit as the source of the gain.

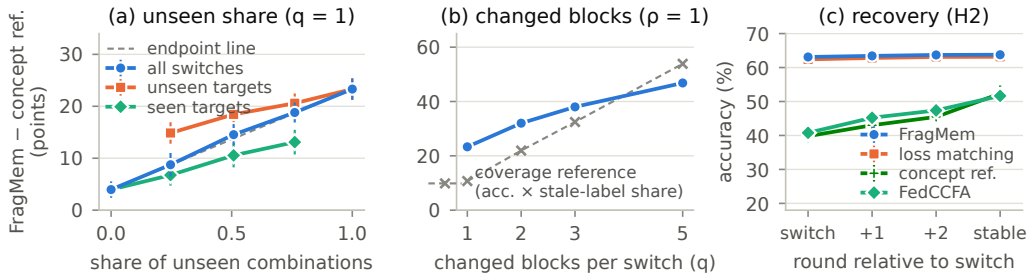

Figure 3: **Dose response and recovery** (CIFAR-100). (a, b) Switch-round gap between FragMem and the concept-identity reference against the realised share of switches into unseen combinations ($q = 1$; also split by target) and against the number of changed blocks q ($\rho = 1$; with the coverage reference of Section 5.4). (c) H2 accuracy at the switch round, the next two rounds and stable rounds (all pools: Figure 4; FragMem with concept units, within 0.06 points of the concept reference, is omitted). Error bars: 95% t half-widths, seeds 44–48.

Secondary choices. A fixed threshold with pooled variance, slot merging or previous-slot initialisation each change accuracy by at most 0.2 points (Table 2a), and varying α over 0.005–5 moves it by up to 0.52 points. We do not claim these choices are needed (the fixed threshold did not open more slots, against expectation); simpler persistent alternatives are untested.

5.4 Why does fine-grained memory help?

Recombination demand. If the gap comes from recombination, it should grow with its demand. It grows linearly with the share of switches into unseen combinations (Figure 3a): the slope is $+19.4\,[16.9,\ 21.9]$ points per unit share, and the middle settings lie within 0.75 points of the endpoint line. It also grows with the number of changed blocks, by $+5.7$ points per block, reaching $+46.8$ at $q = 5$ (Figure 3b). When no combination is new ($\rho = 0$), the gap is only $+4.0$ points.

Beyond combinatorial coverage. If the stale block were the only loss, the reference would lose FragMem's accuracy times the share of labels on which the previous concept is stale (the *coverage reference* of Figure 3b; Appendix A): 10.7 points on CIFAR-100 with one changed block, under half the observed 23.3; it under-predicts the loss for $q \leq 3$ and over-predicts it at $q = 5$. Coverage is thus one, structural part of the gap. Nor does recombination demand alone explain it: at switches into *seen* combinations the gap still grows with ρ (6.8 to 13.1 points), so the unseen-minus-seen difference is only $+7.8$ points. The remaining gap correlates with training exposure: an H2 schedule visits more than 100 distinct concepts, so each concept slot trains on few rounds; the $\log_2$ number of sample presentations (not distinct examples) behind the served rows, L (FragMem minus the concept reference), rises with ρ (slope $+2.66\,[2.21,\ 3.12]$ per unit share), and at seen switches the gap rises by $+4.60\,[4.07,\ 5.13]$ points per unit of L: more concepts spread the same data over more concept slots, whereas a class-level slot pools the updates of every concept that shares its fragment. These are associations; we did not control exposure.

5.5 Robustness

On H2 the lead over the concept reference holds across switch rates 0.1 and 0.5, $K = 3$ fragments per block, 48 and 100 clients and semantic blocks (Table 2b: $+14.7$ to $+35.9$ points; $K = 10$ did not run). The gap narrows but does not close after a switch (Figure 3c; Figure 4): FragMem leads the concept reference by 20.4, 27.6 and 23.6 points one round later and by 11.4, 14.3 and 15.0 in stable rounds, consistent with the statistical part of Section 5.4, which the curves do not establish.

5.6 Communication and efficiency

Per client and round, FragMem uploads per-label statistics, downloads one assembled head and uploads the trained head: $2.03\times$ the bytes of FedAvg (Appendix C). Loss matching downloads

every candidate row of every label each round, 4.45–4.61× (uncached; a row cache is untested); IFCA and FedRC cost about 60×; FedCCFA, FLUX and FEROMA are cheaper (1.17–1.50×) but less accurate (Figure 5, Appendix E). FragMem thus obtains class-level compositional recall without downloading all candidate rows; the sequential mean test of Section 5.7 uploads nothing extra.

5.7 Challenging settings and failure modes

Identifiability: many-to-one conventions. On a pool of CIFAR-100 fine classes, each fragment lumps one pair of neighbouring classes into a single label (Appendix B). FragMem still leads the concept-level methods by 6.7–9.3 points on H2-lump, FedRC and FedDrift by about 11, FedCCFA by 11.5 and loss matching by 9.7 (Table 3a). But purity falls to 0.92–0.94 (Table 7), a slot mixes 1.73 states on average, and when whole concepts recur FragMem trails the concept reference by 1.12 [0.55, 1.70] points and the class oracle by 1.48 and 1.31. One batch carries little evidence of a lumped neighbour (Appendix F). The sequential mean test, which accumulates evidence across rounds, raises accuracy by $+0.53$ and $+0.48$ points (5/5 seeds) at the same bytes and purity to 0.96–0.97, and narrows the H1-lump deficit to $-0.59\,[-0.76, -0.43]$; a directional variant with 1.55–1.82× the bytes was not shown to be better (Table 3b). A class-oracle gap of 0.95/0.83 points remains at the switch and 0.73/0.82 in stable rounds: the loss is not confined to the detection window.

Identifiability: few labels and imbalance. With 5 samples per class and round, FragMem leads the concept reference by 21–26 points on H2 but trails the class oracle by 2.6–6.3, and loss matching is 0.65–2.07 points better. With 1 sample per class, identification fails in a one-seed diagnostic (Table 5). Under class imbalance (Dirichlet 0.5), FragMem leads the concept reference by 19–24 points on H2 but is 3.9, 7.0 and 4.0 points *below* it on H1, with purity 0.83–0.87.

Recombination demand: a real convention. GoEmotions (Demszky et al., 2020) is a real many-to-one convention: 12 fine emotions in 4 Ekman groups, labelled finely or by group (Appendix B); clients and switch times are synthetic. With 16 combinations, only about 8% of H2 switches are unseen. FragMem leads FedAvg, FedCCFA, FEROMA, loss matching and the best refit by 3.4–11.9 points (Table 4a), but the concept reference by only $+2.56\,[1.18, 3.94]$, and is 1.69 below the class oracle. Over concept units it gains $+0.62\,[0.05, 1.18]$ on H2 and loses 0.64 on H1; split by target, the H2 gain comes from unseen targets ($+5.29$; seen $+0.25$; stable -0.87; Appendix G). Low purity (0.672/0.682) accompanies the small unseen share; we cannot separate them.

Value over the current batch. With DINOv2-small features (Oquab et al., 2024) and an MLP head, FragMem matches the class-identity oracle (84.2%) and leads the concept reference by 21.5 [20.3, 22.8] points and FedBABU by 7.7 (Table 4b), at 1.07× FedAvg bytes. FedAvg fine-tuned on the round's labelled batch reaches 80.9%, only 3.23 [2.26, 4.20] points lower at 1.00×: with strong features, most of the benefit of current labels comes without memory.

6 Limitations and conclusion

Scope. We construct concept drift through class-to-label mappings on small fixed feature pools, with frozen encoders and labels before prediction; the GoEmotions experiment uses a real label grouping but synthetic client schedules. Establishing fragment recurrence in deployed systems requires longitudinal data with documented changes. **Identification.** Few labels, imbalance and many-to-one mappings degrade identification: loss matching outperforms FragMem at 5 samples per class, and incorrect slots leave FragMem 3.9–7.0 and 1.1 points below the concept reference when whole concepts recur; the sequential test halves the latter deficit.

Conclusion. A client's new concept can combine fragments encountered elsewhere in the federation. FragMem makes their class-level knowledge reusable through persistent label-state slots: clients retrieve rows and contribute updates to shared states. The results support federated memory that recombines knowledge across clients and time when concept fragments recur.

AI USE STATEMENT

We used generative AI tools (large language model assistants) for the following: polish writing, retrieval and discovery.

ETHICS STATEMENT

All datasets are public: CIFAR-100, GTSRB, 20 Newsgroups and GoEmotions. GoEmotions contains Reddit comments, some of them offensive. We use its released comments and labels only to compute sentence embeddings; we add no annotation and use no user identifiers. Potential applications such as content moderation and medical coding are high-stakes. Recalling an incorrect class state could silently apply an inappropriate label mapping, so deployments would need to audit the states selected for prediction. We report failure modes in Section 5.7.

REPRODUCIBILITY STATEMENT

Sections 3 to 5 and Appendices B to D specify the generator, protocol, method, baselines, hyperparameters, seeds and statistics. The frozen-feature experiments ran on serverless CPU workers; the whole study, including one GPU (RTX 4090) session, cost about US$42 at the most expensive tier's price. The code, configurations, analysis scripts and per-seed result records are organised for anonymous release.

REFERENCES

Gabriel J. Aguiar and Alberto Cano. A comprehensive analysis of concept drift locality in data streams. *Knowledge-Based Systems*, 289:111535, 2024. doi: 10.1016/j.knosys.2024.111535.

Giuseppe Canonaco, Alex Bergamasco, Alessio Mongelluzzo, and Manuel Roveri. Adaptive federated learning in presence of concept drift. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2021. doi: 10.1109/IJCNN52387.2021.9533710.

Junbao Chen, Jingfeng Xue, Yong Wang, Zhenyan Liu, and Lu Huang. Classifier clustering and feature alignment for federated learning under distributed concept drift. In *Advances in Neural Information Processing Systems*, volume 37, pp. 81360–81388. Curran Associates, Inc., 2024. doi: 10.52202/079017-2586. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/942e820be4aa112509b3a281ff398851-Paper-Conference.pdf.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. GoEmotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4040–4054, Online, 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.372. URL https://aclanthology.org/2020.acl-main.372/.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009. doi: 10.1109/CVPR.2009.5206848.

Yuyang Deng, Mohammad Mahdi Kamani, and Mehrdad Mahdavi. Adaptive personalized federated learning, 2020. URL https://arxiv.org/abs/2003.13461.

Jiahua Dong, Lixu Wang, Zhen Fang, Gan Sun, Shichao Xu, Xiao Wang, and Qi Zhu. Federated class-incremental learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10154–10163. IEEE, 2022. doi: 10.1109/CVPR52688.2022.00992.

Liyu Fang, Wu Wen, and Xiaolin Zheng. FedDCA: Stable and unified Wasserstein adaptation to federated concept drift. *Knowledge-Based Systems*, 337:115342, 2026. doi: 10.1016/j.knosys.2026.115342.

Eros Fanì, Raffaello Camoriano, Barbara Caputo, and Marco Ciccone. Accelerating heterogeneous federated learning with closed-form classifiers. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 13029–13048. PMLR, 2024. URL https://proceedings.mlr.press/v235/fani-24a.html.

Dario Fenoglio, Mohan Li, Pietro Barbiero, Nicholas Lane, Marc Langheinrich, and Martin Gjoreski. FLUX: Efficient descriptor-driven clustered federated learning under arbitrary distribution shifts. In *Advances in Neural Information Processing Systems*, volume 38, pp. 71229–71292. Curran Associates, Inc., 2025. doi: 10.52202/085713-2395. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/67496dfa96afddab795530cc7c69b57a-Paper-Conference.pdf.

João Gama and Petr Kosina. Recurrent concepts in data streams classification. *Knowledge and Information Systems*, 40(3):489–507, 2014. doi: 10.1007/s10115-013-0654-6.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):1–37, 2014. doi: 10.1145/2523813.

Avishek Ghosh, Jichan Chung, Dong Yin, and Kannan Ramchandran. An efficient framework for clustered federated learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19586–19597. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/e32cc80bf07915058ce90722ee17bb71-Paper.pdf.

Nuwan Gunasekara, Bernhard Pfahringer, Heitor Murilo Gomes, Albert Bifet, and Yun Sing Koh. Recurrent concept drifts on data streams. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 8029–8037. International Joint Conferences on Artificial Intelligence Organization, 2024. doi: 10.24963/ijcai.2024/888. Survey Track.

Lan-Zhe Guo, Zhi Zhou, Yu-Feng Li, and Zhi-Hua Zhou. Identifying useful learnwares for heterogeneous label spaces. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12122–12131. PMLR, 2023. URL https://proceedings.mlr.press/v202/guo23l.html.

Yongxin Guo, Xiaoying Tang, and Tao Lin. FedRC: Tackling diverse distribution shifts challenge in federated learning by robust clustering. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 16910–16944. PMLR, 2024. URL https://proceedings.mlr.press/v235/guo24f.html.

Ben Halstead, Yun Sing Koh, Patricia Riddle, Mykola Pechenizkiy, Albert Bifet, and Russel Pears. Fingerprinting concepts in data streams with supervised and unsupervised meta-information. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pp. 1056–1067. IEEE, 2021. doi: 10.1109/ICDE51399.2021.00096.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. IEEE, 2016. doi: 10.1109/CVPR.2016.90.

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*, volume 206 of *Proceedings of Machine Learning Research*, pp. 5834–5853. PMLR, 2023. URL https://proceedings.mlr.press/v206/jothimurugesan23a.html.

Ioannis Katakis, Grigorios Tsoumakas, and Ioannis Vlahavas. Tracking recurring contexts using ensemble classifiers: an application to email filtering. *Knowledge and Information Systems*, 22(3):371–391, 2010. doi: 10.1007/s10115-009-0206-2.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Ken Lang. NewsWeeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pp. 331–339. Elsevier, 1995. doi: 10.1016/B978-1-55860-377-6.50048-7.

Gyuejeong Lee and Daeyoung Choi. Class-wise federated averaging for efficient personalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1773–1782, 2025. doi: 10.1109/ICCV51701.2025.00173.

Minghao Li, Dmitrii Avdiukhin, Rana Shahout, Nikita Ivkin, Vladimir Braverman, and Minlan Yu. FIELDING: Clustered federated learning with data drift. In *The 29th International Conference on Artificial Intelligence and Statistics*, 2026a. URL https://openreview.net/forum?id=i4q2xjAuld.

Mohan Li, Dario Fenoglio, Martin Gjoreski, and Marc Langheinrich. Federated learning with profile mapping under distribution shifts and drifts. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=thoPskdIcE.

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*, volume 139 of *Proceedings of Machine Learning Research*, pp. 6357–6368. PMLR, 2021. URL https://proceedings.mlr.press/v139/li21h.html.

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*, volume 80 of *Proceedings of Machine Learning Research*, pp. 3122–3130. PMLR, 2018. URL https://proceedings.mlr.press/v80/lipton18a.html.

Jian-Dong Liu, Zhi-Hao Tan, and Zhi-Hua Zhou. Identifying and reusing learnwares across different label spaces. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI-25*, pp. 5734–5742. International Joint Conferences on Artificial Intelligence Organization, 2025. doi: 10.24963/ijcai.2025/638. Main Track.

Guodong Long, Ming Xie, Tao Shen, Tianyi Zhou, Xianzhi Wang, and Jing Jiang. Multi-center federated learning: clients clustering for better personalization. *World Wide Web*, 26(1):481–500, 2023. doi: 10.1007/s11280-022-01046-x.

Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2019. doi: 10.1109/TKDE.2018.2876857.

Othmane Marfoq, Giovanni Neglia, Aurélien Bellet, Laetitia Kameni, and Richard Vidal. Federated multi-task learning under a mixture of distributions. In *Advances in Neural Information Processing Systems*, volume 34, pp. 15434–15447. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/82599a4ec94aca066873c99b4c741ed8-Paper.pdf.

Jorge A. Mendez and Eric Eaton. Lifelong learning of compositional structures. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=ADWd4TJO13G.

Jaehoon Oh, SangMook Kim, and Se-Young Yun. FedBABU: Toward enhanced representation for federated image classification. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=HuaYQfggn5u.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt.

Kunjal Panchal, Sunav Choudhary, Subrata Mitra, Koyel Mukherjee, Somdeb Sarkhel, Saayan Mitra, and Hui Guan. Flash: Concept drift adaptation in federated learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 26931–26962. PMLR, 2023. URL https://proceedings.mlr.press/v202/panchal23a.html.

Fu Peng and Ming Tang. FedDAA: Dynamic client clustering for concept drift adaptation in federated learning, 2025. URL https://arxiv.org/abs/2506.21054.

Barbara Plank. The "problem" of human label variation: On ground truth in data, modeling and evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 10671–10682, Abu Dhabi, United Arab Emirates, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.731. URL https://aclanthology.org/2022.emnlp-main.731/.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

F. E. Satterthwaite. An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2(6):110–114, 1946. doi: 10.2307/3002019.

Felix Sattler, Klaus-Robert Müller, and Wojciech Samek. Clustered federated learning: Model-agnostic distributed multitask optimization under privacy constraints. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3710–3722, 2021. doi: 10.1109/TNNLS.2020.3015958.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. MPNet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pp. 16857–16867. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/c3a690be93aa602ee2dc0ccab5b7b67e-Paper.pdf.

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012. doi: 10.1016/j.neunet.2012.02.016.

Alysa Ziying Tan, Han Yu, Lizhen Cui, and Qiang Yang. Towards personalized federated learning. *IEEE Transactions on Neural Networks and Learning Systems*, 34(12):9587–9603, 2023. doi: 10.1109/TNNLS.2022.3160699.

Yue Tan, Guodong Long, Lu Liu, Tianyi Zhou, Qinghua Lu, Jing Jiang, and Chengqi Zhang. FedProto: Federated prototype learning across heterogeneous clients. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(8):8432–8440, 2022. doi: 10.1609/aaai.v36i8.20819.

Geoffrey I. Webb, Roy Hyde, Hong Cao, Hai Long Nguyen, and Francois Petitjean. Characterizing concept drift. *Data Mining and Knowledge Discovery*, 30(4):964–994, 2016. doi: 10.1007/s10618-015-0448-4.

Jiaheng Wei, Zhaowei Zhu, Hao Cheng, Tongliang Liu, Gang Niu, and Yang Liu. Learning with noisy labels revisited: A study using real-world human annotations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=TBWA6PLJZQm.

Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In *Advances in Neural Information Processing Systems*, volume 33, pp. 15173–15184. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/ad1f8bb9b51f023cdc80cf94bb615aa9-Paper.pdf.

Xi-Zhu Wu, Wenkai Xu, Song Liu, and Zhi-Hua Zhou. Model reuse with reduced kernel mean embedding specification. *IEEE Transactions on Knowledge and Data Engineering*, 35(1):699–710, 2023. doi: 10.1109/TKDE.2021.3086619.

Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. Federated continual learning with weighted inter-client transfer. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12073–12086. PMLR, 2021. URL https://proceedings.mlr.press/v139/yoon21b.html.

Jia Zhang, Zhi Zhou, Lan-Zhe Guo, and Yu-Feng Li. Enabling small models for zero-shot selection and reuse through model label learning, 2025. URL https://arxiv.org/abs/2408.11449.

Yi-Kai Zhang, Ting-Ji Huang, Yao-Xiang Ding, De-Chuan Zhan, and Han-Jia Ye. Model Spider: Learning to rank pre-trained models efficiently. In *Advances in Neural Information Processing Systems*, volume 36, pp. 13692–13719. Curran Associates, Inc., 2023. doi: 10.52202/075280-0604. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/2c71b14637802ed08eaa3cf50342b2b9-Paper-Conference.pdf.

Zhi-Hua Zhou. Learnware: on the future of machine learning. *Frontiers of Computer Science*, 10 (4):589–590, 2016. doi: 10.1007/s11704-016-6906-3.

Zhi-Hua Zhou and Zhi-Hao Tan. Learnware: small models do big. *Science China Information Sciences*, 67(1):112102, 2024. doi: 10.1007/s11432-023-3823-6.

A Proof of Proposition 1 and accuracy form

Whole heads. A learner that serves one stored head h_s, $s \in S$, serves rows trained for the states $s_1, \ldots, s_C$. This head is trained for the current concept s^* on every label only if $s = s^*$, hence only if $s^* \in S$.

Assembled rows. Let $s^* \in \mathcal{R}_{\text{comp}}$. For every label c, $s^*_c = s_c$ for some $s \in S$, so a row r_{c,s^*_c} trained for that state is stored, and with correct per-label identification the served head is $(r_{1,s^*_1}, \ldots, r_{C,s^*_C})$. Every $s \in S$ lies in $\mathcal{R}_{\text{comp}}$ (take the same s for each label); a valid mapping that takes its states from different seen concepts lies in $\mathcal{R}_{\text{comp}} \setminus S$, so the inclusion is then strict.

Size. Let S contain the K pure concepts $(j, \ldots, j)$, $j = 1, \ldots, K$. Then every block takes all K fragments in S, and $\mathcal{R}_{\text{comp}}$ contains all K^B tuples. □

Accuracy form. Under an idealised compatibility assumption the recall sets translate into accuracies.

Assumption 1 (Row compatibility)**.** *For every label c and state σ there is a row $r_{c,\sigma}$ such that any head $(r_{1,\sigma_1}, \ldots, r_{C,\sigma_C})$ classifies label c's evaluation samples correctly with probability a_c if σ_c is its current state, and with probability 0 if σ_c is stale. Stored heads are assembled from these rows.*

Under Assumption 1 and correct per-label identification, class-level memory serves a head of accuracy $\sum_c \pi_c a_c$ for every $s^* \in \mathcal{R}_{\text{comp}}$, with π_c the evaluation share of label c, since samples of label c are classified correctly with probability a_c. A learner serving one stored whole head attains at most $\max_{s \in S} \sum_c \pi_c a_c \mathbb{1}[s_c = s^*_c]$, because label c contributes $\pi_c a_c$ if $s_c = s^*_c$ and 0 otherwise. If $s^* \notin S$, every $s \in S$ differs from s^* on at least one label, so when all $\pi_c a_c > 0$ the bound is strictly below $\sum_c \pi_c a_c$.

Scope. The proposition and the bound concern learners that serve one stored whole head. It says nothing about how fast a new head is trained. The concept-identity reference in our experiments opens a new slot copied from the client's previous slot when it meets a new state; with $q = 1$ that head is stale on one block. The *coverage reference* in Figure 3b is FragMem's accuracy times the share of labels on which the client's previous concept is stale, that is, the loss predicted if the reference served the previous concept's rows with FragMem's per-label accuracy. It is not the maximum over all stored heads in the accuracy form above. It under-predicts the observed loss for $q \leq 3$ (10.7,

21.9, 32.5 against 23.3, 32.0, 38.0 points), consistent with concept slots in H2 also being trained less: each H2 schedule visits more than 100 distinct concepts. It over-predicts the loss at $q = 5$ (53.9 against 46.8). Assumption 1 also fails in part in practice. Rows trained in different contexts are not exactly score-compatible, and the assembled head of the class-identity oracle is itself trained only on the rounds in which each state occurred.

B BENCHMARK DETAILS

Pools. Each pool has 120 samples per class (20 classes for the permutation pools, 25 fine classes for the lump pool, 12 for GoEmotions). A 20% split per class forms the held-out evaluation pool, and training and evaluation batches are drawn from the respective splits with fixed per-client class proportions. Each client-round has 400 training and 400 held-out evaluation samples with fixed class counts, resampled with replacement from a shared pool of 1,920 training and 480 evaluation examples; intervals condition on this pool. All linear heads share one local budget (Appendix D).

- *CIFAR-100*: the 20 superclasses; ImageNet ResNet-18 penultimate features, PCA to 128 fitted on the whole pool. All unsupervised preprocessing (PCA, TF-IDF, SVD) is fitted on the whole pool, including the evaluation inputs, but never on labels.
- *GTSRB*: 20 of the 43 classes, drawn with a fixed seed; same encoder and PCA. GTSRB is not part of ImageNet.
- *20 Newsgroups*: headers, footers and quotes removed; documents shorter than 100 characters dropped; TF-IDF fitted on the pool, then truncated SVD to 128. Rows are rescaled so that the median row norm matches the CIFAR-100 pool, since the raw scale made all SGD baselines diverge.
- *CIFAR-100 fine (lump)*: 5 superclasses (fish, household furniture, large carnivores, large omnivores and herbivores, reptiles) with their 5 fine classes each. Within each superclass the fine classes are ordered on the shortest open path through their PCA class means (exhaustive over the 60 paths). Fragment $v \in \{0, \dots, 3\}$ lumps positions v and $v + 1$ into one label. This gives 20 labels over 25 fine classes.
- *GoEmotions*: single-label comments from the four groups anger (anger, annoyance, disapproval), joy (joy, admiration, gratitude), sadness (sadness, disappointment, remorse) and surprise (surprise, curiosity, confusion). Each block has two fragments: the fine labels, and the official coarse convention in which all three members take the label of the group's Ekman emotion. Encoder: all-mpnet-base-v2 (Song et al., 2020; Reimers & Gurevych, 2019) (mean pooling, ℓ_2-normalised), PCA to 128, rescaled to the CIFAR-100 median row norm; a linear probe on the 12 classes reaches 0.427 (chance 0.083).
- *DINOv2*: the CIFAR-100 pool images encoded by DINOv2-small (pooled CLS, 384-d; linear probe 0.827), rescaled likewise.

Schedules. Clients: 4 groups $\times$ 5 clients sharing a concept sequence per group; groups switch independently with probability 0.3 per round after a 20-round burn-in; $T = 100$. Fragments: $K = 6$ per block (the identity and 5 derangements for permutation pools; 4 lump fragments; 2 Ekman fragments). We measure co-presence as the share of changed blocks whose target fragment another group held in the previous round; the five clients of a group always share their fragments. The "exclusive" rule makes this share near zero on H2; the $\rho = 0$ arm with $q = 1$ (B) has 0.15–0.22. On the B arm FragMem leads the concept reference by $+3.96\,[2.39,\ 5.53]$ points (CIFAR-100). Raising the rate at which another group held the target fragment in the previous round from about 0 to 0.86 (CIFAR-100) narrows FragMem's lead over FedCCFA at the switch by only $2.2\,[0.05,\ 4.3]$ of about 22 points. The realised unseen share can fall below ρ when no unseen target is reachable; the dose-response settings use realised shares 0, 0.246, 0.508, 0.761 and 1. Unit tests check, among other properties, that each client's class proportions are fixed over rounds, so $P(X)$ does not change.

C METHOD DETAILS

Round of FragMem. For each round t:

1. Every client computes per-label statistics of its batch (sum, count, within-label sum of squares, split-half values) and uploads them.
2. For every client, in order (so a slot's fingerprint already includes statistics uploaded earlier in the round), and for every label c with samples: compute $z_c(j)$ (Equation (1)) for all live slots j of label c. Let $j^* = \arg\min_j z_c(j)$. If $z_c(j^*) > \tau$, open a new slot initialised with the row of j^*; otherwise assign j^*. Add the client's statistics to the slot as pending.
3. Serve every client the head made of its assigned rows. The client predicts its evaluation batch, then trains the head for 10 full-batch epochs.
4. For each (label, slot), set the row to the count-weighted mean of the uploaded rows, and add the statistics to the slot's fingerprint.

Steps 1–2 are class-state memory addressing (Section 4.1); steps 3–4 are compositional recall and consolidation (Section 4.2). With an MLP head, slots keep output rows and the hidden layer is shared and averaged. Labels absent from a client's batch keep the client's previous slot. The per-label variance v_c uses the slot's pooled within-label sum of squares once it has at least 10 degrees of freedom, else the batch's (if $n_c \geq 3$), else a pooled global estimate. The null sample for τ accumulates split-half values over all clients and rounds. Before 100 null values are available, a client's own split-half values are used. The fitted scaled χ^2 has $k = 2\bar{z}^2/\widehat{\mathrm{var}}(z)$ degrees of freedom and scale $\bar{z}/k$.

Concept-unit control. "FragMem, concept units" runs the same code with one unit that holds all C labels. A slot stores a whole head and the fingerprints of all labels. A client's batch is compared with every slot through $z_c(j)$ of each label present in the batch; it takes the slot with the smallest mean over labels and opens a new slot if the largest z_c on that slot exceeds the calibrated threshold. Rows and fingerprints are aggregated per slot as in step 4. Three settings differ from FragMem: (i) after every aggregation, slots are merged greedily, closest pair first, while the largest per-label z between the two slots' committed fingerprints (squared mean distance over the pooled within-label variance times $1/N_a + 1/N_b$) is at most τ; the merged rows are averaged by per-label counts and the fingerprints summed; (ii) the split-half null divides by the pooled variance within the two halves instead of the label's variance over the whole batch; (iii) a new slot copies the client's previous slot instead of the nearest one. With class units these three settings change CIFAR-100 accuracy by -0.10 and $+0.02$ points (Table 2a). The other variants of Table 2a: *no memory* is the best per-round refit of the head without memory (NCM or ridge with $\lambda \in \{0.1, \ldots, 10^5\}$ on the full labelled batch, selected per seed on the evaluation data); *block units* use the known partition into blocks as the memory unit, one slot per (block, fragment); *previous-slot initialisation* copies the client's previous slot into a new slot instead of the nearest one; *fixed* z=3 replaces the calibrated threshold by $\tau = 3$ and the label's variance by a pooled variance.

Per-class loss matching. The main class-level baseline keeps FragMem's banks and novelty rule but chooses each label's slot by the cross-entropy of the client's samples of that label under each candidate row, in two coordinate passes over the labels with the other rows fixed. Our implementation downloads every candidate row of every label each round (no client-side cache) and initialises a new slot from the client's previous slot rather than the nearest one.

Sequential mean test. This optional extension uploads nothing extra. After a label is matched to slot s, for each suffix of the client's last $w \in \{1, \ldots, 4\}$ batches on s, compare the suffix mean with the mean of the rest of slot s by the same z statistic. The threshold is Bonferroni-corrected over tests and suffixes. On detection, the suffix's committed statistics move to an existing slot that passes the same test, or to a new slot whose row is copied. Earlier classifier updates are not undone. On the permutation pools the sequential mean test changes switch-round accuracy by -0.06 to $+0.09$ points (Table 1). The directional variant replaces the mean test by a projection onto directions towards the 8 nearest stored fingerprints. It uses a within-label scatter matrix that clients upload until a row has 200 degrees of freedom, and its null is an $F(1, k)$ law.

Communication. Per client and round, FragMem uploads $C(d{+}2)$ floats for identification, downloads the assembled head ($C(d{+}1)$), and uploads the trained head and the statistics again. FedAvg transfers $2C(d{+}1)$. The recorded ratio of 2.03× follows our byte counter's conventions: it counts 100 rounds of FragMem traffic against FedAvg's 99 aggregation rounds, includes the cross-round

calibration values that clients send before a split-half null exists, and omits the C split-half values per client and round (below 1% of FragMem's traffic). The oracles skip the identification upload ($1.52\times$ under the same conventions). Loss matching downloads all candidate rows of every label. IFCA and FedRC download all cluster models.

Cost. Identification costs $O(Cd \cdot \text{slots per label})$ per client and round on the server. In our runs, the number of slots per state was 1.18–1.64 on permutation pools (Table 7).

D BASELINES

Serving order. We audited every runner in the code for whether the model evaluated at round t has read the round-t labelled batch.

- *Reads the batch before serving:* IFCA and FedRC in their select-first form (lowest loss, plus the label histogram for FedRC), FragMem with concept units, FragMem, loss matching, the no-memory refits, FEROMA and FEROMA-τ (profiles of the batch, after 5 FedAvg warm-up rounds), FedBABU and FedAvg-FT (both fine-tune on the batch before serving).
- *Given the true state instead:* the two oracles.
- *Serve a model trained on rounds up to $t-1$:* all other methods (FedAvg, local only, FedDrift, FLUX, FedCCFA, Ditto, APFL, CFL, FeSEM, Flash, Adaptive-FedAvg, FedProto).

Apart from the fine-tuning of FedBABU and FedAvg-FT and the per-round refits, the rows a method serves at round t were trained on rounds up to $t-1$ only; the first group uses the round-t batch to *choose* what to serve.

Settings. All linear heads use the same local budget (full-batch SGD, learning rate 0.02, 10 epochs).

- *Favourable settings for loss-based baselines:* IFCA and FedRC get $\max(4, \#\text{true concepts})$ clusters; the library of IFCA with concept units is sized from the true number of states; FeSEM on H2 gets 129 clusters, which made it too slow to finish except for one CIFAR-100 seed. Loss matching gets no such information: it opens slots with FragMem's novelty rule.
- *FedCCFA:* the official balanced-classifier step, MADD distances with DBSCAN ($\epsilon = 0.1$, one sample minimum), and uniform per-class averaging within clusters. With frozen features there is no representation to align, so the prototype term is inactive.
- *FLUX:* the official label-conditional descriptor and knee-point ϵ for DBSCAN, without fallback.
- *FEROMA:* the released code's inverse-distance weights over class-conditional profiles; FEROMA-τ adds the paper's threshold $\tau = 1/K$.
- *Flash:* the server update of Algorithm 1 with server step 0.01 (best of seven values on synthetic data), $\beta_1 = 0.9$, $\beta_2 = 0.99$ and $\tau = 10^{-3}$. Two changes: the client early stopping is replaced by the common local budget, and the denominator is floored at τ.
- *FedProto:* local models with per-round prototypes; we report both local-model and prototype inference.
- *FedBABU and FedAvg-FT:* fine-tune a copy of the model on the round's batch (momentum 0.9, 5 epochs, batch 50, learning rate 0.03) before predicting, and discard the copy afterwards. We chose the learning rate on synthetic data because the code default of 10^{-3} gave 0.384 there.
- *MLP heads* (DINOv2, width 256) use learning rate 0.3 for every method, chosen on synthetic data with the class oracle; the budget is otherwise unchanged.

E IDENTIFICATION QUALITY AND COMMUNICATION

Table 7 (at the end of the appendix, with Table 6) reports purity and slot counts. On permutation pools FragMem reaches purity 0.997 or more with 1.18–1.64 slots per state; IFCA's clusters mix many states on H2. Figure 4 shows recovery after a switch on all three pools (Section 5.5; the main text shows CIFAR-100 in Figure 3c), and Figure 5 places each method on the accuracy–communication plane (Section 5.6).

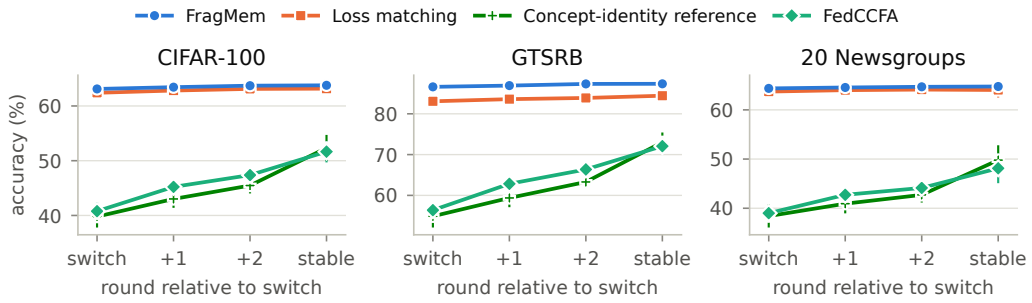

Figure 4: **Recovery after a switch** (H2, all pools): accuracy at the switch round, the next two rounds and stable rounds. FragMem with concept units lies within 0.06 points of the concept-identity reference at every stage and is omitted. Error bars: 95% t half-widths (seeds 44–48).

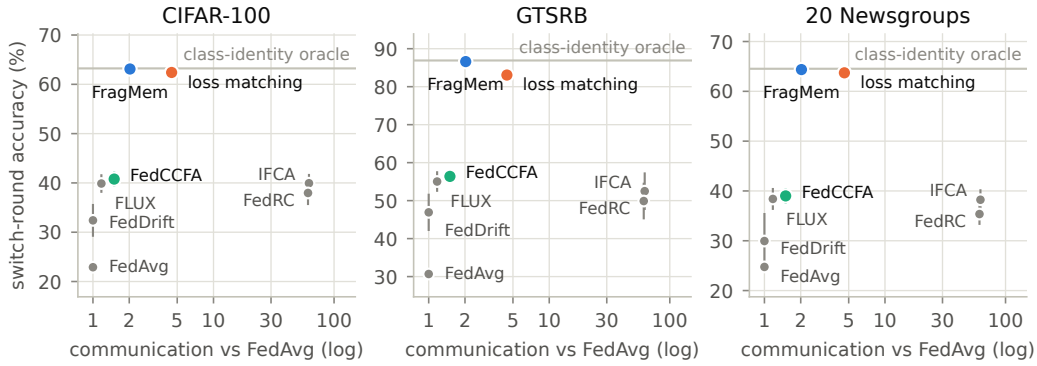

Figure 5: **Accuracy against communication** (H2): switch-round accuracy against bytes relative to FedAvg (log scale); error bars: 95% t half-widths, seeds 44–48. FLUX as in Table 1. FragMem + sequential mean test has the same bytes as FragMem and lies within 0.1 points of it (Table 1); it is omitted, as is the directional sequential variant, which uploads scatter matrices (Appendix C).

F THE SINGLE-BATCH IDENTIFICATION LIMIT

Figure 6 shows why single-batch tests miss many-to-one changes on the lump pool. We took the 20 adjacent fine-class pairs on the chains and used the true class covariances with the best test direction. The estimated noncentrality of a lump change in one batch has median 11.7 (quartiles 8.1 and 17.3), while the Bonferroni cutoff is 23.8, so only 5% of pairs exceed it. This estimate is optimistic: the method tests more directions and has only about 16 samples per label. Accumulated over the ≈ 3.3 rounds a client stays on a slot, the median would pass the cutoff, which motivates the sequential tests. The multi-batch boxes in panel (c) are extrapolated, not measured.

Split-state control. To ask whether slot splitting explains the lump pool's remaining gap to the class oracle, we artificially split 30% of the oracle's states into two slots. This reproduces only 3–5% of the original method's gap.

Loosened-threshold control. In the lump confirmation, the single-batch mean test with a loosened threshold ($\alpha = 20$) reached 70.00 / 70.17% (H1-lump / H2-lump; purity 0.981 / 0.984) but opened about 4.24 slots per state. The directional sequential test exceeded it by $+0.30\,[0.22,\ 0.39]$ and $+0.32\,[0.12,\ 0.52]$ points. This control was run against the directional test; it is not a direct comparison with the sequential mean test, which was compared with the directional test separately:

directional minus sequential mean is $+0.07\,[-0.03,\ 0.17]$ and $+0.03\,[-0.06,\ 0.12]$ points on H1-lump and H2-lump (Table 3b).

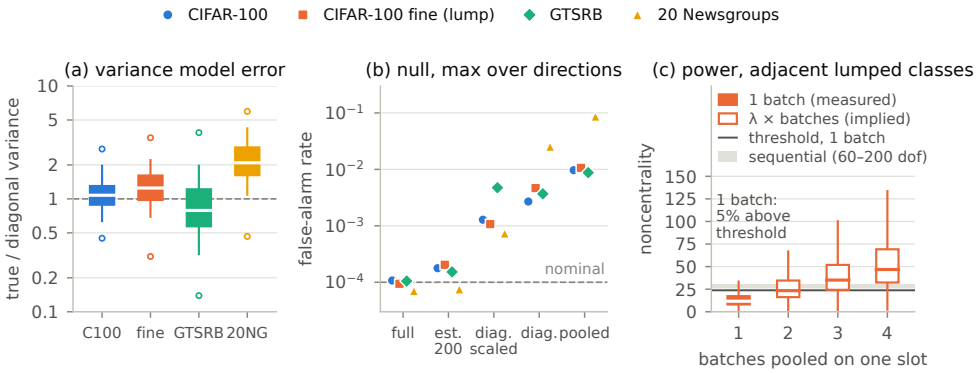

Figure 6: **Single-batch identification limit** on the lump pool. (a) Error of a diagonal variance model along test directions. (b) False-alarm rates under five variance models (nominal 10^{-4}). (c) Noncentrality of neighbouring lumps per batch (measured) and extrapolated to several batches, with the single-batch and sequential thresholds.

G Full tables for Section 5.7

Table 3 gives the results behind the many-to-one paragraph of Section 5.7, and Table 4 those behind its GoEmotions and DINOv2 paragraphs; the component ablation is Table 2a in the main text. On GoEmotions, split by target, FragMem minus concept units is $+5.29\,[2.01,\ 8.58]$ at unseen targets, $+0.25\,[-0.23,\ 0.73]$ at seen targets and $-0.87\,[-1.34,\ -0.39]$ in stable rounds.

Table 3: **(a) Many-to-one drift** (CIFAR-100 fine classes; a fragment lumps two neighbouring classes): switch-round accuracy (%), mean ± 95% t half-width; bytes relative to FedAvg. **(b) Sequential variants** (lump pool): paired difference to FragMem.

(a)

Method	H1-lump	H2-lump	Bytes
Class-identity oracle	71.2 ± 0.4	71.3 ± 0.2	1.52
Concept-identity ref.	70.8 ± 0.4	63.3 ± 0.5	1.52
FedAvg	58.5 ± 0.3	59.1 ± 0.3	1.00
No memory (best refit)	61.1 ± 0.3	61.0 ± 0.3	0
IFCA	61.9 ± 5.7	60.7 ± 2.2	60.91
FedRC	58.0 ± 3.9	58.8 ± 1.2	59.80
FedDrift	58.6 ± 0.3	59.2 ± 0.2	1.00
FragMem, concept units	$\mathbf{70.5 \pm 0.4}$	62.4 ± 0.8	2.03
FEROMA (τ=1/K)	59.4 ± 0.3	60.2 ± 0.5	1.17
FedCCFA	57.6 ± 0.7	58.5 ± 0.8	1.50
Loss matching	60.0 ± 0.5	60.3 ± 0.6	3.97
FragMem	69.7 ± 0.5	70.0 ± 0.5	2.03
FragMem + seq. mean	70.2 ± 0.3	$\mathbf{70.5 \pm 0.3}$	2.03

(b)

Variant − FragMem	H1-lump	H2-lump
Seq. mean test	$+0.53 \pm 0.44$	$+0.48 \pm 0.37$
Directional seq.	$+0.60 \pm 0.43$	$+0.51 \pm 0.32$
Mean, α=20	$+0.30 \pm 0.49$	$+0.19 \pm 0.46$

Table 4: **(a) GoEmotions, Ekman grouping** (16 combinations): switch-round accuracy (%); H2 also split by unseen and seen targets and stable rounds. **(b) DINOv2 features** (CIFAR-100, H2), MLP (width 256) or linear head; "+FT": fine-tuned on the round's labelled batch before serving. Mean ± 95% t half-width, seeds 44–48.

(a)

			H2 switches		
Method	H1	H2	unseen	seen	H2 stable
Class-identity oracle	55.0 ± 0.4	55.0 ± 0.9	56.5 ± 3.2	54.9 ± 0.9	55.0 ± 2.3
Concept-identity ref.	54.2 ± 0.4	50.7 ± 1.3	44.2 ± 3.9	51.2 ± 1.2	53.1 ± 2.2
FedAvg	44.2 ± 0.8	45.2 ± 0.7	45.0 ± 1.2	45.2 ± 0.9	47.2 ± 2.9
No memory (best refit)	50.5 ± 0.3	49.9 ± 1.1	–	–	–
FEROMA (τ=$1/K$)	48.3 ± 2.1	48.2 ± 1.0	48.1 ± 1.4	48.2 ± 1.1	53.6 ± 2.5
FedCCFA	41.8 ± 0.9	45.1 ± 1.0	46.1 ± 1.2	45.0 ± 1.1	48.5 ± 3.1
Loss matching	40.8 ± 0.3	41.4 ± 0.4	42.9 ± 2.5	41.3 ± 0.5	42.4 ± 1.5
FragMem, concept units	$\mathbf{54.9 \pm 0.5}$	52.7 ± 0.7	48.2 ± 2.4	53.0 ± 0.5	$\mathbf{54.1 \pm 2.2}$
FragMem	54.2 ± 0.5	$\mathbf{53.3 \pm 0.9}$	$\mathbf{53.5 \pm 3.5}$	$\mathbf{53.3 \pm 0.9}$	53.3 ± 2.0

(b)

Method	MLP	Linear
Class oracle	84.2 ± 0.6	83.7 ± 0.3
Concept ref.	62.7 ± 1.2	49.1 ± 2.8
FedAvg	31.6 ± 0.9	31.1 ± 1.2
FedBABU	76.4 ± 0.5	–
FedAvg + FT	80.9 ± 0.8	–
FEROMA-τ	65.6 ± 0.7	–
Concept units	62.7 ± 1.2	–
FragMem	$\mathbf{84.2 \pm 0.6}$	$\mathbf{83.6 \pm 0.3}$

H STRESS TESTS

Table 5: **Stress tests.** For each pool: FragMem − concept-identity reference (left) and class-identity oracle − FragMem (right), switch-round accuracy in points, mean and 95% t interval over seeds 44–48; "–": not run. Few labels: $n = 20m$ samples per client and round (5 per class) for identification and training; evaluation batch 400. $m = 1$ (seed 44 only, provisional-merge variant): FragMem 37.0 / 58.6 / 25.2, class oracle 61.8 / 85.0 / 63.4, concept reference 37.5 / 54.5 / 36.4. $K = 10$ did not run: with the one extra variant the generator reserves, it needs 10 derangements of a 4-label block, and only 9 exist.

	CIFAR-100		GTSRB		20 Newsgroups	
Setting	FragMem − concept	class − FragMem	FragMem − concept	class − FragMem	FragMem − concept	class − FragMem
Few labels (m=5), H2	+21.0 [+19.0, +22.9]	+2.6 [+2.1, +3.1]	+25.8 [+21.9, +29.6]	+6.3 [+5.0, +7.7]	+21.9 [+19.6, +24.2]	+4.4 [+3.1, +5.8]
Dirichlet 0.5, H2	+19.0 [+17.3, +20.6]	+2.1 [+1.1, +3.1]	+24.0 [+21.6, +26.5]	+4.6 [+3.4, +5.9]	+18.9 [+17.4, +20.5]	+1.6 [+0.2, +2.9]
Dirichlet 0.5, H1	−3.9 [−5.3, −2.6]	+4.2 [+3.0, +5.5]	−7.0 [−9.6, −4.5]	+7.8 [+5.5, +10.2]	−4.0 [−4.7, −3.3]	+3.8 [+3.6, +4.1]
Switch rate 0.1	+14.7 [+13.2, +16.2]	+0.1 [−0.0, +0.1]	+19.9 [+17.9, +21.9]	+0.3 [−0.1, +0.8]	+16.4 [+15.0, +17.7]	−0.0 [−0.1, +0.0]
Switch rate 0.5	+26.3 [+25.0, +27.6]	+0.1 [+0.1, +0.1]	+35.7 [+34.3, +37.0]	+0.3 [−0.1, +0.8]	+28.9 [+27.8, +29.9]	+0.1 [+0.0, +0.2]
K=3 fragments/block	+26.4 [+24.3, +28.5]	+0.1 [−0.0, +0.2]	+35.9 [+33.9, +37.9]	+0.3 [+0.1, +0.5]	+28.9 [+27.8, +30.1]	+0.0 [−0.0, +0.1]
48 clients	+23.4 [+21.1, +25.6]	+0.2 [+0.1, +0.2]	+31.7 [+29.0, +34.5]	+0.2 [+0.2, +0.3]	–	–
100 clients	+23.2 [+20.9, +25.4]	+0.3 [+0.2, +0.4]	+31.8 [+29.0, +34.6]	+0.3 [+0.1, +0.4]	–	–
Semantic blocks, H2	+21.9 [+19.2, +24.6]	+0.2 [+0.0, +0.4]	–	–	–	–
Semantic blocks, H1	+0.3 [+0.1, +0.5]	+0.1 [−0.0, +0.3]	–	–	–	–

The $m = 1$ run (seed 44, which we also used to diagnose this setting; seeds 45–48 not run) includes a provisional slot merge that raised CIFAR-100 from 33.1 to 37.0 and GTSRB from 45.3 to 58.6 but lowered 20NG from 30.8 to 25.2 by merging different states (purity 0.42). With 5 samples per class, FragMem minus loss matching is $-0.65\,[-1.51,\ 0.21]$, $-1.52\,[-2.98,\ -0.05]$ and $-2.07\,[-3.46,\ -0.69]$, and FragMem leads the best per-round refit by 13.4–17.3 points. Under Dirichlet 0.5, FragMem leads loss matching by 4.3–15.6 points on H2 and 6.0–16.1 on H1.

Table 6: **All baselines**, switch-round accuracy (%), mean ± 95% t half-width over seeds 44–48; "–": not run. "Earlier port (audited)" rows are kept for transparency; the re-implementations are the comparisons. FeSEM H2: only CIFAR-100 seed 44 finished. Bold: highest non-oracle mean. Names: "Oracle (class-level identity)" is the class-identity oracle, "Oracle (concept identity)" the concept-identity reference, and "Class-level memory (ours)" is FragMem.

Method	CIFAR-100 H1	CIFAR-100 H2	GTSRB H1	GTSRB H2	20 Newsgroups H1	20 Newsgroups H2	CIFAR-100 lump H1	CIFAR-100 lump H2	Comm. (× FedAvg, H2)
reference									
Oracle (class-level identity)	63.7 ± 0.3	63.2 ± 0.3	87.2 ± 0.7	86.9 ± 0.3	65.0 ± 0.3	64.5 ± 0.2	71.2 ± 0.4	71.3 ± 0.2	1.52
Oracle (concept identity)	63.2 ± 0.4	39.8 ± 2.0	85.8 ± 0.7	54.9 ± 2.8	65.1 ± 0.2	38.4 ± 2.3	70.8 ± 0.4	63.3 ± 0.5	1.52
no drift handling									
FedAvg	17.8 ± 2.0	22.9 ± 1.1	22.5 ± 1.6	30.7 ± 1.0	21.3 ± 1.9	24.7 ± 0.9	58.5 ± 0.3	59.1 ± 0.3	1.00
Local only	16.1 ± 1.6	38.4 ± 2.1	19.9 ± 2.2	53.8 ± 3.0	18.3 ± 1.8	36.4 ± 2.4	55.4 ± 0.6	62.0 ± 0.5	0.00
personalisation									
Ditto	15.6 ± 1.3	35.2 ± 0.9	17.9 ± 1.6	50.8 ± 1.8	20.0 ± 1.5	32.9 ± 1.0	–	–	1.00
APFL	17.0 ± 1.5	38.3 ± 1.9	20.9 ± 2.0	53.5 ± 2.9	20.1 ± 1.7	36.5 ± 2.1	–	–	1.00
Adaptive-FedAvg	20.0 ± 2.1	21.1 ± 0.7	25.5 ± 1.3	26.4 ± 1.8	23.1 ± 2.2	23.7 ± 1.3	–	–	1.00
clustering									
CFL	17.0 ± 1.8	21.5 ± 1.0	21.7 ± 1.5	29.3 ± 1.1	20.3 ± 1.7	23.1 ± 0.9	–	–	1.00
FeSEM	11.1 ± 1.6	35.0	13.8 ± 1.3	–	12.4 ± 1.7	–	–	–	65.00
FLUX, earlier port (audited)	16.8 ± 1.8	39.7 ± 1.5	20.0 ± 3.1	49.8 ± 2.7	19.5 ± 2.6	38.4 ± 1.6	58.0 ± 0.5	62.7 ± 0.5	2.10
FLUX, official descriptor and eps	16.6 ± 2.1	39.9 ± 1.9	20.3 ± 2.5	55.0 ± 2.7	19.7 ± 2.5	38.4 ± 2.2	–	–	1.17
drift adaptation									
Flash, earlier port (audited: equals FedAvg)	17.9 ± 2.0	22.9 ± 1.1	22.5 ± 1.6	30.6 ± 1.0	21.3 ± 1.9	24.7 ± 0.9	–	–	1.00
Flash, Algorithm 1	19.7 ± 1.8	21.1 ± 0.8	28.0 ± 2.0	30.2 ± 0.8	21.1 ± 1.8	22.1 ± 0.6	–	–	1.01
prototypes									
FedProto, earlier port (audited)	17.9 ± 2.0	19.1 ± 0.9	19.6 ± 0.7	21.2 ± 2.3	23.1 ± 2.2	24.0 ± 1.3	–	–	0.99
FedProto, local model	16.0 ± 1.6	38.4 ± 2.0	20.0 ± 2.1	53.8 ± 2.9	18.3 ± 1.7	36.4 ± 2.4	–	–	1.00
FedProto, global-prototype inference	7.1 ± 1.1	20.3 ± 1.0	8.0 ± 1.1	23.2 ± 0.8	9.0 ± 1.3	25.1 ± 0.8	–	–	1.00
profile mapping (ICLR 2026)									
FEROMA, released code	17.9 ± 2.0	23.3 ± 1.1	22.7 ± 1.7	31.9 ± 1.0	21.3 ± 2.0	25.0 ± 0.9	58.6 ± 0.3	59.2 ± 0.3	1.17
FEROMA, Eq. 4 threshold $\tau = 1/K$	19.3 ± 2.4	34.6 ± 1.4	26.2 ± 3.1	54.8 ± 1.7	22.1 ± 2.2	33.0 ± 1.6	59.4 ± 0.3	60.2 ± 0.5	1.17
class-level memory									
Class-level memory (ours)	**63.6 ± 0.4**	**63.1 ± 0.3**	**86.8 ± 0.7**	**86.6 ± 0.6**	**64.9 ± 0.3**	**64.4 ± 0.2**	**69.7 ± 0.5**	**70.0 ± 0.5**	2.03

Table 7: **Identification quality**: purity (fraction of assignments to a slot's majority state), slots per true state, states per slot; mean $\pm$ 95% t half-width. "Class-level memory (ours)" is FragMem. The "+ directional seq. test" rows report the directional variant (Appendix C), not the sequential mean test.

Method	Metric	CIFAR-100 H1	CIFAR-100 H2	GTSRB H1	GTSRB H2	20 Newsgroups H1	20 Newsgroups H2	CIFAR-100 fine (lump) H1	CIFAR-100 fine (lump) H2
Class-level memory (ours)	purity	1.000 ± 0.000	1.000 ± 0.000	0.997 ± 0.007	0.998 ± 0.003	0.999 ± 0.002	0.998 ± 0.004	0.921 ± 0.025	0.936 ± 0.012
	slots per state	1.29 ± 0.04	1.28 ± 0.06	1.18 ± 0.03	1.18 ± 0.03	1.64 ± 0.04	1.60 ± 0.06	1.24 ± 0.13	1.28 ± 0.07
	states per slot	1.01 ± 0.01	1.02 ± 0.02	1.03 ± 0.05	1.04 ± 0.06	1.02 ± 0.03	1.03 ± 0.03	1.73 ± 0.07	1.73 ± 0.10
Class-level memory, loss matching	purity	0.980 ± 0.027	0.981 ± 0.024	0.955 ± 0.019	0.946 ± 0.018	0.980 ± 0.022	0.980 ± 0.019	0.674 ± 0.010	0.668 ± 0.010
	slots per state	1.23 ± 0.05	1.23 ± 0.05	1.11 ± 0.06	1.10 ± 0.04	1.42 ± 0.03	1.42 ± 0.03	2.13 ± 0.43	2.15 ± 0.44
	states per slot	1.07 ± 0.05	1.07 ± 0.04	1.11 ± 0.04	1.14 ± 0.01	1.06 ± 0.04	1.05 ± 0.04	1.99 ± 0.05	1.97 ± 0.04
FPT, concept granularity	purity	1.000 ± 0.000	0.998 ± 0.006	1.000 ± 0.000	0.999 ± 0.002	1.000 ± 0.000	0.994 ± 0.007	1.000 ± 0.000	0.780 ± 0.045
	slots per state	1.17 ± 0.29	1.04 ± 0.01	1.13 ± 0.17	1.01 ± 0.01	2.83 ± 0.25	1.27 ± 0.04	1.35 ± 0.35	0.68 ± 0.07
	states per slot	1.00 ± 0.00	1.01 ± 0.01	1.00 ± 0.00	1.00 ± 0.01	1.00 ± 0.00	1.02 ± 0.02	1.00 ± 0.00	2.04 ± 0.21
IFCA	purity	0.728 ± 0.082	0.166 ± 0.039	0.734 ± 0.080	0.141 ± 0.043	0.633 ± 0.118	0.159 ± 0.045	0.469 ± 0.236	0.082 ± 0.032
	slots per state	0.70 ± 0.09	0.06 ± 0.02	0.70 ± 0.09	0.05 ± 0.01	0.63 ± 0.09	0.05 ± 0.02	0.45 ± 0.26	0.02 ± 0.01
	states per slot	1.44 ± 0.17	18.65 ± 4.47	1.44 ± 0.17	22.02 ± 4.13	1.65 ± 0.28	20.14 ± 6.97	2.67 ± 1.55	58.30 ± 31.91
Class-level memory + directional seq. test	purity	1.000 ± 0.000	1.000 ± 0.000	0.999 ± 0.001	1.000 ± 0.001	0.999 ± 0.001	0.999 ± 0.001	0.969 ± 0.005	0.969 ± 0.005
	slots per state	1.29 ± 0.04	1.28 ± 0.06	1.21 ± 0.04	1.20 ± 0.05	1.68 ± 0.03	1.66 ± 0.06	1.31 ± 0.13	1.34 ± 0.05
	states per slot	1.01 ± 0.01	1.01 ± 0.02	1.01 ± 0.02	1.02 ± 0.03	1.02 ± 0.02	1.03 ± 0.03	1.67 ± 0.04	1.68 ± 0.05