# IS RETRAINING ENOUGH? COORDINATED LOCAL EDITING FOR CERTIFIED VERTICAL FEDERATED UNLEARNING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Vertical federated unlearning (VFU) removes a departed party's feature block from a model jointly trained by institutions holding different features of one population. VFU faces a distinctive obstacle: feature blocks are correlated across institutions, so retraining leaves a predictable share of the departed signal in the retained encoders. Current methods edit inside the task objective and settle at the level retraining attains, and they certify model parameters rather than the embeddings, read-out and caches an auditor reads. To address these, we first establish a two-floor separation: the recoverable share splits into a task-required part and a free part whose removal locates an encoding-layer floor beneath the task-objective floor. We then design **CLEAVE**, a Coordinated Local Editing framework for Attested Vertical Erasure. Within this framework, Residual Splitting measures the joint residual from statistics the active party already holds and separates task-required from removable directions. Coordinated Local Editing then solves one block-diagonal edit that hands each retained party a local adapter, so no party combines features held by others. In addition, State Purge and Certification refits the read-out, purges the requested state and certifies leakage for a declared decoder class and audited interface. Across three datasets, four deletion granularities and thirteen published baselines, **CLEAVE** most consistently lowers the recovered share and the issued level while retaining utility close to retraining in one communication round. For example, on Letter under party unlearning, **CLEAVE** lowers the recovered share from $0.727$ under retraining to $0.331$ at higher utility.

## 1 INTRODUCTION

Vertical federated learning (VFL) (Yang et al., 2019; Liu et al., 2024) trains a shared model across institutions holding different feature blocks of the same population. Passive parties transmit embeddings from private bottom models; the active party holds the labels and top model (Vepakomma et al., 2018). When an agreement expires or authorization is revoked, vertical federated unlearning must remove the departed block's influence (Tae & Chan, 2025). Its bottom model leaves with it, while the federation retains the other encoders, the fitted top model and cached training embeddings.

As illustrated in Figure 1, the reference procedure retrains the federation from scratch on the retained blocks, and every published vertical unlearning method is measured against it (Wu et al., 2026; Varshney et al., 2025; Wang et al., 2024), reviewed in Appendix A. Retraining, however, is attribution-clean rather than information-clean: institutional feature blocks are correlated (Wu et al., 2024), and retraining on the retained blocks need not remove their predictable share of the departed signal. Measuring this share, the residual index, finds published methods (Han et al., 2025; Tang et al., 2026; Jiang et al., 2025) clustered on it: their task-oriented edits leave much of the measured reconstructible share in the retained representation. This raises the first question: ***I) How much of the departed unit's remaining signal does the task actually require, and how much is removable at no utility cost?***

Leakage remains a joint property of all retained blocks, whereas each adapter acts before its holder transmits an embedding. Deployable edits are therefore block diagonal, with one matrix per retained passive party. This raises the second question: ***II) How can retained passive parties, each editing its own encoding map, approach the erasure of a joint editor, and what does that locality cost them?***



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.

A deletion that regulators and counterparties accept must also be checkable (Sommer et al., 2022; Thudi et al., 2022). The state an auditor faces spans fresh embeddings (Luo et al., 2021; Fu et al., 2022), the read-out layer, and the caches accumulated during training, and a guarantee stated over model parameters alone (Guo et al., 2020; Sekhari et al., 2021) leaves every one of these channels unexamined, while the correlated signal the task itself uses sets a floor beneath which no deletion reads. This raises the third question: ***III) How can the federation issue a certificate that names the level it achieved and the lowest level attainable at the required utility, and survives independent verification on the complete post-deletion state against a declared decoder class and audited interface?***

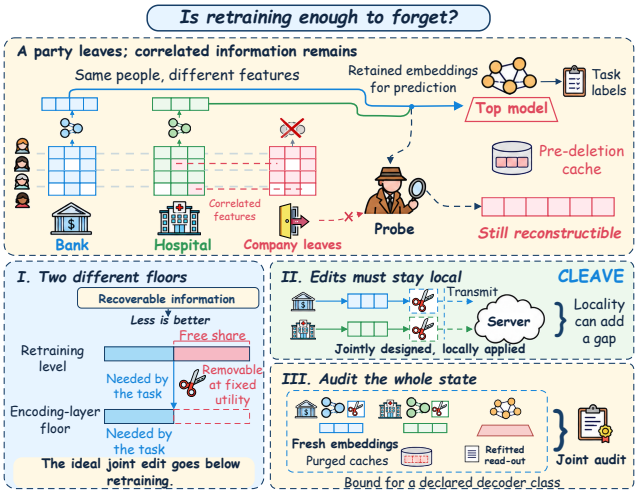



Figure 1: **Problem Illustration.** Correlated feature blocks let the retained parties reconstruct a share of the departed block that retraining preserves. **I)** Task-objective floor: the task requires part of that share, and a free share serves reconstruction alone. **II)** Federated locality: the joint editor that removes the free share becomes one adapter per retained party. **III)** Post-deletion state: an auditor recomputes the certificate on the caches, fresh embeddings and read-out layer.

We address these challenges with **CLEAVE**: leakage is measured jointly, and each party applies its edit locally. For challenge **I)**, **Residual Splitting (RS)** separates task-required and removable reconstruction directions using second-order statistics already held by the active party. For challenge **II)**, **Coordinated Local Editing (CLE)** constructs block-local adapters from joint statistics in one communication round. The federation surcharge measures the minimum locality cost; the deployed edit's excess is accounted for separately. For challenge **III)**, **State Purge and Certification (SPC)** refits the read-out, purges the requested state, and reports achieved leakage with the decoder class, audited interface and benchmark status. An auditor checks the committed release using fresh samples. Party, feature, sample and label requests enter through their footprints in the same statistics. Our main contributions are:

❶ ***Two Floors.*** We identify and formalize a two-floor separation for vertical federated unlearning: the departed block's recoverable share splits into a task-required part, whose level the task-objective floor fixes and where retraining and published task-oriented edits settle, and a free part that serves reconstruction alone, whose removal locates an encoding-layer floor beneath the first. The separation is falsifiable: it predicts that any edit confined to the task objective stays on the first floor whatever its mechanism, while an edit reaching the retained encoding maps moves below it at held utility, and our controlled and real vertical splits confirm this prediction through the residual index.

❷ ***Local Editing Protocol.*** We introduce **CLEAVE**, realizing the separation through three coordinated mechanisms: RS measures the joint residual of the deleted unit from second-order statistics the active party already holds and separates task-required from removable directions; CLE solves one block-diagonal edit that hands each retained passive party a local adapter, deploying the joint editor without any party combining features held by others, and prices the minimum cost of that locality, the federation surcharge, apart from solver excess; and SPC refits the read-out, purges the requested state and issues a leakage level for a declared decoder class and audited interface. We back the protocol with linear-decoding floors, a locality-surcharge characterization and a finite-sample certificate for a fixed declared class.

❸ ***Empirical Verification.*** Experiments on Letter, Fashion-MNIST and CIFAR-100 across party, feature, sample and label deletion show that **CLEAVE** most consistently lowers the recovered share and the issued leakage level among thirteen published baselines while retaining utility close to retraining and deploying in one communication round, with task utility, recovered share, membership inference, attack success, the issued level and communication cost characterizing different aspects of erasure over five independent runs. Independent audits recompute the issued

level on the committed release, and nonlinear attacks quantify recovery beyond the declared linear class.

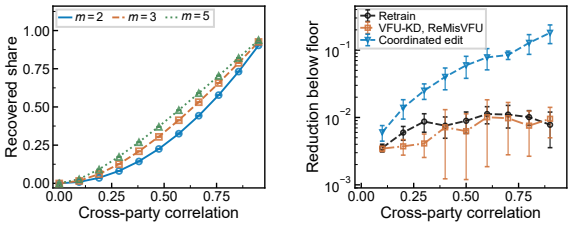
## 2 PRELIMINARIES

**Vertical federated learning.** We consider $m$ institutions holding different feature blocks of the same population. The active party holds labels $y$ and top model $g_\theta$; each passive party holds private features $X_c$ and bottom model $h_c$. Passive parties transmit cut-layer embeddings $H_c$, which the active party concatenates for prediction. Joint training solves

$$\min_{\theta,\,\{h_c\}} \frac{1}{n}\sum_{i=1}^{n} \ell\Big(g_\theta\big(h_1(x_{1,i}),\ldots,h_{m-1}(x_{m-1,i})\big),\ y_i\Big), \tag{1}$$

where $\ell$ is cross-entropy and $x_{c,i}$ is party $c$'s feature block for sample $i$. Training transmits embeddings forward and gradients backward. Let $k$ denote the departing party, $X_{-k}$ the retained features, and $Z_k$ its cached embeddings at the active party. The objective preserves information useful for predicting $y$, including signal shared across correlated blocks. Consequently, retained features can reveal information about $X_k$ after party $k$ leaves.

**Unlearning problem.** A request names a party, features, samples or labels. We use party deletion as the running example; the remaining units enter through their corresponding footprints. When party $k$ leaves, its features and bottom model depart, while the federation retains the other bottom models, top model and training cache. Retraining on $X_{-k}$ removes direct dependence on $X_k$ but can preserve its predictable signal. Attribution unlearning addresses that direct dependence; informational erasure additionally constrains reconstruction from the post-deletion state. We measure this goal by declaring a footprint, such as $Z_k$, and a decoder class. The residual index is the footprint energy recoverable by that class. Our target is the portion that reconstruction uses and the retained task does not require.




(a) Predicted floor. (b) Distance to floor.

Figure 2: Reconstruction versus cross-party correlation at a common utility target.

**Threat model.** The guarantee is written against an adversary reading the state the protocol leaves: the retained bottom models with adapters installed, the embeddings that composition emits, the read-out layer, and whatever survives the purge. Retained parties run the protocol they commit to, so the deployed encoder is a bottom model followed by its adapter. The declared interface exposes edited outputs; access to private raw features or unedited activations by bypassing an adapter is outside this guarantee. Every surviving cache and read-out channel available through that interface must be included in the audit.

**Desiderata.** We require four properties. (i) The deployer specifies leakage target $\varepsilon$ and utility requirement $\tau$; the protocol checks both before accepting an edit. (ii) The declaration identifies the decoder class and every surviving channel exposed through the audited interface, including embeddings, read-out and caches. (iii) The protocol reports achieved leakage beside the same-utility benchmark, states how that benchmark was computed, and supplies the information needed for an independent audit. (iv) Deployment takes one communication round using statistics already available during VFL training. Each retained party therefore edits its encoding before transmitting new embeddings.

**Observation 1**

*Retraining does not remove what the retained parties can reconstruct. As cross-party correlation grows, the share of the departed block recoverable from the embeddings rises with the residual index, and retraining settles precisely at that level rather than at any point strictly beneath it.*

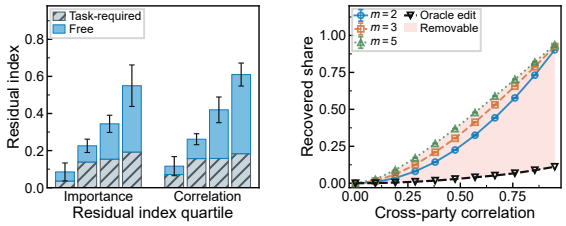
We sweep cross-party correlation on a vertical split with a redundancy knob and measure how much of the departed signal a decoder recovers from the post-deletion state. Figure 2a shows retraining tracking the residual index rather than falling to zero, and Figure 2b measures how far below it

published methods reach at a common utility bar: each edits inside the task objective and stays within two hundredths of it. For a fixed retained representation, any mechanism leaving its embeddings observable retains their reconstructible share; the following theory formalizes this statement for linear decoders.

**Observation 2**

*Part of what remains recoverable is unnecessary for the task. An edit reaching the encoding maps removes that part while utility is held fixed, which places a second level beneath the one retraining attains, reachable only through an edit of the retained parties' own encoding maps.*

Observation 1 leaves open how much of the residual the task requires. Figure 3a decomposes it on Adult federations built with VertiBench importance and correlation splits: alongside the part the task requires, a distinct share is reconstructible while contributing nothing the label needs, and that share exceeds half of the residual for two thirds of the departing parties. Grouping those parties by the residual they leave separates the two parts further, since in both split families the task-required part settles below a fifth while the free part carries most of the growth. Figure 3b then removes that share on an instance where the optimum is computable, editing the embeddings jointly with utility held at $\tau$, and the residual moves below the curve of Figure 2a at unchanged utility, which locates a second floor beneath the first. That second level barely moves as parties are added while the level retraining attains rises with each retained block, so every increment belongs to the removable part. That joint edit is an oracle, since it mixes across parties, whereas a deployable protocol reaches its target with each retained passive party editing only its own block. Closing that distance, and pricing the block-diagonal restriction, is the problem we take up.




(a) Free share, Adult. (b) Oracle joint edit.

Figure 3: Free share on Adult VertiBench federations, and the residual an oracle edit reaches at held utility.

## 3 METHODOLOGY

### 3.1 FRAMEWORK OVERVIEW

Motivated by Observations 1–2, **CLEAVE** edits retained encodings to reduce joint reconstruction while each party acts locally. All four deletion units enter through their footprints in second-order statistics. Three components form one round (Figure 4): **I)** Residual Splitting (RS, Section 3.2) separates task-required and removable directions; **II)** Coordinated Local Editing (CLE, Section 3.3) constructs block-local adapters applied before transmission; **III)** State Purge and Certification (SPC, Section 3.4) refits the read-out, purges the requested state, and audits declared-class leakage. Algorithm 1 states the protocol.

### 3.2 RESIDUAL SPLITTING (RS)

**Motivation.** A deleted unit's information spreads across the retained parties' joint embeddings, so an embedding that looks safe alone can still reconstruct it with the others, and per-party measurement underestimates the residual and produces an incomplete edit. RS measures the joint residual once at the active party, from embeddings, caches and labels VFL already provides.

**Splitting the residual.** RS measures recoverability in a fixed decoder dictionary, optionally formed from $D$ random features per block and a public seed. The displayed trace formulas use unrestricted linear least squares, including norm-bounded decoding when the bound is inactive; an active bound uses the prediction-risk definition in Eq. (48). Write $H$ for retained embeddings, $Z$ for the requested footprint (the cached embedding $Z_k$ for party $k$), $\Sigma$ for empirical covariances, and $\hat{Z}$ for the best

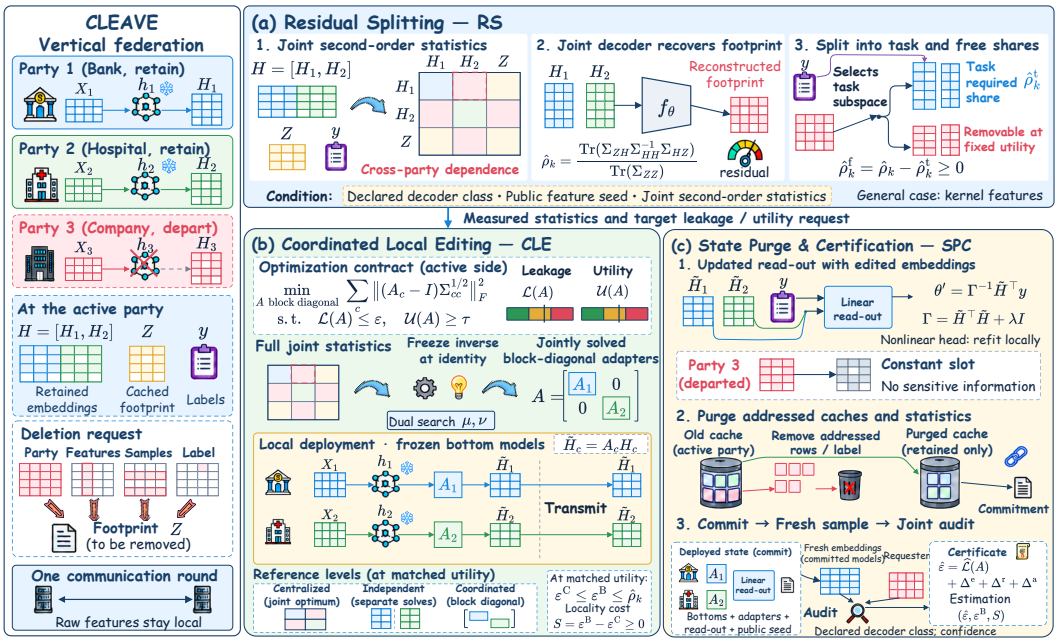



Figure 4: **Overview of CLEAVE.** RS measures the joint residual of the requested unit at the active party and splits off the free share the task does without. CLE solves one leakage-constrained block-diagonal edit problem and returns one adapter to each retained passive party, installed behind its own cut layer. SPC absorbs the edit into the read-out layer, purges the state the unit addresses, and issues the certificate, all within the single communication round.

declared decoder. The residual index is the fraction of footprint energy recoverable from $H$:

$$\hat{\rho}_k \;=\; \frac{\mathbb{E}\big\|\hat{Z}(H)\big\|^2}{\mathbb{E}\|Z\|^2} \;=\; \frac{\mathrm{Tr}\left(\Sigma_{ZH}\,\Sigma_{HH}^{-1}\,\Sigma_{HZ}\right)}{\mathrm{Tr}(\Sigma_{ZZ})}, \qquad \hat{Z}(H) \;=\; \Sigma_{ZH}\,\Sigma_{HH}^{-1}H, \tag{2}$$

where the second statement is the least-squares decoder of $Z$ from $H$ whose energy the first one reads. We next separate the recoverable information the prediction task requires. Let $\Pi$ be the orthogonal projector onto the column space of $\Sigma_{HH}^{-1}\Sigma_{Hy}$, the coefficients of the best linear label predictor, and let $\hat{\rho}_k^{\mathrm{t}}$ denote the residual index of a decoder seeing only $\Pi H$, so the whitening operator $\Sigma_{HH}^{-1}$ becomes its task-restricted form $P$. The difference between the full residual and the task-required residual defines the free share:

$$\hat{\rho}_k^{\mathrm{f}} \;=\; \hat{\rho}_k - \hat{\rho}_k^{\mathrm{t}} \;=\; \frac{\mathrm{Tr}\left(\Sigma_{ZH}\big(\Sigma_{HH}^{-1} - P\big)\Sigma_{HZ}\right)}{\mathrm{Tr}(\Sigma_{ZZ})} \;\geq\; 0, \qquad P \;=\; \Pi(\Pi\Sigma_{HH}\Pi)^{+}\Pi, \tag{3}$$

where $P$ is the task-restricted whitening operator, so $\hat{\rho}_k^{\mathrm{t}}$ is Eq. (2) with $P$ in place of $\Sigma_{HH}^{-1}$. This free share is removable by a centralized linear edit at unchanged second-order utility. In whitened coordinates it is the energy outside the label-predictive subspace, making $\Sigma_{HH}^{-1} - P$ positive semidefinite; Theorem 2 proves the equivalence and attainability. The quantities use empirical second-order statistics already held by the active party. Lemma 4 separately bounds the finite-sample error of the declared-class audit.

**Footprints of the four units.** The definition of $Z$ follows the granularity. **(i)** Party: the cached embedding $Z_k$ of the departing party. **(ii)** Features: the difference between the requesting party's cached embeddings and those generated from the reduced feature block, which reaches the active party through ordinary forward passes while within-block reconstruction is measured locally. **(iii)** Label: the indicator of the deleted class, so $\Sigma_{HZ}$ is the corresponding column of $\Sigma_{Hy}$; rebuilding the task projector from the retained labels removes directions used only by the deleted class. **(iv)** Samples: the indicator of the deleted rows, whose cross-party structure the active party reads off its per-sample embedding cache. Every footprint is computed at the active party or by the holder of the requested unit, so no granularity adds a communication round.

### 3.3 COORDINATED LOCAL EDITING (CLE)

**Motivation.** RS identifies the removable part of the residual, and CLE removes it within the scope the certificate covers. Since that scope includes transmitted embeddings as well as retained caches, the edit applies before an embedding leaves its holder, which makes the feasible edit block diagonal, one matrix per party's own block. A centralized erasure method mixes features across parties, so its transformation exists only after transmission and cannot be applied at the point the certificate covers.

**The edit problem.** Write $\mathcal{C}$ for the editing parties: the passive parties the federation retains, joined at feature-level deletion by the requesting party. Each editing party $c \in \mathcal{C}$ applies an edit matrix $A_c$ to its cut-layer embedding, and $A$ denotes the block-diagonal assembly of all local edits. After editing, the optimal decoder observes $AH$ in place of $H$, and the residual leakage is

$$\mathcal{L}(A) \;=\; \frac{\mathbb{E}\big\|\hat{Z}(AH)\big\|^2}{\mathbb{E}\|Z\|^2} \;=\; \frac{\mathrm{Tr}\left(\Sigma_{ZH}\, Q_A\, \Sigma_{HZ}\right)}{\mathrm{Tr}(\Sigma_{ZZ})}, \qquad Q_A \;=\; A^\top \big(A\Sigma_{HH}A^\top\big)^{+} A, \tag{4}$$

where $Q_A$ is the whitening operator of the edited embeddings. The utility functional puts the label in the place of the footprint:

$$\mathcal{U}(A) \;=\; \frac{\mathbb{E}\big\|\hat{y}(AH)\big\|^2}{\mathbb{E}\|y\|^2} \;=\; \frac{\mathrm{Tr}\left(\Sigma_{yH}\, Q_A\, \Sigma_{Hy}\right)}{\mathrm{Tr}(\Sigma_{yy})}, \tag{5}$$

where $\hat{y}$ denotes the best predictor of $y$ in the declared class and $\Sigma_{yy}$ the label covariance. At the identity edit, $\mathcal{L}$ returns the residual index and $\mathcal{U}$ the original label-prediction energy. Given a requested leakage level $\varepsilon$ and utility level $\tau$, CLE targets

$$\min_{\substack{\{A_c\}_{c\in\mathcal{C}} \\ \mathcal{L}(A)\le\varepsilon,\; \mathcal{U}(A)\ge\tau}} \; \sum_{c\in\mathcal{C}} \Big\| (A_c - I)\, \Sigma_{cc}^{1/2} \Big\|_F^2, \tag{6}$$

where $\Sigma_{cc}$ is the $c$-th diagonal block of $\Sigma_{HH}$ and $A$ the block-diagonal assembly $\mathrm{blkdiag}\left((A_c)_{c\in\mathcal{C}}\right)$ entering $\mathcal{L}$ and $\mathcal{U}$. Weighting the change by $\Sigma_{cc}^{1/2}$ measures the edit in the geometry induced by the party's data rather than treating all embedding directions equally. A nonlinear residual adapter can instead take the form $H_c - W_c\varphi_c(H_c)$ and is audited directly in the declared decoder class. The displayed projector analysis concerns edits linear in the chosen representation; the identity feature map recovers $A_c = I - W_c$.

**One linear system.** The pseudo-inverse inside $Q_A$ makes Eq. (6) nonquadratic. To construct a candidate, CLE freezes this term at the identity edit, replacing $Q_A$ with $A^\top\Sigma_{HH}^{-1}A$. Eq. (41) bounds the discrepancy when this surrogate is used for certification. Under this approximation, the leakage and utility functionals become trace forms in a leakage Gram matrix $M$ and a task Gram matrix $N$. For multipliers $\mu, \nu \ge 0$ absorbing the footprint and label normalizers, the block-diagonal stationarity condition becomes

$$(A_c - I)\,\Sigma_{cc} \;+\; \sum_b \left(\Sigma_{HH}^{-1}\right)_{cb} A_b \left(\mu M_{bc} - \nu N_{bc}\right) \;=\; 0, \tag{7}$$

for each editing party $c$, with $M = \Sigma_{HZ}\Sigma_{ZH}$ and $N = \Sigma_{Hy}\Sigma_{yH}$. Each multiplier pair gives one linear system; a two-dimensional search retains candidates meeting both evaluated constraints and labels the remaining targets unmet. Positive semidefiniteness, feasibility and complementary slackness certify surrogate optimality (Appendix E.5). Cross-party terms with $b \neq c$ coordinate the edits through joint statistics. Each party receives its own $A_c$ and installs it after the cut layer, leaving the bottom model unchanged.

**Three ways to apply an edit.** We compare **(i)** centralized editing, **(ii)** independent local editing retaining only $b = c$ in Eq. (7), and **(iii)** coordinated block-local editing. The minimum locality surcharge at the same utility is

$$S \;=\; \min_{\substack{A\in\mathcal{A} \\ \mathcal{U}(A)\ge\tau}} \mathcal{L}(A) \;-\; \min_{\mathcal{U}(A)\ge\tau} \mathcal{L}(A) \;=\; \varepsilon^{\mathrm{B}} - \varepsilon^{\mathrm{C}} \;\ge\; 0, \tag{8}$$

For a deployed feasible $A$, the total excess over centralized editing is $\mathcal{L}(A) - \varepsilon^{\mathrm{C}} = S + g_A$, where $g_A = \mathcal{L}(A) - \varepsilon^{\mathrm{B}} \ge 0$ is solver excess. Theorem 3 gives an exact-utility construction for correlated blocks; Figure 5d reports numerical candidate values at relaxed utility, rather than these exact floors.

### 3.4 State Purge and Certification (SPC)

**Motivation.** The read-out and caches retain state from before deletion. SPC refits the read-out, purges the requested records, and binds the certificate to the surviving channels exposed by the declared interface.

**Absorption and purge.** Let $\tilde{H}$ denote the edited sample matrix before any row deletion, let $\Gamma$ denote their regularized Gram matrix $\tilde{H}^\top \tilde{H} + \lambda I$, and let $\mathcal{I}$ denote a deleted set of samples. The active party absorbs the embedding edit by re-solving its read-out layer in closed form:

$$\theta' \;=\; \arg\min_{\theta} \; \big\|\tilde{H}\theta - y\big\|^2 + \lambda\|\theta\|^2 \;=\; \Gamma^{-1}\tilde{H}^\top y. \tag{9}$$

This operation is local to the active party and requires no cross-party communication. For party-level deletion, the input slot of the departing party is fixed to a constant rather than replaced with an estimate from the retained features. Such an estimate could reintroduce reconstructible signal through the removed slot. For sample-level deletion, the active party removes the deleted rows from the same objective. The updated read-out is

$$\theta_{-\mathcal{I}} \;=\; \theta' + \Gamma^{-1}\tilde{H}_{\mathcal{I}}^\top \big(I - \tilde{H}_{\mathcal{I}}\Gamma^{-1}\tilde{H}_{\mathcal{I}}^\top\big)^{-1}\big(\tilde{H}_{\mathcal{I}}\theta' - y_{\mathcal{I}}\big), \tag{10}$$

where $\tilde{H}_{\mathcal{I}}$ and $y_{\mathcal{I}}$ are the deleted rows of the edited embeddings and labels. Eq. (10) equals the read-out refitted on the retained rows alone and computes it from $\theta'$ through a matrix solve of size $|\mathcal{I}|$. This is the classical exact downdate for a linear read-out and appears here as the sample-level case of the same protocol. The appendix writes the second-order utility loss as the label energy removed in whitened coordinates. An edit whose whitened row space contains those label directions preserves this utility; Eq. (3) identifies its removable share. These geometric benchmarks concern the encoding layer; changing the read-out alone does not change leakage from the embeddings. Eq. (9) and Eq. (10) are closed form for a linear read-out, and a nonlinear head is absorbed by refitting it on the edited embeddings at the active party, which keeps the single round.

For party- or feature-level deletion, the active party removes $Z_k$ and all statistics derived from it, keeping only a commitment for verification. For label-level deletion, it re-solves the read-out on the retained labels and reconstructs $\Pi$ without the deleted class. For sample-level deletion, every cache removes the deleted rows and the active party applies Eq. (10).

**Certificate assembly.** The issued level adds three slacks to the held-out estimate of the leakage, and the appendix shows that it then upper-bounds the leakage of the declared class:

$$\hat{\varepsilon} \;=\; \widehat{\mathcal{L}}(A) + \Delta^{\mathrm{e}} + \Delta^{\mathrm{r}} + \Delta^{\mathrm{a}}, \qquad \mathbb{P}\big[\mathcal{L}(A) \le \hat{\varepsilon}\big] \;\ge\; 1 - \delta, \tag{11}$$

where $\widehat{\mathcal{L}}(A)$ is evaluated on $n'$ independent audit samples in the declared class. Lemma 4 supplies the sampling term; exact evaluation sets $\Delta^{\mathrm{r}} = 0$, and a fixed finite dictionary sets $\Delta^{\mathrm{a}} = 0$. A kernel-class extension uses the lemma's uniform approximation event. The certificate names the footprint, decoder class and audited channels, together with the computation status of its block benchmark. Joint commitments fix the bottom models, adapters, read-out and feature seed before sampling. An auditor then recomputes the reading from fresh embeddings and footprint records supplied by the requesting party. Surviving channels enter this audit explicitly or satisfy the same-class factorization in Theorem 5. If the identity meets both targets, it attains zero edit cost.

**Cost.** Retraining spends $R$ full rounds of vertical training, exchanging embeddings and gradients in every one. **CLEAVE** sends each editing party one matrix of size $d_c \times D$ with the public seed of the feature map, solves one linear system at the active party, and costs one feature-map evaluation and one matrix multiplication at inference, drawing on no cross-party quantity beyond what the active party already observes during normal VFL training.

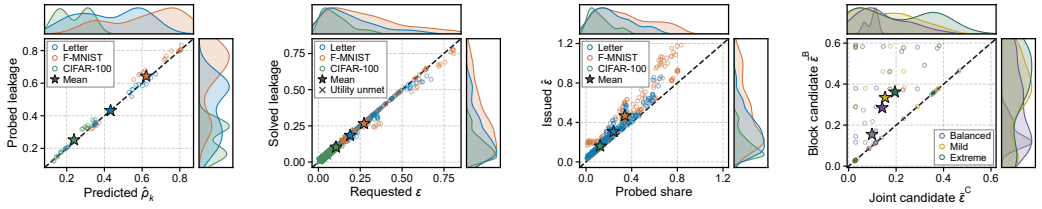
## 4 Experiments

### 4.1 Experimental Setup

We evaluate on Letter, Fashion-MNIST and CIFAR-100 across party, feature, sample and label deletion, comparing with thirteen published baselines and full retraining. Each main benchmark

Table 1: Main comparison on Letter across the four deletion granularities, mean over five seeds. ASR reads a trigger planted inside the deleted unit: a retrain reproduces the share of that unit the retained blocks predict, so it keeps answering the trigger, which is the gap a local edit closes. Best in **red**, runner-up in blue among the compared methods; *Retrain* does not participate in ranking.

| Method | Util.↑ | Share↓ | MIA↓ | ASR↓ | $\hat{\varepsilon}$↓ | Comm (MB)↓ |
|---|---|---|---|---|---|---|
| *Party Unlearning* | | | | | | |
| ReMisVFU [AAAI'26] (Wu et al., 2026) | 0.626(±0.031) | 0.591(±0.016) | 0.018(±0.004) | 0.086(±0.008) | 0.714(±0.021) | 78.1(±2.6) |
| VFU-KD [PoPETs'25] (Varshney et al., 2025) | 0.681(±0.004) | 0.686(±0.020) | **0.012**(±0.006) | 0.143(±0.019) | 0.802(±0.033) | 0.084(±0.012) |
| FedR2S [ACM TOIT'24] (Wang et al., 2024) | 0.690(±0.009) | 0.706(±0.018) | 0.014(±0.007) | 0.169(±0.035) | 0.768(±0.028) | 50.8(±1.7) |
| LVFUS [IEEE TNSE'26] (Tang et al., 2026) | 0.681(±0.014) | 0.699(±0.016) | 0.014(±0.005) | 0.149(±0.044) | 0.744(±0.039) | 38.0(±1.4) |
| **CLEAVE** | **0.739**(±0.009) | **0.331**(±0.014) | 0.014(±0.004) | **0.053**(±0.007) | **0.347**(±0.014) | **0.023**(±0.005) |
| *Retrain* | 0.747(±0.002) | 0.727(±0.010) | 0.012(±0.004) | 0.115(±0.008) | 0.889(±0.026) | 78.1(±2.4) |
| *Feature Unlearning* | | | | | | |
| VFU-KD [PoPETs'25] (Varshney et al., 2025) | 0.661(±0.036) | 0.390(±0.013) | 0.012(±0.006) | 0.766(±0.347) | 0.563(±0.030) | 0.066(±0.011) |
| Cer-FeaUn [IEEE TMC'26] (Wang et al., 2026) | 0.750(±0.015) | 0.402(±0.010) | **0.007**(±0.004) | 0.997(±0.001) | 0.596(±0.021) | 5.9(±0.4) |
| LVFUS [IEEE TNSE'26] (Tang et al., 2026) | 0.793(±0.012) | 0.421(±0.004) | 0.015(±0.007) | 0.995(±0.002) | 0.677(±0.034) | 62.3(±6.1) |
| Pan et al. [IEEE TMC'25] (Pan et al., 2025) | 0.803(±0.005) | 0.408(±0.004) | 0.009(±0.002) | 0.999(±0.004) | 0.641(±0.018) | 1.0(±0.2) |
| Ferrari [NeurIPS'24] (Gu et al., 2024) | 0.141(±0.110) | 0.681(±0.168) | 0.012(±0.006) | 0.479(±0.418) | 0.734(±0.052) | 0.049(±0.009) |
| **CLEAVE** | **0.833**(±0.005) | **0.209**(±0.010) | **0.007**(±0.003) | **0.004**(±0.002) | **0.256**(±0.011) | **0.020**(±0.004) |
| *Retrain* | 0.841(±0.005) | 0.444(±0.004) | 0.011(±0.005) | 0.996(±0.002) | 0.708(±0.031) | 117.2(±3.4) |
| *Sample Unlearning* | | | | | | |
| VFU-GA [PoPETs'25] (Varshney et al., 2025) | 0.844(±0.004) | 0.213(±0.023) | 0.014(±0.004) | 0.789(±0.063) | 0.382(±0.022) | 0.052(±0.008) |
| Han et al. [IEEE TSC'25] (Han et al., 2025) | 0.532(±0.042) | 0.215(±0.010) | 0.029(±0.024) | 0.006(±0.003) | 0.337(±0.026) | 30.8(±1.5) |
| FedORA [IEEE TIFS'25] (Jiang et al., 2025) | 0.700(±0.014) | 0.213(±0.022) | 0.038(±0.024) | 0.038(±0.023) | 0.374(±0.025) | 36.3(±1.4) |
| LVFUS [IEEE TNSE'26] (Tang et al., 2026) | 0.759(±0.007) | 0.215(±0.011) | 0.018(±0.007) | 0.005(±0.007) | 0.343(±0.019) | 59.3(±5.8) |
| SCRUB [NeurIPS'23] (Kurmanji et al., 2023) | 0.547(±0.002) | 0.088(±0.002) | 0.008(±0.005) | 0.011(±0.004) | 0.241(±0.018) | 56.6(±2.2) |
| SalUn [ICLR'24] (Fan et al., 2024) | 0.827(±0.004) | 0.237(±0.024) | 0.040(±0.016) | 0.012(±0.014) | 0.356(±0.020) | 58.6(±2.0) |
| **CLEAVE** | **0.848**(±0.004) | **0.079**(±0.009) | **0.006**(±0.003) | **0.004**(±0.002) | **0.108**(±0.011) | **0.022**(±0.004) |
| *Retrain* | 0.856(±0.005) | 0.210(±0.034) | 0.014(±0.013) | 0.025(±0.013) | 0.328(±0.021) | 111.5(±3.1) |
| *Label Unlearning* | | | | | | |
| LUV [ICLR'26] (Gu et al., 2026) | 0.276(±0.280) | 0.598(±0.040) | 0.060(±0.076) | 0.649(±0.091) | 0.781(±0.045) | 4.3(±0.4) |
| FedORA [IEEE TIFS'25] (Jiang et al., 2025) | 0.666(±0.030) | 0.578(±0.044) | 0.113(±0.054) | 0.012(±0.024) | 0.709(±0.039) | 34.7(±1.5) |
| SalUn [ICLR'24] (Fan et al., 2024) | 0.811(±0.004) | 0.550(±0.032) | 0.256(±0.100) | 0.007(±0.003) | 0.668(±0.034) | 58.6(±2.0) |
| **CLEAVE** | **0.817**(±0.004) | **0.317**(±0.023) | **0.031**(±0.010) | **0.003**(±0.001) | **0.359**(±0.011) | **0.022**(±0.004) |
| *Retrain* | 0.825(±0.003) | 0.516(±0.049) | 0.270(±0.111) | 0.004(±0.002) | 0.642(±0.031) | 112.6(±3.3) |




(a) Residual index. (b) Feasibility. (c) Soundness. (d) Locality candidates.

Figure 5: Residual, constraint and audit diagnostics. Crosses in (b) mark unmet utility targets; (d) compares best feasible candidates from joint and block-local searches.

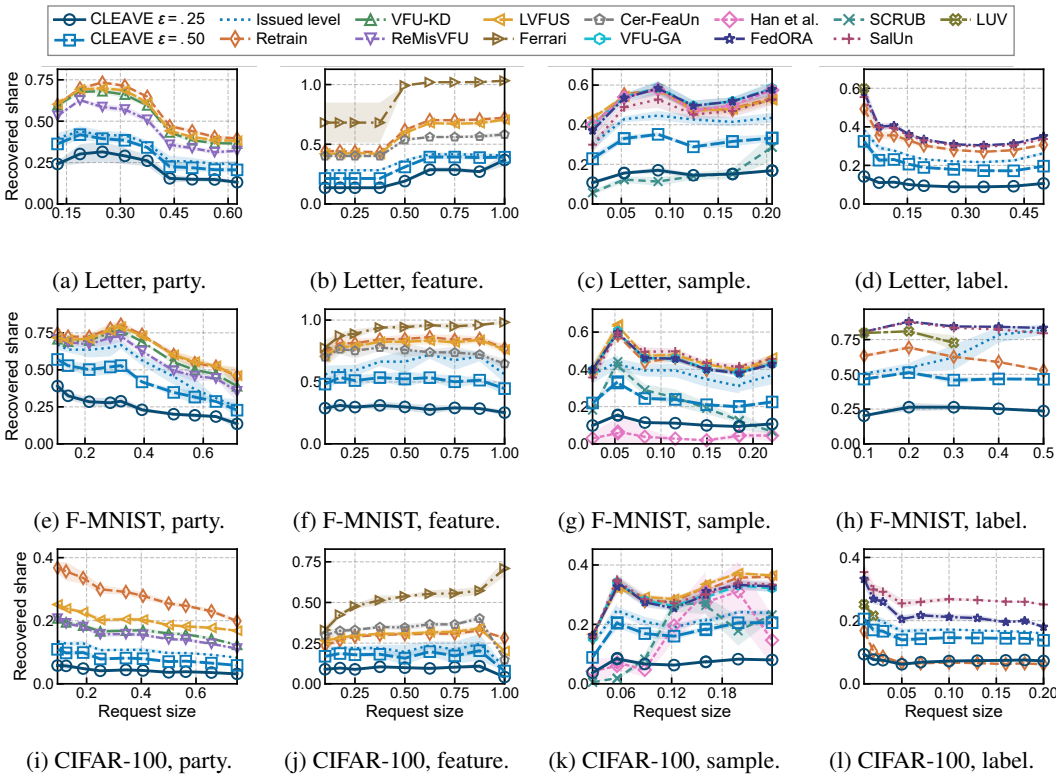
experiment is repeated independently five times; tables report the mean and standard deviation across runs. We pair reconstruction with task utility and communication cost. Appendices C.2–C.5 specify the data partitions, architectures, hyperparameters and evaluation protocols.

### 4.2 Results

Table 1 shows that **CLEAVE** reduces recovery of the requested footprint while retaining predictive utility close to retraining. Its communication cost reflects the one-round adapter deployment. The Fashion-MNIST and CIFAR-100 comparisons in Tables 2 and 3 extend this evaluation to image representations.

Table 4 attributes the largest reconstruction reduction to coordination. Absorption aligns the read-out with edited embeddings, while purge addresses the requested cached state. The utility columns distinguish selective suppression of the footprint from degradation of the retained task.

Figure 5 connects the construction to measured residuals, constraint satisfaction and audit margins. Panel (b) distinguishes reaching the leakage target from also preserving the required utility. Panel (d) compares feasible candidates found at the same relaxed utility bar; the joint search includes the block-local candidates. The exact-utility floors and the decomposition of deployed excess into




(a) Letter, party. (b) Letter, feature. (c) Letter, sample. (d) Letter, label.

(e) F-MNIST, party. (f) F-MNIST, feature. (g) F-MNIST, sample. (h) F-MNIST, label.

(i) CIFAR-100, party. (j) CIFAR-100, feature. (k) CIFAR-100, sample. (l) CIFAR-100, label.

Figure 6: Measured recovery and issued levels across deletion-request sizes.

locality cost and solver excess are given by Theorem 3 and Eq. (61). Figure 6 shows distinct recovery trajectories for the two requested levels, with issued curves accompanying the edited releases as requests grow. Further sweeps and solver diagnostics appear in Appendix B.

The LEACE and adversarial controls in Appendix D.1 use shared CE-head training and validation utility targets. Their MLP recovery curves, paired with test accuracy in Figure 13, compare editing mechanisms at different utility requirements. On Letter, the separation at relaxed utility narrows as the task requirement tightens; the changing ordering on Fashion-MNIST links the outcome to the dataset and selected subspace. Reading each attack score alongside task accuracy identifies operating points that suppress recovery while retaining a useful predictor. Table 6 separately quantifies recovery beyond a declared linear class and after changing the read-out. The nonlinear controls thus measure information accessible beyond the linear declaration.

## 5 Conclusion

**CLEAVE** makes joint reconstruction a direct target of vertical federated unlearning. It separates task-required and removable information, coordinates locally deployed adapters, and completes publication in one communication round. The analysis distinguishes the minimum locality surcharge from solver excess, while the experiments pair reconstruction with retained utility and stronger-decoder attacks. Across three datasets, four deletion granularities and thirteen published baselines, **CLEAVE** most consistently lowers the recovered share and the issued level while keeping utility close to retraining. On Letter under party unlearning it lowers the recovered share from $0.727$ under retraining to $0.331$ at higher utility, with the smallest communication cost of every compared method. An outside auditor can recompute the declared-class reading on the committed interface using fresh samples. The resulting certificate makes the measured target, audited state and decoder class explicit parts of the deletion claim.

## AI USE STATEMENT

In this work, we used generative AI tools for proposing and refining hypotheses, assisting in the writing of proofs, designing or providing feedback on research methodology or experiments, cleaning and reformatting datasets, and interpreting results. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, implementing methods, or assisting with translation, and supporting qualitative and thematic data analysis is not applicable to this work. Additionally, we used generative AI tools for discovering research topics, plotting the figures that report experimental results, searching for and summarizing relevant literature, drafting the notation table in the appendix, editing the paper for readability, and formatting references. We have reviewed all AI-assisted work. The authors checked every AI-assisted proof step by step against the stated assumptions, checked the plotted values of every AI-plotted figure against the experiment logs, checked the selected research topic for overlap with existing work through a literature survey, and verified every reference against its publisher page, CrossRef or DBLP. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

## REFERENCES

Martín Abadi, Andy Chu, Ian J. Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 308–318. ACM, 2016.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A. Choquette-Choo, Hengrui Jia, Adelin Travers, Baiwu Zhang, David Lie, and Nicolas Papernot. Machine unlearning. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pp. 141–159. IEEE, 2021.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *43rd IEEE Symposium on Security and Privacy, SP 2022, San Francisco, CA, USA, May 22-26, 2022*, pp. 1897–1914. IEEE, 2022.

Min Chen, Zhikun Zhang, Tianhao Wang, Michael Backes, Mathias Humbert, and Yang Zhang. When machine unlearning jeopardizes privacy. In *CCS '21: 2021 ACM SIGSAC Conference on Computer and Communications Security, Virtual Event, Republic of Korea, November 15 - 19, 2021*, pp. 896–911. ACM, 2021.

Thorsten Eisenhofer, Doreen Riepel, Varun Chandrasekaran, Esha Ghosh, Olga Ohrimenko, and Nicolas Papernot. Verifiable and provably secure machine unlearning. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning, SaTML 2025, Copenhagen, Denmark, April 9-11, 2025*, pp. 479–496. IEEE, 2025. doi: 10.1109/SaTML64287.2025.00033.

Chongyu Fan, Jiancheng Liu, Yihua Zhang, Eric Wong, Dennis Wei, and Sijia Liu. Salun: Empowering machine unlearning via gradient-based weight saliency in both image classification and generation. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Peter W. Frey and David J. Slate. Letter recognition using holland-style adaptive classifiers. *Mach. Learn.*, 6:161–182, 1991.

Chong Fu, Xuhong Zhang, Shouling Ji, Jinyin Chen, Jingzheng Wu, Shanqing Guo, Jun Zhou, Alex X. Liu, and Ting Wang. Label inference attacks against vertical federated learning. In *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pp. 1397–1414. USENIX Association, 2022.

Aditya Golatkar, Alessandro Achille, and Stefano Soatto. Eternal sunshine of the spotless net: Selective forgetting in deep networks. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 9301–9309. IEEE, 2020. doi: 10.1109/CVPR42600.2020.00932.

Hanlin Gu, WinKent Ong, Chee Seng Chan, and Lixin Fan. Ferrari: Federated feature unlearning via optimizing feature sensitivity. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024.

Hanlin Gu, Hong Xi Tae, Lixin Fan, and Chee Seng Chan. Towards privacy-guaranteed label unlearning in vertical federated learning: Few-shot forgetting without disclosure, 2026. URL https://arxiv.org/abs/2410.10922.

Chuan Guo, Tom Goldstein, Awni Y. Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 3832–3842. PMLR, 2020.

Mengde Han, Tianqing Zhu, Lefeng Zhang, Huan Huo, and Wanlei Zhou. Vertical federated unlearning via backdoor certification. *IEEE Trans. Serv. Comput.*, 18(2):1110–1123, 2025.

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 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90.

Zachary Izzo, Mary Anne Smart, Kamalika Chaudhuri, and James Zou. Approximate data deletion from machine learning models. In *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS 2021, April 13-15, 2021, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pp. 2008–2016. PMLR, 2021.

Hengrui Jia, Mohammad Yaghini, Christopher A. Choquette-Choo, Natalie Dullerud, Anvith Thudi, Varun Chandrasekaran, and Nicolas Papernot. Proof-of-learning: Definitions and practice. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pp. 1039–1056. IEEE, 2021. doi: 10.1109/SP40001.2021.00106.

Yu Jiang, Xindi Tong, Ziyao Liu, Xiaoxi Zhang, Kwok-Yan Lam, and Chee-Wei Tan. Certifying the right to be forgotten: Primal-dual optimization for sample and label unlearning in vertical federated learning. *IEEE Trans. Inf. Forensics Secur.*, 20:13143–13158, 2025.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Meghdad Kurmanji, Peter Triantafillou, Jamie Hayes, and Eleni Triantafillou. Towards unbounded machine unlearning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Yang Liu, Yan Kang, Tianyuan Zou, Yanhong Pu, Yuanqin He, Xiaozhou Ye, Ye Ouyang, Ya-Qin Zhang, and Qiang Yang. Vertical federated learning: Concepts, advances, and challenges. *IEEE Trans. Knowl. Data Eng.*, 36(7):3615–3634, 2024.

Zhaobo Lu, Xu Zhang, Tao Li, Yilei Wang, Guangshun Li, and Zhiquan Liu. Feaun: Feature unlearning in vertical federated learning for IIoT against feature inference attacks. *Neurocomputing*, 652:131110, 2025. doi: 10.1016/j.neucom.2025.131110.

Xinjian Luo, Yuncheng Wu, Xiaokui Xiao, and Beng Chin Ooi. Feature inference attack on model predictions in vertical federated learning. In *37th IEEE International Conference on Data Engineering, ICDE 2021, Chania, Greece, April 19-22, 2021*, pp. 181–192. IEEE, 2021.

Zijie Pan, Zuobin Ying, Yajie Wang, Chuan Zhang, Weiting Zhang, Wanlei Zhou, and Liehuang Zhu. Feature-based machine unlearning for vertical federated learning in iot networks. *IEEE Trans. Mob. Comput.*, 24(6):5031–5044, 2025.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pp. 1177–1184. Curran Associates, Inc., 2007.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: Algorithms for machine unlearning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 18075–18086, 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, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pp. 3–18. IEEE Computer Society, 2017.

David Marco Sommer, Liwei Song, Sameer Wagh, and Prateek Mittal. Athena: Probabilistic verification of machine unlearning. *Proc. Priv. Enhancing Technol.*, 2022(3):268–290, 2022. doi: 10.56553/popets-2022-0072.

Hong Xi Tae and Chee Seng Chan. A survey of challenges and opportunities in vertical federated unlearning. *IEEE Access*, 13:153959–153978, 2025.

Xiangyun Tang, Xinxin Hong, Minggang Gan, Yijing Lin, Tao Zhang, Junxian Duan, and Liehuang Zhu. LVFUS: vertical federated unlearning for intelligent network security via adaptive optimizer switching. *IEEE Trans. Netw. Sci. Eng.*, 13:4138–4154, 2026.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022*, pp. 4007–4022. USENIX Association, 2022.

Ayush K. Varshney, Konstantinos Vandikas, and Vicenç Torra. Unlearning clients, features and samples in vertical federated learning. *Proc. Priv. Enhancing Technol.*, 2025(2):39–53, 2025.

Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *CoRR*, abs/1812.00564, 2018.

Yilei Wang, Zhaobo Lu, Zhiquan Liu, Tao Li, Zhenhua Chen, and Willy Susilo. Cer-feaun: Certified feature unlearning in vertical federated learning. *IEEE Trans. Mob. Comput.*, 25(1):817–829, 2026.

Zichen Wang, Xiangshan Gao, Cong Wang, Peng Cheng, and Jiming Chen. Efficient vertical federated unlearning via fast retraining. *ACM Trans. Internet Techn.*, 24(2):11:1–11:22, 2024.

Alexander Warnecke, Lukas Pirch, Christian Wressnegger, and Konrad Rieck. Machine unlearning of features and labels. In *30th Annual Network and Distributed System Security Symposium, NDSS 2023, San Diego, California, USA, February 27 - March 3, 2023*. The Internet Society, 2023. doi: 10.14722/ndss.2023.23087.

Wenhan Wu, Zhili He, Huanghuang Liang, Yili Gong, Jiawei Jiang, Chuang Hu, and Dazhao Cheng. REMISVFU: vertical federated unlearning via representation misdirection for intermediate output feature. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 26983–26991. AAAI Press, 2026.

Zhaomin Wu, Junyi Hou, and Bingsheng He. Vertibench: Advancing feature distribution diversity in vertical federated learning benchmarks. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *CoRR*, abs/1708.07747, 2017.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol.*, 10(2):12:1–12:19, 2019. doi: 10.1145/3298981.

# APPENDIX CONTENTS

**A Related Work** **14**

**B Detailed Benchmark and Ablation Results** **14**

B.1 Sensitivity (Q3) . . . . . 16

**C Technical Appendix** **16**

C.1 Notation Table . . . . . 16

C.2 Dataset Details . . . . . 16

C.3 Baseline Descriptions . . . . . 16

C.4 Evaluation Metrics . . . . . 18

C.5 Implementation Details . . . . . 19

C.6 Algorithm . . . . . 19

C.7 Audits across Decoder Classes and Read-Outs . . . . . 19

C.8 Broader Impacts . . . . . 20

**D Additional Experiments on Task-Aware Rank Reduction** **22**

D.1 Recovery under Stronger Decoders . . . . . 22

D.2 Utility–Recovery Tradeoffs across Datasets . . . . . 22

D.3 Recovery of Original Numerical Fields . . . . . 23

D.4 Direct Target Recovery with Auxiliary Data . . . . . 23

D.5 Independent Audits and Publication Faults . . . . . 24

D.6 Deployment Cost and Scaling . . . . . 24

**E Proofs of Theoretical Results** **25**

E.1 Assumptions and Scope . . . . . 25

E.2 Proof of Lemma 1 (Decoder Dictionary) . . . . . 25

E.3 Proof of Theorem 2 (Two Floors) . . . . . 26

E.4 Proof of Theorem 3 (Surcharge) . . . . . 27

E.5 Derivation of the Coordinated Edit . . . . . 28

E.6 Derivation of the Exact Read-Out Downdate . . . . . 30

E.7 Proof of Lemma 4 (Sampling and Approximation Slacks) . . . . . 30

E.8 Proof of Theorem 5 (Two-Sided Soundness) . . . . . 32

## A Related Work

**Vertical federated unlearning.** Party-level approaches include teacher distillation (Varshney et al., 2025), representation misdirection (Wu et al., 2026), aggregation replay (Tang et al., 2026) and checkpoint recovery (Wang et al., 2024). Feature methods use certified removal (Wang et al., 2026), sensitivity erasure (Gu et al., 2024), feature removal read against inference attacks (Lu et al., 2025) and noise-calibrated updates (Pan et al., 2025); sample and label methods use gradient updates (Varshney et al., 2025; Han et al., 2025), orthogonal projections (Jiang et al., 2025) and variational suppression (Gu et al., 2026). Our comparison evaluates the reconstructible footprint that remains after these task-oriented procedures. **CLEAVE** directly constrains joint reconstruction while restricting deployment to local adapters.

**Centralized machine unlearning.** Centralized methods include shard retraining (Bourtoule et al., 2021), influence-based removal of points, features and labels (Izzo et al., 2021; Warnecke et al., 2023), weight scrubbing (Golatkar et al., 2020), teacher–student objectives (Kurmanji et al., 2023) and saliency-selected updates (Fan et al., 2024), with membership inference providing an attack-based audit (Shokri et al., 2017). Vertical federations additionally distribute a sample's coordinates across institutions and expose intermediate embeddings. This makes local deployment and cross-party reconstruction central to the deletion problem.

**Certified deletion and verification.** Certified removal bounds changes to trained models under assumptions on the objective or training procedure (Guo et al., 2020; Abadi et al., 2016; Sekhari et al., 2021). Verification can also use planted markers (Sommer et al., 2022), membership probes (Chen et al., 2021) or cryptographic proofs over the training and deletion transcript (Jia et al., 2021; Eisenhofer et al., 2025). **CLEAVE** supplies a complementary reconstruction-based declaration: a footprint, an audited interface and a decoder class. Its geometric benchmarks separate the minimum locality cost from the excess of the deployed candidate, while fresh-sample audits evaluate the committed release.

## B Detailed Benchmark and Ablation Results

The image benchmarks complement the Letter comparison in Table 1. All benchmark tables report means and standard deviations over five independent runs; the component ablation isolates how each stage changes reconstruction and utility.

Table 4 separates the contributions of local editing, coordination, absorption and purge. The coordinated edit accounts for the largest reduction across deletion granularities, while the read-out and cache operations affect different surviving channels. The utility columns distinguish reduced recovery from damage to the retained task. Removing coordination leaves cross-party reconstruction available even though each party applies its own local transformation. The full-state operations remain necessary because embedding-level edits do not remove old read-outs or records from surviving caches.

Table 4: Component ablation of **CLEAVE** on Fashion-MNIST over five seeds, the party and feature units above and the sample and label units below; within each, the upper block accumulates components and the lower removes one at a time. Best in **red**, runner-up in blue among the variants.

| | Party | | | Feature | | |
|---|---|---|---|---|---|---|
| **Variant** | Share↓ | Util.↑ | $\hat{\varepsilon}$↓ | Share↓ | Util.↑ | $\hat{\varepsilon}$↓ |
| Baseline | 0.775(±0.018) | **0.924**(±0.001) | 1.163(±0.066) | 0.924(±0.020) | **0.924**(±0.001) | 1.352(±0.071) |
| + absorption | 0.655(±0.022) | 0.911(±0.001) | 0.973(±0.052) | 0.765(±0.051) | 0.918(±0.001) | 1.157(±0.078) |
| + purge | 0.639(±0.020) | 0.910(±0.001) | 0.935(±0.048) | 0.750(±0.038) | 0.917(±0.001) | 1.121(±0.064) |
| + local edit | 0.562(±0.034) | 0.908(±0.001) | 0.748(±0.083) | 0.662(±0.050) | 0.916(±0.001) | 0.943(±0.081) |
| **+ coordination (CLEAVE)** | **0.429**(±0.023) | 0.904(±0.002) | **0.446**(±0.059) | **0.486**(±0.035) | 0.913(±0.002) | **0.539**(±0.047) |
| w/o edit | 0.639(±0.020) | 0.910(±0.001) | 0.935(±0.048) | 0.750(±0.038) | 0.917(±0.001) | 1.121(±0.064) |
| w/o coordination | 0.562(±0.034) | 0.908(±0.001) | 0.748(±0.083) | 0.662(±0.050) | 0.916(±0.001) | 0.943(±0.081) |
| w/o absorption | 0.435(±0.021) | 0.889(±0.003) | 0.500(±0.044) | 0.522(±0.027) | 0.905(±0.006) | 0.642(±0.033) |
| w/o purge | 0.447(±0.022) | 0.910(±0.001) | 0.520(±0.043) | 0.534(±0.026) | 0.916(±0.001) | 0.670(±0.045) |
| | **Sample** | | | **Label** | | |
| **Variant** | Share↓ | Util.↑ | $\hat{\varepsilon}$↓ | Share↓ | Util.↑ | $\hat{\varepsilon}$↓ |
| Baseline | 0.326(±0.018) | 0.925(±0.003) | 0.671(±0.044) | 0.798(±0.005) | **0.924**(±0.002) | 1.104(±0.033) |
| + absorption | 0.334(±0.010) | 0.929(±0.003) | 0.539(±0.034) | 0.797(±0.007) | 0.862(±0.001) | 0.906(±0.018) |
| + purge | 0.277(±0.021) | 0.930(±0.002) | 0.433(±0.037) | 0.773(±0.010) | 0.860(±0.002) | 0.876(±0.021) |
| + local edit | 0.221(±0.039) | 0.931(±0.002) | 0.336(±0.065) | 0.701(±0.007) | 0.858(±0.001) | 0.731(±0.009) |
| **+ coordination (CLEAVE)** | **0.069**(±0.016) | **0.934**(±0.001) | **0.103**(±0.028) | **0.448**(±0.016) | 0.856(±0.002) | **0.476**(±0.018) |
| w/o edit | 0.277(±0.021) | 0.930(±0.002) | 0.433(±0.037) | 0.773(±0.010) | 0.860(±0.002) | 0.876(±0.021) |
| w/o coordination | 0.221(±0.039) | 0.931(±0.002) | 0.336(±0.065) | 0.701(±0.007) | 0.858(±0.001) | 0.731(±0.009) |
| w/o absorption | 0.131(±0.018) | 0.933(±0.003) | 0.172(±0.028) | 0.474(±0.014) | 0.855(±0.001) | 0.503(±0.013) |
| w/o purge | 0.174(±0.019) | 0.928(±0.002) | 0.249(±0.041) | 0.501(±0.015) | 0.854(±0.002) | 0.535(±0.017) |

The accumulation rows show how the components combine; the removal rows show the effect of withdrawing each component from the complete protocol. These comparisons use the same workload and deletion units, so their interpretation is about the component's role within this protocol rather than a ranking of unrelated algorithms. The certificate readings are reported with the achieved recovery

Table 2: Main comparison on Fashion-MNIST across the four deletion granularities, mean over five seeds. Marking as in Table 1.

| Method | Util.↑ | Share↓ | MIA↓ | ASR↓ | $\hat{\varepsilon}$↓ | Comm (MB)↓ |
|---|---|---|---|---|---|---|
| *Party Unlearning* | | | | | | |
| ReMisVFU [AAAI'26] (Wu et al., 2026) | 0.810(±0.014) | 0.652(±0.022) | 0.020(±0.004) | 0.125(±0.007) | 0.826(±0.027) | 1757.8(±43.9) |
| VFU-KD [PoPETs'25] (Varshney et al., 2025) | 0.870(±0.003) | 0.717(±0.020) | 0.015(±0.004) | 0.143(±0.014) | 0.913(±0.036) | 0.112(±0.019) |
| FedR2S [ACM TOIT'24] (Wang et al., 2024) | 0.887(±0.001) | 0.738(±0.016) | 0.030(±0.005) | 0.134(±0.020) | 0.947(±0.024) | 1142.6(±31.8) |
| LVFUS [IEEE TNSE'26] (Tang et al., 2026) | 0.891(±0.002) | 0.755(±0.019) | 0.041(±0.010) | 0.178(±0.012) | 0.998(±0.031) | 1673.8(±46.2) |
| **CLEAVE** | **0.904**(±0.005) | **0.429**(±0.020) | **0.009**(±0.003) | **0.071**(±0.009) | **0.446**(±0.017) | **0.028**(±0.005) |
| *Retrain* | 0.912(±0.003) | 0.756(±0.025) | 0.043(±0.009) | 0.149(±0.011) | 1.034(±0.044) | 1757.8(±48.6) |
| *Feature Unlearning* | | | | | | |
| VFU-KD [PoPETs'25] (Varshney et al., 2025) | 0.822(±0.011) | 0.741(±0.027) | 0.024(±0.006) | 0.384(±0.043) | 0.731(±0.035) | 0.103(±0.016) |
| Cer-FeaUn [IEEE TMC'26] (Wang et al., 2026) | 0.649(±0.046) | 0.752(±0.023) | 0.012(±0.002) | **0.005**(±0.005) | 0.812(±0.028) | 87.9(±3.6) |
| LVFUS [IEEE TNSE'26] (Tang et al., 2026) | **0.917**(±0.002) | 0.781(±0.027) | 0.053(±0.004) | 0.999(±0.006) | 0.978(±0.029) | 2493.0(±35.5) |
| Pan et al. [IEEE TMC'25] (Pan et al., 2025) | 0.874(±0.005) | 0.820(±0.035) | 0.025(±0.007) | 0.362(±0.054) | 0.759(±0.033) | 14.6(±1.1) |
| Ferrari [NeurIPS'24] (Gu et al., 2024) | 0.905(±0.006) | 0.910(±0.017) | 0.041(±0.006) | 0.795(±0.050) | 1.107(±0.046) | 0.071(±0.010) |
| **CLEAVE** | 0.913(±0.003) | **0.486**(±0.024) | **0.011**(±0.003) | 0.006(±0.002) | **0.539**(±0.015) | **0.023**(±0.004) |
| *Retrain* | 0.917(±0.002) | 0.815(±0.034) | 0.058(±0.010) | 1.000(±0.003) | 1.136(±0.052) | 2636.7(±59.1) |
| *Sample Unlearning* | | | | | | |
| VFU-GA [PoPETs'25] (Varshney et al., 2025) | 0.925(±0.002) | 0.330(±0.009) | 0.053(±0.005) | 0.450(±0.022) | 0.563(±0.029) | 0.079(±0.011) |
| Han et al. [IEEE TSC'25] (Han et al., 2025) | 0.264(±0.057) | 0.071(±0.012) | **0.005**(±0.005) | 0.426(±0.308) | 0.219(±0.020) | 372.9(±8.0) |
| FedORA [IEEE TIFS'25] (Jiang et al., 2025) | 0.832(±0.067) | 0.326(±0.006) | 0.030(±0.010) | 0.288(±0.156) | 0.516(±0.033) | 187.7(±5.7) |
| LVFUS [IEEE TNSE'26] (Tang et al., 2026) | 0.926(±0.001) | 0.360(±0.015) | 0.008(±0.004) | 0.098(±0.003) | 0.534(±0.021) | 2579.6(±57.3) |
| SCRUB [NeurIPS'23] (Kurmanji et al., 2023) | 0.895(±0.005) | 0.139(±0.014) | 0.012(±0.005) | 0.109(±0.005) | 0.311(±0.019) | 848.2(±19.6) |
| SalUn [ICLR'24] (Fan et al., 2024) | 0.929(±0.001) | 0.358(±0.015) | 0.032(±0.003) | 0.430(±0.163) | 0.574(±0.026) | 878.9(±21.7) |
| **CLEAVE** | **0.934**(±0.003) | **0.069**(±0.006) | **0.005**(±0.002) | **0.006**(±0.003) | **0.103**(±0.010) | **0.028**(±0.004) |
| *Retrain* | 0.942(±0.001) | 0.348(±0.017) | 0.011(±0.001) | 0.099(±0.003) | 0.483(±0.024) | 2505.2(±55.6) |
| *Label Unlearning* | | | | | | |
| LUV [ICLR'26] (Gu et al., 2026) | 0.259(±0.068) | 0.793(±0.004) | **0.014**(±0.007) | 0.006(±0.003) | 0.904(±0.037) | 17.2(±1.1) |
| FedORA [IEEE TIFS'25] (Jiang et al., 2025) | 0.847(±0.037) | 0.805(±0.005) | 0.037(±0.009) | 0.004(±0.002) | 0.861(±0.041) | 319.9(±7.8) |
| SalUn [ICLR'24] (Fan et al., 2024) | 0.850(±0.001) | 0.803(±0.007) | 0.064(±0.006) | 0.005(±0.002) | 0.882(±0.032) | 878.9(±22.4) |
| **CLEAVE** | **0.856**(±0.003) | **0.448**(±0.013) | 0.018(±0.005) | **0.002**(±0.001) | **0.476**(±0.013) | **0.025**(±0.004) |
| *Retrain* | 0.864(±0.004) | 0.632(±0.006) | 0.030(±0.011) | 0.003(±0.001) | 0.716(±0.026) | 2373.0(±51.2) |

Table 3: Main comparison on CIFAR-100 across the four deletion granularities, mean over five seeds. Marking as in Table 1.

| Method | Util.↑ | Share↓ | MIA↓ | ASR↓ | $\hat{\varepsilon}$↓ | Comm (MB)↓ |
|---|---|---|---|---|---|---|
| *Party Unlearning* | | | | | | |
| ReMisVFU [AAAI'26] (Wu et al., 2026) | 0.238(±0.014) | 0.154(±0.012) | 0.106(±0.024) | 0.005(±0.003) | 0.386(±0.019) | 1464.8(±39.4) |
| VFU-KD [PoPETs'25] (Varshney et al., 2025) | 0.274(±0.004) | 0.166(±0.013) | 0.107(±0.017) | 0.207(±0.015) | 0.441(±0.025) | 0.091(±0.014) |
| FedR2S [ACM TOIT'24] (Wang et al., 2024) | 0.304(±0.007) | 0.207(±0.015) | 0.294(±0.017) | 0.259(±0.059) | 0.562(±0.041) | 952.1(±27.6) |
| LVFUS [IEEE TNSE'26] (Tang et al., 2026) | 0.306(±0.005) | 0.202(±0.011) | 0.357(±0.025) | 0.385(±0.026) | 0.651(±0.034) | 1461.5(±31.1) |
| **CLEAVE** | **0.379**(±0.006) | **0.075**(±0.006) | **0.012**(±0.004) | **0.004**(±0.002) | **0.089**(±0.008) | **0.025**(±0.004) |
| *Retrain* | 0.387(±0.007) | 0.281(±0.008) | 0.314(±0.022) | 0.121(±0.032) | 0.528(±0.030) | 1464.8(±41.2) |
| *Feature Unlearning* | | | | | | |
| VFU-KD [PoPETs'25] (Varshney et al., 2025) | 0.271(±0.018) | 0.319(±0.021) | 0.083(±0.014) | 0.291(±0.037) | 0.394(±0.023) | 0.088(±0.013) |
| Cer-FeaUn [IEEE TMC'26] (Wang et al., 2026) | 0.146(±0.032) | 0.327(±0.014) | 0.049(±0.007) | 0.018(±0.006) | 0.276(±0.017) | 73.2(±2.8) |
| LVFUS [IEEE TNSE'26] (Tang et al., 2026) | 0.357(±0.006) | 0.306(±0.015) | 0.394(±0.025) | 0.993(±0.003) | 0.721(±0.040) | 2192.2(±46.7) |
| Pan et al. [IEEE TMC'25] (Pan et al., 2025) | 0.294(±0.008) | 0.340(±0.015) | 0.126(±0.017) | 0.229(±0.016) | 0.453(±0.026) | 12.2(±0.9) |
| Ferrari [NeurIPS'24] (Gu et al., 2024) | 0.363(±0.022) | 0.516(±0.046) | 0.217(±0.029) | 0.784(±0.122) | 0.602(±0.039) | 0.063(±0.008) |
| **CLEAVE** | **0.417**(±0.006) | **0.159**(±0.010) | **0.010**(±0.004) | **0.006**(±0.003) | **0.219**(±0.010) | **0.025**(±0.005) |
| *Retrain* | 0.425(±0.008) | 0.303(±0.008) | 0.337(±0.046) | 0.993(±0.005) | 0.655(±0.038) | 2197.3(±44.9) |
| *Sample Unlearning* | | | | | | |
| VFU-GA [PoPETs'25] (Varshney et al., 2025) | 0.441(±0.006) | 0.189(±0.012) | 0.332(±0.026) | 0.664(±0.029) | 0.426(±0.025) | 0.068(±0.010) |
| Han et al. [IEEE TSC'25] (Han et al., 2025) | 0.015(±0.008) | 0.023(±0.006) | **0.005**(±0.001) | **0.004**(±0.009) | 0.071(±0.012) | 384.0(±6.1) |
| FedORA [IEEE TIFS'25] (Jiang et al., 2025) | 0.382(±0.010) | 0.208(±0.015) | 0.183(±0.035) | 0.607(±0.026) | 0.382(±0.031) | 168.4(±4.6) |
| LVFUS [IEEE TNSE'26] (Tang et al., 2026) | 0.469(±0.006) | 0.203(±0.004) | 0.008(±0.002) | 0.010(±0.003) | 0.391(±0.018) | 2930.8(±38.8) |
| SCRUB [NeurIPS'23] (Kurmanji et al., 2023) | 0.055(±0.006) | 0.010(±0.001) | 0.007(±0.002) | 0.552(±0.091) | 0.129(±0.013) | 706.2(±18.8) |
| SalUn [ICLR'24] (Fan et al., 2024) | 0.464(±0.007) | 0.191(±0.009) | 0.258(±0.014) | 0.619(±0.079) | 0.405(±0.028) | 732.4(±16.9) |
| **CLEAVE** | **0.476**(±0.005) | **0.008**(±0.002) | 0.006(±0.002) | **0.004**(±0.002) | **0.061**(±0.007) | **0.026**(±0.004) |
| *Retrain* | 0.484(±0.002) | 0.186(±0.014) | 0.007(±0.002) | 0.011(±0.004) | 0.244(±0.018) | 2085.1(±42.4) |
| *Label Unlearning* | | | | | | |
| LUV [ICLR'26] (Gu et al., 2026) | 0.208(±0.048) | 0.253(±0.011) | 0.010(±0.008) | 0.005(±0.003) | 0.376(±0.024) | 17.2(±1.0) |
| FedORA [IEEE TIFS'25] (Jiang et al., 2025) | 0.409(±0.006) | 0.316(±0.019) | 0.140(±0.029) | 0.239(±0.090) | 0.472(±0.027) | 78.2(±2.8) |
| SalUn [ICLR'24] (Fan et al., 2024) | 0.462(±0.003) | 0.343(±0.019) | 0.094(±0.027) | 0.530(±0.066) | 0.554(±0.032) | 732.4(±17.1) |
| **CLEAVE** | **0.468**(±0.003) | **0.167**(±0.008) | **0.007**(±0.003) | **0.003**(±0.001) | **0.236**(±0.008) | **0.027**(±0.004) |
| *Retrain* | 0.476(±0.004) | 0.159(±0.006) | 0.018(±0.011) | 0.003(±0.001) | 0.251(±0.019) | 2175.3(±47.0) |

and utility, preserving the distinction between a measured attack outcome and an issued level. Further diagnostic figures follow in Appendix C.

Figure 5(a) compares the identity-edit residual with independently probed recovery. Panel (b) contains 540 requests from three datasets, four granularities, nine leakage levels and five independent runs. All satisfy the requested leakage level, while 423 also satisfy the utility constraint: the joint acceptance rate is 78.3%, and 21.7% of targets remain unmet. Crosses identify these latter requests. The mean leakage slack therefore measures the leakage constraint, while the acceptance flag checks both constraints. Panel (c) reports empirical audit coverage under the declared probe.

Figure 5(d) uses the best feasible values returned by descending leakage-target searches at utility ratio $0.85$: 60 correlated Gaussian cells and ten tabular cells, each configuration repeated five times. Both search families retain their feasible candidates, and the joint pool also includes each block-local candidate; this construction explains their ordering. These plotted values are candidate upper bounds in the implemented regularized leakage functional. They are distinct from the exact-utility, unrestricted-decoder floors of Theorem 3. Eq. (27) gives the latter for general correlated covariance, and Eq. (61) separates the locality surcharge from solver excess.

The frozen-inverse objective proposes candidates; constraint checks use the unfrozen evaluation. Eq. (41) gives the exact trace discrepancy and its operator-norm bound for the theoretical functional. The archived sweep records leakage, utility and acceptance, so its feasibility diagnostic directly evaluates the deployed candidates.

### B.1 Sensitivity (Q3)

For **Q3**, we vary the size of the deletion request. Figure 6 follows the certificate as the request grows: **`CLEAVE`** runs at two requested levels the deployer picks, a tuning range no published method offers, and the certificate issued at the looser one stays above the share measured in every panel. Appendix C carries the requested-level, editing-size and party-count sweeps, the stressed certificate and the edit geometry.

## C Technical Appendix

### C.1 Notation Table

We summarize the commonly used notation throughout this paper in Table 5.

### C.2 Dataset Details

We evaluate on one tabular and two image classification benchmarks, chosen so that every deletion granularity has enough classes and features to be exercised.

**Letter** [ML'91] **(Frey & Slate, 1991).** Letter holds $20{,}000$ samples of $16$ integer features describing distorted capital letters, labeled with the $26$ letters of the alphabet. Its features are hand-designed statistics of the glyph image, so blocks assigned to different parties are correlated by construction, which is the regime in which the residual index of Section 2 is furthest from zero. It is the tabular workload and the one on which the certificate is stressed hardest in Appendix C.7.

**Fashion-MNIST** [arXiv'17] **(Xiao et al., 2017).** Fashion-MNIST contains $70{,}000$ grayscale images of $28 \times 28$ pixels in $10$ clothing categories, split into $60{,}000$ training and $10{,}000$ test images. It is the smaller image workload, where a vertical strip of the image is a party's feature block.

**CIFAR-100** [TR'09] **(Krizhevsky et al., 2009).** CIFAR-100 contains $60{,}000$ color images of $32 \times 32$ pixels in $100$ classes, split into $50{,}000$ training and $10{,}000$ test images. Its class count is what the label unit needs, and it is the workload on which the retained blocks predict the departed block least well.

**Vertical partition.** Every federation holds one active party and three passive parties. Letter is partitioned into per-party feature blocks by the VertiBench importance protocol (Wu et al., 2024) at parameter $0.5$, whose knobs set the cross-party redundancy at the center of our analysis, and the image datasets follow the contiguous vertical-strip split, one strip per party, with the middle strip as the default deletion target. The split is drawn once and held fixed, so method comparisons carry no split variance. A linear-Gaussian federation with closed-form ground truth calibrates the leakage probes and does not enter the comparison tables.

### C.3 Baseline Descriptions

This section describes the thirteen counterparts of Section 4.1, grouped by the deletion unit at which each is run. Every counterpart is a published method run at the operating point its own paper sets, under the shared vertical partition above.

Table 5: Notation and Definitions.

| Notation | Definition |
|---|---|
| $m$ | The number of parties, one active and $m-1$ passive. |
| $X_c$, $h_c$, $H_c$ | The feature block, bottom model and embedding of party $c$. |
| $g_\theta$, $y$ | The active party's top model and the labels it holds. |
| $n$ | The number of aligned samples across the federation. |
| $k$ | The passive party named in a deletion request. |
| $X_{-k}$ | The features retained by the parties other than $k$. |
| $Z_k$ | The active party's cache of the embeddings received from party $k$. |
| $Z$ | The footprint of the requested unit ($Z_k$ for a departing party). |
| $H$ | The concatenated embeddings of the retained parties. |
| $\Sigma_{ZH}$, $\Sigma_{HH}$, $\Sigma_{Hy}$ | The empirical covariances over the retained samples. |
| $\hat{Z}(\cdot)$, $\hat{y}(\cdot)$ | The best decoder of $Z$ and predictor of $y$ in the declared class. |
| $\varphi_c$, $D$ | The feature map of party $c$ and its number of random features. |
| $\hat{\rho}_k$ | The residual index, the footprint energy recoverable from $H$. |
| $\hat{\rho}_k^{\mathrm{t}}$, $\hat{\rho}_k^{\mathrm{f}}$ | The task-required and free shares of the residual index. |
| $\Pi$, $P$ | The label-coefficient projector and the task-restricted whitening. |
| $\mathcal{C}$ | The editing parties, the retained passive parties. |
| $A_c$, $A$ | The edit of party $c$ and the block-diagonal assembly of all edits. |
| $\mathcal{L}(A)$, $\mathcal{U}(A)$ | The leakage and utility functionals of an edit. |
| $Q_A$ | The whitening operator of the edited embeddings. |
| $\Sigma_{cc}$ | The $c$-th diagonal block of $\Sigma_{HH}$. |
| $\varepsilon$, $\tau$ | The requested leakage level and utility level. |
| $M$, $N$ | The leakage Gram matrix and the task Gram matrix. |
| $\mu$, $\nu$ | The multipliers of the leakage and utility constraints. |
| $\mathcal{A}$ | The block-diagonal feasible set of edits. |
| $\varepsilon^{\mathrm{B}}$, $\varepsilon^{\mathrm{C}}$ | The block-diagonal floor and the centralized anchor. |
| $S$, $G$ | The federation surcharge and the coordination gain. |
| $\tilde{H}$, $\Gamma$, $\lambda$ | The edited embeddings, their regularized Gram matrix and the ridge. |
| $\theta'$, $\theta_{-\mathcal{I}}$ | The re-solved read-out and its downdate after deleting $\mathcal{I}$. |
| $\widehat{\mathcal{L}}(A)$ | The held-out estimate of the leakage of the deployed edit. |
| $\Delta^{\mathrm{e}}$, $\Delta^{\mathrm{r}}$, $\Delta^{\mathrm{a}}$ | The estimation, relaxation and feature-approximation slacks. |
| $\hat{\varepsilon}$, $\delta$ | The issued leakage level and the failure probability. |
| $R$ | The number of vertical training rounds retraining spends. |

**Party-level methods.**

- **ReMisVFU** [AAAI'26] (Wu et al., 2026). ReMisVFU serves a party's exit by representation misdirection: the departing party's representation is driven toward a fixed random anchor while the server projects subsequent gradients orthogonally to the departed contribution. In the setting of Section 2 the departing party's bottom model leaves with it, so the misdirection acts on the cached embeddings of that party rather than on a live encoder.
- **VFU-KD** [PoPETs'25] (Varshney et al., 2025). VFU-KD distills the post-deletion federation from the pre-deletion model as a teacher, retraining the top model on the reduced embedding cache while the retained passive parties' bottom models stay as trained. It is the same paper's method for feature requests, which it serves by distilling against the reduced feature block.
- **FedR2S** [ACM TOIT'24] (Wang et al., 2024). FedR2S keeps per-party checkpoints along the training lineage and, on a party's exit, restarts vertical training from the last checkpoint that predates the departed contribution, which trades storage for fewer rounds than retraining from scratch.
- **LVFUS** [IEEE TNSE'26] (Tang et al., 2026). LVFUS holds a ledger of the aggregation-point updates and replays it without the requested contribution, serving party, feature and sample requests through one replay mechanism; it enters the comparison at all three of those units.

**Feature-level methods.**

- **Cer-FeaUn** [IEEE TMC'26] (Wang et al., 2026). Cer-FeaUn is certified feature unlearning for vertical federations: the requesting party removes the features from its block and the federation applies a calibrated update whose certificate states statistical indistinguishability from retraining without those features.
- **Pan et al.** [IEEE TMC'25] (Pan et al., 2025). Pan et al. serve a feature request by a noise-calibrated perturbation of the requesting party's contribution, sized so that the features' influence on the top model is masked, and are the lightest of the feature-level counterparts in communication.
- **Ferrari** [NeurIPS'24] (Gu et al., 2024). Ferrari arrives from horizontal federated feature unlearning and optimizes a feature-sensitivity objective that makes the model's output insensitive to the requested features. Here it runs at the requesting party's own bottom model, which is where the requested features enter the federation.

**Sample-level methods.**

- **VFU-GA** [PoPETs'25] (Varshney et al., 2025). VFU-GA runs gradient ascent on the requested samples through the assembled split network, with gradients crossing the cut layer as in ordinary vertical training, and stops once the samples' loss reaches the level of unseen data.
- **Han et al.** [IEEE TSC'25] (Han et al., 2025). Han et al. certify sample removal through a planted backdoor: gradient ascent on the requested samples is confined to a ball around the trained weights and stopped by a radius rule, and removal is read from the backdoor no longer firing. Its native unit is a share of samples on one party's block of a two-party split network, and we run it on the same cohort the other sample-level rows receive.
- **FedORA** [IEEE TIFS'25] (Jiang et al., 2025). FedORA is a primal-dual constrained optimization at the server that projects the update orthogonally to the retained contribution while a dual variable enforces the removal constraint; it serves sample and label requests and enters the comparison at both units.
- **SCRUB** [NeurIPS'23] (Kurmanji et al., 2023). SCRUB is a centralized teacher-student method that distills the retained data from the trained teacher while pushing the requested samples away from it, and runs here on the assembled split network.
- **SalUn** [ICLR'24] (Fan et al., 2024). SalUn selects parameters by the saliency of the forget gradient and updates only those with random relabeling of the requested samples; it is the second centralized method run on the assembled split network, and it also enters the label unit.

**Label-level methods.**

- **LUV** [ICLR'26] (Gu et al., 2026). LUV is label unlearning for vertical federations with a privacy guarantee: a variational objective suppresses the deleted class in the active party's read-out while the passive parties' embeddings are kept from re-encoding it. The published construction fixes two parties, and we run its loop over the $m - 1$ passive parties of our federations.

### C.4 Evaluation Metrics

Six metrics cover what the task keeps, what an attacker still recovers, whether the reported guarantee can be independently checked, and what deletion costs the network. Every leakage reading is paired with the utility measured at the same operating point, since the guarantee of Section 3 is a tradeoff between the two. Every number in the tables is the mean over five random seeds covering pretraining initialization and request draw, with its standard deviation beside it; the party sweep runs three.

**Task utility.** Top-1 accuracy. **`CLEAVE`** solves at the requested level $\tau$, a share of the task-relevant energy in Eq. (6), and each published method runs at the operating point its own paper sets.

**Recovered share.** The fraction of the unit's footprint energy the strongest decoder in the declared class recovers from the complete post-deletion state: fresh embeddings, surviving caches, and the read-out layer. It is the leakage functional at the state a method leaves, corrected by a zero anchor on the same setup, and Appendix C.7 reads the issued level against the out-of-class probe.

**Membership inference.** The true positive rate of a LiRA-style attack (Carlini et al., 2022) on the deleted unit at a false positive rate of $1\%$, which is also this reading's chance level.

**Backdoor success rate.** A trigger is planted inside the requested unit before training, and the reading is the fraction of triggered inputs that still activate the planted behavior after deletion.

**Certificate triple.** The reading collects the issued level $\hat{\varepsilon}$, which adds the slacks of Section 3 to the held-out estimate $\widehat{\mathcal{L}}(A)$, with the block floor $\varepsilon^{\mathrm{B}}$ at the same utility and the surcharge $S$. The theorem bounds the failure probability by $\delta$ under its assumptions; the observed violation frequency is an empirical diagnostic. Tightness compares the achieved level against the same-class block benchmark, whose computation status is reported.

**Communication cost.** The extra cross-party traffic deletion introduces beyond normal VFL operation, in megabytes, counted from the request to the certificate and summed over parties and rounds.

### C.5 Implementation Details

**Base models.** Each passive party runs a three-layer perceptron bottom model onto a $16$-dimensional embedding on Letter and a ResNet-18 (He et al., 2016) onto a $64$-dimensional one on the image datasets, with the active party concatenating and reading out linearly. Pretraining uses Adam at learning rate $10^{-3}$, with batch size $256$ for $20$ epochs on Letter and $128$ for $30$ epochs on the image datasets.

**Deletion requests.** A party request removes one passive party, a feature request removes a third of its block, a sample request removes a $5\%$ cohort of correlated rows, and a label request removes one class. **`CLEAVE`** runs at requested utility level $\tau$ of $0.85$ and a target deletion level of half the residual left by the identity edit, with a ridge of $10^{-8}$ on the second-order statistics and empirical intervals evaluated at nominal failure level $\delta$ of $0.05$ using $200$ bootstrap resamples. These intervals assess coverage empirically; the finite-sample guarantee of Theorem 5 requires its stated analytic slacks. The declared class is the norm-bounded linear decoder, whose radius must be fixed before auditing; the trace formulas require an inactive bound, and the out-of-class probe carries $1024$ cosine random features. Three sweeps run beside this setting, over requested leakage level, editing federation size and party count, with the grids reported below.

### C.6 Algorithm

Algorithm 1 states the round the three components of Section 3 share: what the active party forms from the statistics it already holds, what each retained passive party receives, and what the certificate is assembled from.

**What each component carries.** Table 4 prices the components against the full method. The coordinated edit carries the largest single drop at every granularity, from $0.562$ to $0.429$ at the party unit, from $0.662$ to $0.486$ at the feature unit, from $0.221$ to $0.069$ at the sample unit, and from $0.701$ to $0.448$ at the label unit; coordination itself accounts for $0.133$ to $0.253$ of that. Read-out absorption is a smaller effect: removing it moves the party row from $0.429$ to $0.435$ while utility falls from $0.904$ to $0.889$, and at the sample unit the purge stage trims share from $0.174$ to $0.069$ while utility stays essentially flat. The purge clears whatever state the requested unit addresses, the row cache included at the sample unit.

**The three sweeps.** The leakage-level sweep traces nine requested levels per dataset and granularity at five seeds, and the surcharge instrument runs two, three, five and eight parties at three block-size imbalances over five seeds, sixty cells beside ten on the native split. Figure 10 follows the share left when the largest block departs over two to sixteen parties under four Dirichlet concentrations at three seeds, and that block shrinks as the count grows, so the two are read together. Across all concentrations and every party count from 2 to 16, CIFAR-100 keeps the lowest recovered share of the three datasets, with every one of its bars below 0.3, while every Letter and Fashion-MNIST bar stays below 0.75; each bar averages the three seeds, with their spread drawn as an error bar.

### C.7 Audits across Decoder Classes and Read-Outs

Figure 9 reports the issued level's margin over the declared probe. Table 6 and Figure 8 compare that reading across a retrained federation, a near-exact utility requirement and a one-hidden-layer read-out. Each condition includes the declared linear probe and an adaptive decoder using $1024$ cosine random features. The margin is the issued level minus the declared-probe reading on the evaluated state.

**Algorithm 1** CLEAVE: **C**oordinated **L**ocal **E**diting for **A**ttested **V**ertical **E**rasure

**Input:** retained embeddings $H$ and caches; editing parties $\mathcal{C}$; footprint $Z$ of the requested unit; labels $y$; read-out layer $\theta$; requested levels $(\varepsilon, \tau)$; feature dimension $D$; decoder radius $\beta$; ridge weight $\lambda$; confidence $\delta$; local search limit.

Map each retained block $H_c$ through $\varphi_c$ built from $D$ random features of the public seed
▷ *Phase I: Residual Splitting (RS)*
Form $\Sigma_{HH}, \Sigma_{HZ}, \Sigma_{Hy}$ from the retained samples
Residual index $\hat{\rho}_k$; projector $\Pi$ from the columns of $\Sigma_{HH}^{-1}\Sigma_{Hy}$ ▷ Eq. (2)
Free share $\hat{\rho}_k^{\mathrm{f}}$ separating task-required from removable directions ▷ Eq. (3)
▷ *Phase II: Coordinated Local Editing (CLE)*
Gram matrices $M$ and $N$ from $\Sigma_{HZ}$ and $\Sigma_{Hy}$
**repeat**
    Solve for a candidate edit at multipliers $(\mu, \nu)$ ▷ Eq. (7)
    Check actual decoder-class leakage and utility on the design statistics
**until** the candidate meets $(\varepsilon, \tau)$ or the local search limit is reached
**if** the requested pair is unmet **then**
    **return** target unmet
**end if**
Send each editing party its own $A_c$; the party installs the adapter after its cut layer
▷ *Phase III: State Purge and Certification (SPC)*
Absorb the edit into the read-out layer ▷ Eq. (9)
**if** the request deletes a sample set $\mathcal{I}$ **then**
    Downdate the read-out over the deleted rows ▷ Eq. (10)
**end if**
Purge the footprint's caches and derived statistics; keep the adapter commitments
Fix the deployed state and decoder class; draw an independent audit sample
Issue $\hat{\varepsilon}$ from the audited estimate and the applicable slacks ▷ Eq. (11)
**return** adapters $\{A_c\}_{c\in\mathcal{C}}$; updated read-out layer; certificate $(\hat{\varepsilon}, \varepsilon^{\mathrm{B}}, S)$ with the floors' computation status

Table 6 separates decoder strength from changes to the published state. The adaptive decoder extracts additional information on Letter and Fashion-MNIST even at the operating point. Refitting a nonlinear head after issuance changes the release and turns the declared-probe margin negative on those datasets; the old issued level therefore does not certify the new release. The near-exact utility settings select the same edit and share one row. Across these comparisons, the paired probes identify which recovery is covered by the linear declaration and which is revealed by a richer decoder or changed read-out.

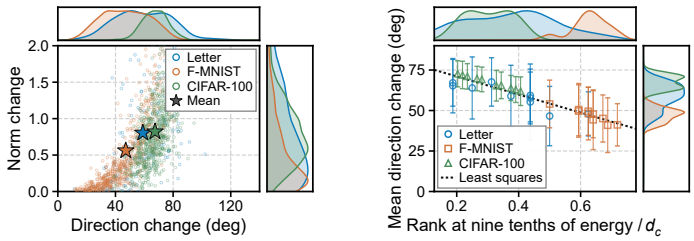
Figure 7 describes how each adapter transforms its block: column length measures scaling, and the angle from its original axis measures rotation. Larger rotations accompany stronger scaling changes, while more concentrated spectra accompany greater rotation. The closing rows of Table 6 summarize these properties alongside the rank containing nine tenths of spectral energy. The projector in Theorem 2 describes the retained subspace in whitened coordinates; these measurements describe the deployed adapter that realizes it.

Table 6: The certificate read at the operating point and under three stressed conditions, each against both probes, at the party unit over five seeds; the margin is taken against the declared probe.

| Condition | Util.↑ | Decl.↓ | Ada.↓ | $\hat{\varepsilon}$↓ | Margin |
|---|---|---|---|---|---|
| *Letter* | | | | | |
| Operating point | 0.739(±0.009) | 0.331(±0.013) | 0.575(±0.033) | 0.347(±0.016) | +0.016 |
| Edit after retraining | 0.717(±0.006) | 0.346(±0.011) | 0.628(±0.027) | 0.389(±0.010) | +0.043 |
| Near-exact bar ($\tau \geq 0.95$) | 0.747(±0.007) | 0.364(±0.018) | 0.517(±0.072) | 0.391(±0.026) | +0.027 |
| Nonlinear read-out | 0.900(±0.004) | 0.540(±0.015) | 0.576(±0.033) | 0.347(±0.016) | −0.193 |
| *Fashion-MNIST* | | | | | |
| Operating point | 0.904(±0.006) | 0.429(±0.036) | 0.619(±0.055) | 0.446(±0.042) | +0.017 |
| Edit after retraining | 0.883(±0.006) | 0.537(±0.046) | 0.598(±0.049) | 0.540(±0.041) | +0.003 |
| Near-exact bar ($\tau \geq 0.95$) | 0.911(±0.011) | 0.443(±0.039) | 0.531(±0.123) | 0.461(±0.048) | +0.018 |
| Nonlinear read-out | 0.920(±0.004) | 0.609(±0.044) | 0.620(±0.054) | 0.446(±0.042) | −0.163 |
| *CIFAR-100* | | | | | |
| Operating point | 0.379(±0.008) | 0.075(±0.005) | 0.086(±0.005) | 0.089(±0.005) | +0.014 |
| Edit after retraining | 0.358(±0.008) | 0.143(±0.007) | 0.129(±0.003) | 0.157(±0.009) | +0.014 |
| Near-exact bar ($\tau \geq 0.95$) | 0.392(±0.008) | 0.083(±0.005) | 0.094(±0.005) | 0.097(±0.005) | +0.014 |
| Nonlinear read-out | 0.369(±0.008) | 0.077(±0.005) | 0.086(±0.005) | 0.089(±0.005) | +0.012 |
| *Edit geometry: d, rank at 0.9, tail, norm change, turn* | | | | | |
| Letter | 16 | 6.0(±2.1) | 0.091(±0.008) | 0.804(±0.323) | 58.98(±7.05) |
| Fashion-MNIST | 64 | 40.3(±3.7) | 0.097(±0.001) | 0.562(±0.146) | 47.12(±3.99) |
| CIFAR-100 | 64 | 19.5(±4.5) | 0.096(±0.002) | 0.821(±0.231) | 67.43(±3.76) |

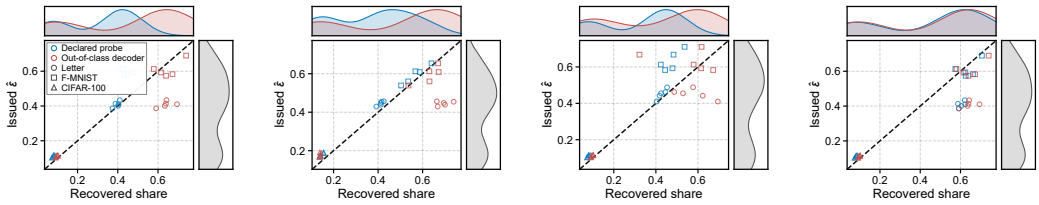
## C.8 Broader Impacts

**CLEAVE** reduces the communication required to respond to a withdrawal and provides a declared-class reading that a counterparty can



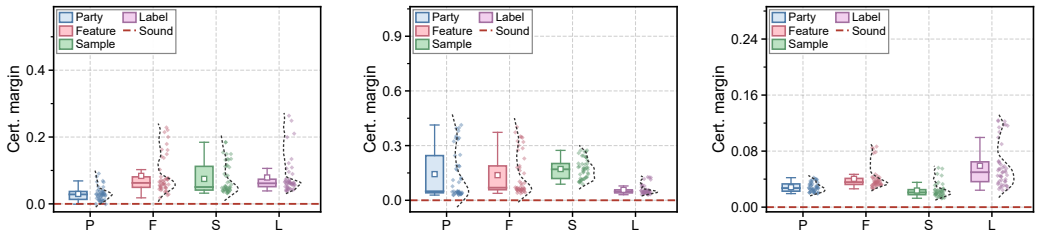

(a) Per coordinate. (b) Per adapter.

Figure 7: The edits turn what they keep and drop the rest.



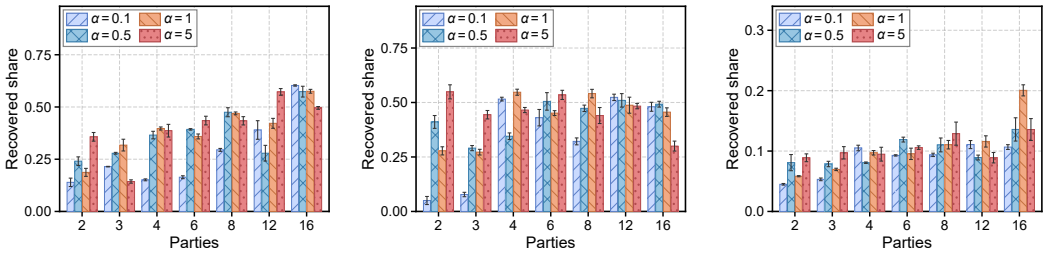

(a) Operating point. (b) After retraining. (c) Near-exact bar. (d) Nonlinear read-out.

Figure 8: The declared class is where the issued level binds.




(a) Letter. (b) Fashion-MNIST. (c) CIFAR-100.

Figure 9: The declared probe reads under the issued level.




(a) Letter. (b) Fashion-MNIST. (c) CIFAR-100.

Figure 10: The edit holds as the federation widens.

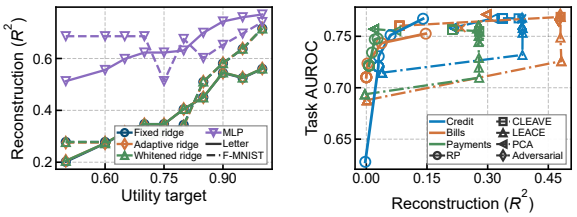
audit. Experiments use public benchmarks and established architectures. Publishing the footprint, audited interface and decoder class beside the issued level gives that assurance an explicit scope. The nonlinear-probe experiments quantify why the decoder declaration matters: in Table 6, a nonlinear head refitted after issuance turns the declared-probe margin negative on Letter and Fashion-MNIST. Federations can also use these utility–reconstruction measurements when defining withdrawal procedures and the predictive-service obligations that remain after a party leaves.

# D ADDITIONAL EXPERIMENTS ON TASK-AWARE RANK REDUCTION

These experiments examine CLEAVE-RP, a task-aware rank-reduction variant that selects a subspace using covariance with the deletion target and task labels. Every editor receives the same linear cross-entropy (CE) head-training budget, denoted by "+ CE". The additional release diagnostics use the recorded seed-0 configuration. Their empirical audits evaluate the released representation and declared attack protocol.

Candidates are selected at ten validation utility targets relative to the identity-adapter head's utility above chance. The horizontal label "Utility target" refers to this requirement rather than to the test accuracy a release attains. Among eligible candidates, validation ridge and random-feature decoders select the one with the lowest recovery. Test attacks then evaluate the selected release. Repeated selections are retained, and crosses identify unmet requirements. Reconstruction is reported as test $R^2$; lower recovery is preferable, while higher task accuracy or AUROC is preferable.

## D.1 RECOVERY UNDER STRONGER DECODERS



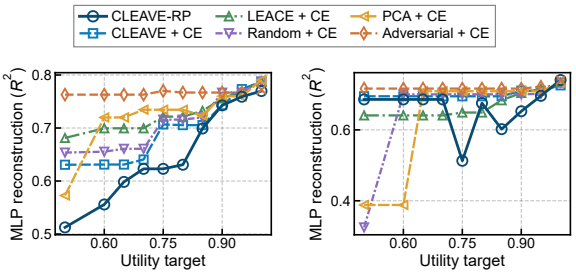

(a) Decoder attacks. (b) Numerical fields.

Figure 11: Reconstruction under stronger decoders (a) and recovery of numerical fields (b). Solid/dashed lines distinguish Letter/F-MNIST in (a); colors distinguish fields in (b). RP denotes CLEAVE-RP; all editors use CE heads.

Figure 11(a) compares reconstruction by linear and nonlinear decoders. Fixed, adaptive and whitened ridge give similar readings, whereas the MLP recovers additional footprint information over much of the utility range. The separation locates information accessible beyond the linear declaration and motivates reporting decoder strength beside the issued level.

Stricter utility requirements in Figure 11(a) generally preserve more reconstructible information. The Letter trajectory changes gradually, while the F-MNIST MLP responds more sharply to subspace selection. Flat segments correspond to utility targets selecting the same candidate. Validation ridge and random-feature recovery select the release; the held-out MLP independently evaluates its nonlinear residual.

Figure 12 compares CLEAVE-RP with LEACE, adversarial editing and the remaining controls under a common nonlinear attack and head-training protocol. The Letter curves separate at relaxed utility and converge as the task requirement tightens, linking the reconstruction reduction to the selected representation.




(a) Letter. (b) F-MNIST.

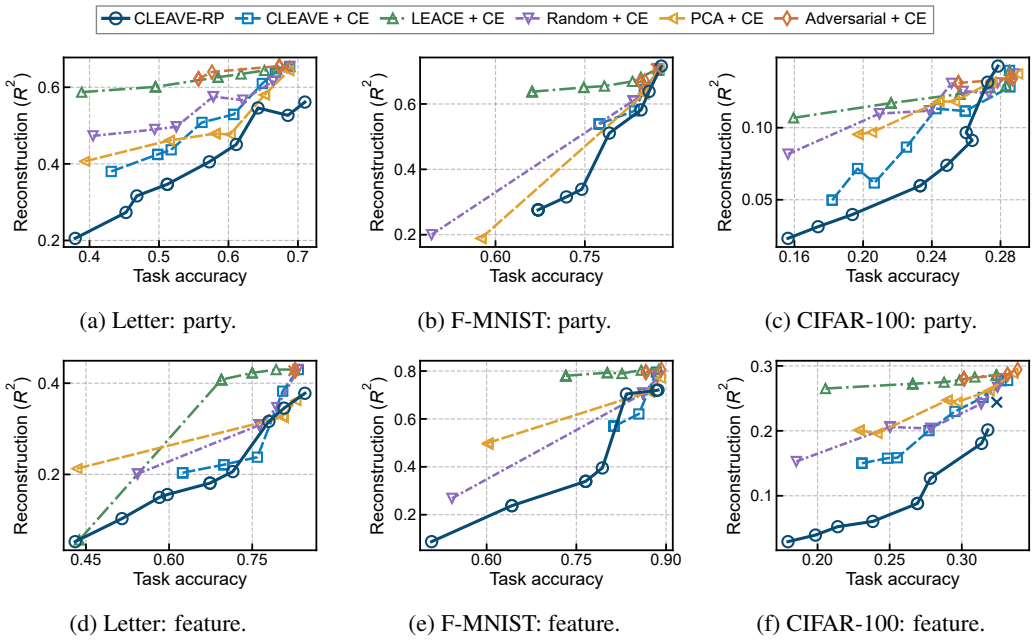
Figure 12: MLP recovery against LEACE and adversarial controls with shared CE-head training. Ten validation utility targets are shared; crosses mark unmet requirements.

On F-MNIST, the ordering changes across utility targets. Figure 13 supplies the corresponding test accuracy, allowing each recovery value to be read together with the task information retained by that release.

The complete target sequence preserves repeated selections and transitions between subspaces. It shows where an editor offers useful reconstruction reduction and where preserving utility leaves the compared releases close together.

## D.2 UTILITY–RECOVERY TRADEOFFS ACROSS DATASETS

Figure 13 pairs reconstruction with test accuracy for party and feature deletion on all three datasets. Traces follow validation-target order, including repeated candidates and marked unmet requirements.




(a) Letter: party. (b) F-MNIST: party. (c) CIFAR-100: party.

(d) Letter: feature. (e) F-MNIST: feature. (f) CIFAR-100: feature.

Figure 13: Adaptive reconstruction versus task accuracy. Lower right is preferable; crosses mark unmet validation utility targets.

Figure 13 shows the largest reconstruction reductions in parts of the party-deletion curves and the CIFAR-100 feature comparison. The curves converge near the high-utility end, where more reconstructible task information must remain. The joint position on both axes identifies the operating points at which subspace selection suppresses recovery while retaining predictive performance.

### D.3 Recovery of Original Numerical Fields

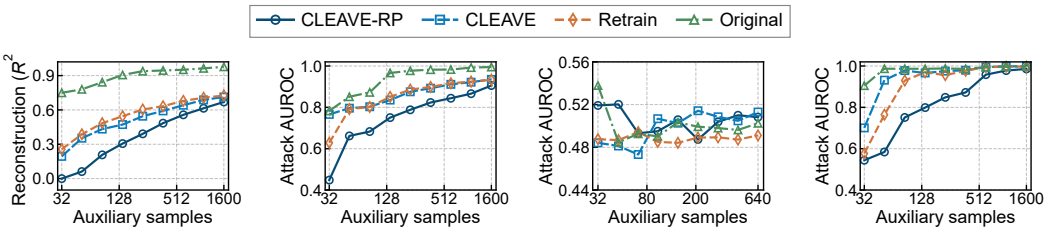
Figure 11(b) tests whether reduced representation recovery translates into difficulty reconstructing original numerical fields. Credit limit, bill amounts, and payment amounts share one set of axes, so the figure exposes both field-specific behavior and the task cost of suppressing it. The protected targets are standardized original fields. All target-aware editors receive the same raw-field objective, while PCA remains target-independent. Task AUROC measures the retained classification performance and is distinct from the reconstruction score on the horizontal axis.

Figure 11(b) shows a field-dependent tradeoff. CLEAVE-RP shifts releases toward lower reconstruction, with the accompanying task cost determined by how strongly each field supports prediction. Simpler projections occupy competitive regions at relaxed utility, while the shared axes identify which targets admit more selective suppression.

The original-field targets give the reconstruction score a direct interpretation, while the accompanying AUROC identifies the retained predictive value.

### D.4 Direct Target Recovery with Auxiliary Data

Figure 14 tests whether the surviving release exposes the requested data unit directly. The Letter targets are the departing party's raw fields, an indicator of a deleted attribute, deleted-member status, and the deleted-label indicator. Each panel uses nine nested auxiliary-data budgets with shared validation and test splits. The attacker is selected on validation data from the declared MLP and linear alternatives. Membership attacks additionally receive the known class label and observable loss and entropy statistics. CLEAVE-RP uses the validation-selected operating point at the retained-utility target of 0.9; the original, retrained and CLEAVE controls reuse their recorded releases and attack results.




(a) Party fields. (b) Deleted attribute. (c) Deleted members. (d) Deleted label.

Figure 14: Direct recovery as the attacker receives more auxiliary samples. Every marker is a measured budget. The membership panel uses a tighter vertical scale to resolve behavior near chance.

Figure 14 shows how auxiliary data changes direct recovery. Rank reduction lowers party-field and attribute recovery relative to the original release at larger budgets. Label inference grows with the attacker's data, while membership scores remain close to chance across the compared releases. At every budget from 32 to 1600 auxiliary samples, CLEAVE-RP gives the lowest party-field reconstruction and the lowest deleted-attribute and deleted-label AUROC of the four releases, and at 1600 samples every release approaches a label AUROC of 1.0. Each target therefore probes a different property of the surviving representation; embedding reconstruction and membership inference complement the direct field and label measurements.

### D.5 Independent Audits and Publication Faults

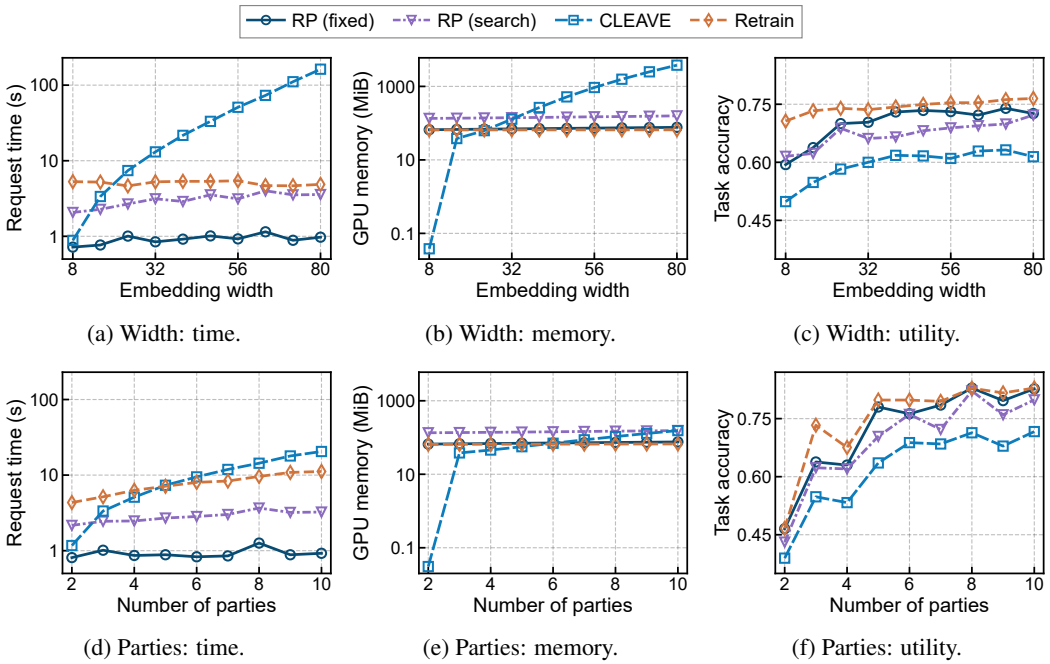
Audit variability is measured against known Gaussian population moments. A half-rank edit is fitted independently and then held fixed, each of nine audit sizes runs 1,000 independent trials with 399 bootstrap draws per trial, and the exact embedding-state leakage supplies the population reference for the declared covariance functional. Both the excess of the issued level over the exact leakage and the bootstrap sampling allowance decrease as the audit grows, a higher nominal confidence widens the allowance at the same sample size, and every issued level covers the population reference across the sweep.

Publication faults are evaluated on Letter in 200 trials with disjoint calibration and evaluation pools and nominal $97.5\%$ confidence for each one-sided bound. The clean release passes both checks. Artifact binding rejects every missing-adapter and stale-head release, while the statistical check detects every release exposing the old embedding cache. On its own, the statistical check rejects $6.5\%$ of missing-adapter releases and none of the stale-head releases. Their combination rejects all tested faulty releases: parameter identity detects substitutions, and leakage measurement detects information exposed by the published state.

### D.6 Deployment Cost and Scaling

Figure 15 separates applying a frozen configuration from selecting one through the full candidate search. The Letter party-deletion sweep covers ten embedding widths and nine federation sizes, reusing the corresponding trained base models. Each measured path includes projection, head fitting, publication and an empirical audit. Both RP paths share a runtime warm-up, and peak GPU allocation includes the resident original model. CLEAVE and retraining retain their archived measurements on the same H100 hardware class. Task accuracy accompanies resource use because the released models are not accuracy-matched.

Figure 15 shows the deployment benefit of reusing a selected RP configuration: both request time and peak GPU allocation fall relative to a full candidate search. RP (fixed) stays near 1 s per request at every width and federation size, whereas the CLEAVE request time rises past 100 s by width 80 and its peak allocation passes 1000 MiB at the largest widths. The search cost reflects selecting a reconstruction–utility operating point. Paired accuracy curves show how that selection changes with embedding width and federation size, allowing resource use and predictive performance to be assessed together. In the utility panels, retraining attains the highest task accuracy at every embedding width, and RP (fixed) exceeds CLEAVE in task accuracy at every width and federation size.




(a) Width: time. (b) Width: memory. (c) Width: utility.

(d) Parties: time. (e) Parties: memory. (f) Parties: utility.

Figure 15: Request cost, peak GPU memory and test accuracy across embedding width and federation size. RP timings use shared warm-up; CLEAVE and Retrain are archived controls. Time and memory axes are logarithmic.

# E Proofs of Theoretical Results

## E.1 Assumptions and Scope

Condition on the retained encoders. In the geometric results, $H \in \mathbb{R}^d$, $Z \in \mathbb{R}^p$ and $y \in \mathbb{R}^q$ are centered, $\Sigma = \Sigma_{HH} \succ 0$, and the energies $s_Z = \mathrm{Tr}(\Sigma_{ZZ})$ and $s_y = \mathrm{Tr}(\Sigma_{yy})$ are positive. Write $C = \Sigma_{HZ}$, $B = \Sigma_{Hy}$, $V = \Sigma^{-1/2}C$, and $U = \Sigma^{-1/2}B$. These results concern unrestricted linear least-squares decoders and the second-order utility in Eq. (5); preservation of this utility does not assert unchanged classification accuracy. The same algebra holds for fixed finite features and empirical covariances, with hats distinguishing sample quantities from population quantities below. The closed-form floors refer to the exact-utility level $\tau_0 = \mathcal{U}(I)$; other levels retain the constrained definition in Eq. (8).

For a norm-bounded decoder class, the trace formulas apply when the least-squares optimizers lie inside the declared ball. Otherwise the appropriate leakage is the normalized reduction in prediction risk, defined in Eq. (48), and its certificate is proved separately below. This distinction also separates a finite feature dictionary from an entire kernel class.

## E.2 Proof of Lemma 1 (Decoder Dictionary)

**Lemma 1** (Decoder dictionary)**.** *For an edit $G$, let $T_G$ project orthogonally onto* $\mathrm{col}(\Sigma^{1/2}G^\top)$*. Then $s_Z\mathcal{L}(G) = \mathrm{Tr}(V^\top T_G V)$ and $s_y\mathcal{U}(G) = \mathrm{Tr}(U^\top T_G U)$. Every orthogonal projector is realizable by an edit. In particular, invertible edits preserve both quantities.*

*Proof.* Put $K_G = G\Sigma G^\top$. Expanding the decoder's squared error gives

$$\begin{aligned}\mathbb{E}\|Z - WGH\|^2 &= s_Z - 2\,\mathrm{Tr}(WGC) + \mathrm{Tr}(WK_GW^\top)\\ &= s_Z - \mathrm{Tr}(C^\top G^\top K_G^+ GC) + \|(W - W_G)K_G^{1/2}\|_F^2,\end{aligned} \tag{12}$$

where $W_G = C^\top G^\top K_G^+$. To justify this completion of the square without assuming that $K_G$ is invertible, take $a \in \ker K_G$. Centering gives

$$\mathbb{E}(a^\top GH)^2 = a^\top K_G a = 0, \qquad a^\top GH = 0 \text{ almost surely}, \qquad a^\top GC = \mathbb{E}[a^\top GHZ^\top] = 0. \tag{13}$$

Consequently $\mathrm{col}(GC) \subseteq \mathrm{col}(K_G)$, so $K_G K_G^+ GC = GC$ and $W_G K_G = C^\top G^\top$. For $D = W - W_G$, the terms linear in $D$ cancel explicitly:

$$\begin{aligned} &-2\,\mathrm{Tr}(DGC) + \mathrm{Tr}(DK_G W_G^\top) + \mathrm{Tr}(W_G K_G D^\top) \\ &\quad = -2\,\mathrm{Tr}(DGC) + \mathrm{Tr}(DGC) + \mathrm{Tr}(DGC) = 0. \end{aligned} \tag{14}$$

The remaining term is $\mathrm{Tr}(DK_G D^\top) = \|DK_G^{1/2}\|_F^2 \geq 0$. All minimizers have the form $W_G + N$ with $NK_G^{1/2} = 0$; they give the same prediction almost surely. Thus the fitted decoder energy is well-defined even when its coefficient matrix is not unique:

$$\begin{aligned} \mathbb{E}\|W_G GH\|^2 &= \mathrm{Tr}(W_G K_G W_G^\top) = \mathrm{Tr}(C^\top G^\top K_G^+ K_G K_G^+ GC) \\ &= \mathrm{Tr}(C^\top G^\top K_G^+ GC) = s_Z - \min_W \mathbb{E}\|Z - WGH\|^2. \end{aligned} \tag{15}$$

For $\Psi = \Sigma^{1/2} G^\top$, its thin singular value decomposition $\Psi = L\Lambda R^\top$ on the nonzero singular directions yields

$$\begin{aligned} \Sigma^{1/2} G^\top K_G^+ G \Sigma^{1/2} &= \Psi(\Psi^\top \Psi)^+ \Psi^\top \\ &= L\Lambda R^\top (R\Lambda^2 R^\top)^+ R\Lambda L^\top = LL^\top = T_G. \end{aligned} \tag{16}$$

Substitute $C = \Sigma^{1/2} V$ in Eq. (15); replacing $C$ by $B$ proves the utility identity. If $\Psi$ has rank zero, both its pseudoinverse expression and $T_G$ are zero, so the same identity applies. Conversely, for any orthogonal projector $T$, choose $G = T\Sigma^{-1/2}$. Then $\Sigma^{1/2} G^\top = T$, whose column-space projector is $T$ itself. Since $0 \preceq T_G \preceq I$, neither quantity increases; invertible $G$ gives $T_G = I$. In particular,

$$\rho_k = \mathcal{L}(I) = \frac{\mathrm{Tr}(C^\top \Sigma^{-1} C)}{s_Z} = \frac{\mathrm{Tr}(V^\top V)}{s_Z} = \frac{\|V\|_F^2}{s_Z}. \tag{17}$$

Its empirical counterpart is $\hat{\rho}_k$ in Eq. (2). □

### E.3 Proof of Theorem 2 (Two Floors)

**Theorem 2** (Two floors)**.** *At $\tau_0 = \mathcal{U}(I)$, the centralized floor is $\varepsilon^C(\tau_0) = \|\Pi_U V\|_F^2 / s_Z$, where $\Pi_U$ projects onto $\mathrm{col}(U)$. It is attained, and the removable share is $\|(I - \Pi_U)V\|_F^2 / s_Z$. A mechanism exposing the same unedited $H$ retains leakage $\rho_k$ from that channel, irrespective of its read-out.*

*Proof.* For $T = T_G$, symmetry and idempotence give

$$\begin{aligned} s_y\big(\tau_0 - \mathcal{U}(G)\big) &= \mathrm{Tr}(U^\top U) - \mathrm{Tr}(U^\top T U) \\ &= \mathrm{Tr}\left(U^\top (I-T)^\top (I-T) U\right) = \|(I-T)U\|_F^2. \end{aligned} \tag{18}$$

Consequently exact utility is equivalent to $TU = U$: the squared Frobenius norm vanishes if and only if each column of $U$ lies in $\mathrm{col}(T)$. Therefore $T$ acts as the identity on $\mathrm{col}(U)$, giving $T\Pi_U = \Pi_U$; transposition also gives $\Pi_U T = \Pi_U$. In particular,

$$(T - \Pi_U)^2 = T - T\Pi_U - \Pi_U T + \Pi_U = T - \Pi_U, \qquad (T - \Pi_U)\Pi_U = 0. \tag{19}$$

Together with symmetry, these identities show that $T' = T - \Pi_U$ is an orthogonal projector onto directions orthogonal to the task subspace. The two projected components of $V$ are orthogonal in the Frobenius inner product, and

$$\begin{aligned} s_Z \mathcal{L}(G) &= \mathrm{Tr}\left(V^\top (\Pi_U + T') V\right) \\ &= \|\Pi_U V\|_F^2 + \|T'V\|_F^2 \geq \|\Pi_U V\|_F^2 = s_Z \varepsilon^{\mathrm{C}}(\tau_0). \end{aligned} \tag{20}$$

Equality is attained by $G = \Pi_U \Sigma^{-1/2}$, for which $T' = 0$. To connect this result to RS in the original coordinates, let $\Pi$ project onto $\mathrm{col}(\Sigma^{-1} B)$, the coefficient space of the best label predictor. Then

$$\begin{aligned} \mathrm{col}(\Sigma^{1/2}\Pi) &= \mathrm{col}(\Sigma^{-1/2} B) = \mathrm{col}(U), \\ P = \Pi(\Pi\Sigma\Pi)^+\Pi &= \Sigma^{-1/2}\Pi_U \Sigma^{-1/2}. \end{aligned} \tag{21}$$

For completeness, the second identity in Eq. (21) follows without commuting $\Pi$ and $\Sigma$. Let $R$ have orthonormal columns spanning $\mathrm{col}(\Sigma^{-1}B)$, so $\Pi = RR^\top$, and put $D = R^\top \Sigma R \succ 0$. Then

$$\begin{aligned}(\Pi\Sigma\Pi)^+ &= (RDR^\top)^+ = RD^{-1}R^\top,\\ \Pi_U &= (\Sigma^{1/2}R)\big((\Sigma^{1/2}R)^\top\Sigma^{1/2}R\big)^{-1}(\Sigma^{1/2}R)^\top\\ &= \Sigma^{1/2}RD^{-1}R^\top\Sigma^{1/2},\\ \Sigma^{-1/2}\Pi_U\Sigma^{-1/2} &= RD^{-1}R^\top = \Pi(\Pi\Sigma\Pi)^+\Pi.\end{aligned} \tag{22}$$

Here $D \succ 0$ because $R$ has full column rank and $\Sigma \succ 0$. If $B = 0$, the task subspace is zero and both sides are zero by definition. Thus the task-restricted decoder has leakage $\varepsilon^{\mathrm{C}}(\tau_0)$ and Eq. (3) becomes

$$\begin{aligned}\rho_k - \varepsilon^{\mathrm{C}}(\tau_0) &= \frac{\mathrm{Tr}\left(C^\top(\Sigma^{-1} - P)C\right)}{s_Z}\\ &= \frac{\mathrm{Tr}\left(V^\top(I - \Pi_U)V\right)}{s_Z} = \frac{\|(I - \Pi_U)V\|_F^2}{s_Z} \geq 0.\end{aligned} \tag{23}$$

If the unedited $H$ remains observable, Lemma 1 with $G = I$ gives $\rho_k$ exactly for that channel; additional surviving channels can only increase the best achievable reconstruction. This is a statement about a fixed retained representation, not an invariance theorem for encoders changed by retraining. □

For a general feasible $0 \leq \tau \leq \tau_0$, the correct centralized benchmark is

$$\varepsilon^{\mathrm{C}}(\tau) = \min_{\substack{T=T^\top=T^2\\ \mathrm{Tr}(U^\top TU)\geq\tau s_y}} \frac{\mathrm{Tr}(V^\top TV)}{s_Z} \leq \varepsilon^{\mathrm{C}}(\tau_0). \tag{24}$$

The minimum exists: symmetric idempotent matrices form a closed bounded subset of the finite-dimensional matrix space, the utility inequality defines a closed subset, and $T = I$ makes it nonempty for $\tau \leq \tau_0$. The objective is continuous. The feasible sets are nested as $\tau$ increases, so $\varepsilon^{\mathrm{C}}(\tau)$ is nondecreasing and $\varepsilon^{\mathrm{C}}(0) = 0$, attained by $T = 0$. The exact-utility closed form is therefore not a lower bound at a relaxed utility bar. A norm constraint on the decoder likewise requires recomputing the benchmark if it excludes the least-squares optimizer.

### E.4 Proof of Theorem 3 (Surcharge)

**Theorem 3** (Surcharge)**.** *For a fixed representation and feasible utility level, $0 \leq S(\tau) \leq \rho_k - \varepsilon^{C}(\tau)$; with one freely editable block, $S(\tau) = 0$. At exact utility, put $W = \Sigma^{-1}B$ and $D = \mathrm{blkdiag}((\Pi_{W_c})_c)$, where $\Pi_{W_c}$ projects onto the coefficient space in block $c$. For any $\Sigma \succ 0$, $D$ attains $\varepsilon^{B}(\tau_0) = \mathcal{L}(D)$. When $\Sigma$ is block diagonal, this reduces to $\sum_c \|\Pi_{U_c}V_c\|_F^2/s_Z$, and the scalar-label surcharge is a weighted variance across task-active blocks.*

*Proof.* The identity is feasible and the block-diagonal family is a subset of all edits, giving

$$\varepsilon^{\mathrm{C}}(\tau) \leq \varepsilon^{\mathrm{B}}(\tau) \leq \mathcal{L}(I) = \rho_k, \qquad S(\tau) = \varepsilon^{\mathrm{B}}(\tau) - \varepsilon^{\mathrm{C}}(\tau) \geq 0. \tag{25}$$

With one block the feasible families coincide. For general correlated blocks, Eq. (18) makes exact utility equivalent to

$$T_A U = U \iff \mathrm{col}(\Sigma^{-1}B) \subseteq \mathrm{col}(A^\top) \iff \mathrm{col}(W_c) \subseteq \mathrm{col}(A_c^\top) \quad \text{for every } c. \tag{26}$$

The first equivalence follows by multiplying the subspace inclusion $\mathrm{col}(U) \subseteq \mathrm{col}(\Sigma^{1/2}A^\top)$ by $\Sigma^{-1/2}$; the second uses the direct sum of the local row spaces. Hence every feasible local edit contains $\mathrm{col}(D)$ in its row space. Multiplication by the invertible $\Sigma^{1/2}$ preserves inclusion, so $T_D \preceq T_A$. The edit $D$ itself preserves $W$, and therefore attains the minimum:

$$\varepsilon^{\mathrm{B}}(\tau_0) = \frac{\mathrm{Tr}\big(C^\top D(D\Sigma D)^+ DC\big)}{s_Z}, \qquad S(\tau_0) = \frac{\|(T_D - \Pi_U)V\|_F^2}{s_Z}. \tag{27}$$

Here $T_D - \Pi_U$ is an orthogonal projector because $T_D$ contains the task subspace. This construction retains the off-diagonal covariance blocks. If a block is fixed to the identity, set its block of $D$ to the identity as well.

When $\Sigma$ is block diagonal, $T_A = \mathrm{blkdiag}((T_c)_c)$ and every such family of projectors is realizable locally. Eq. (18) then reads

$$s_y\big(\tau_0 - \mathcal{U}(A)\big) = \|(I - T_A)U\|_F^2 = \sum_c \|(I - T_c)U_c\|_F^2, \tag{28}$$

which vanishes precisely when every summand vanishes. Applying the preceding theorem block by block yields

$$\begin{aligned} s_Z\mathcal{L}(A) &= \sum_c \mathrm{Tr}(V_c^\top T_c V_c) \\ &= \sum_c \big(\|\Pi_{U_c} V_c\|_F^2 + \|(T_c - \Pi_{U_c})V_c\|_F^2\big) \geq s_Z\varepsilon^{\mathrm{B}}(\tau_0). \end{aligned} \tag{29}$$

Equality is attained by $A_c = \Pi_{U_c}\Sigma_{cc}^{-1/2}$. Indeed, block diagonality of $\Sigma$ makes $\Sigma^{1/2}A^\top$ block diagonal as well, and the column-space projector of its $c$-th block is exactly $\Pi_{U_c}$. Thus this construction realizes all local minimizers simultaneously, rather than merely bounding their individual objectives. This closed form assumes that every block in this decomposition can be edited. If a block is fixed to the identity, its projector is fixed to $I_{d_c}$ and its contribution to the block floor is $\|V_c\|_F^2/s_Z$ instead. Likewise, the one-block equality requires that the sole block be freely editable.

For scalar $y$, write $u_c = U_c$, $w_c = \|u_c\|^2$, and $a_{c,j} = u_c^\top v_{c,j}$, where $v_{c,j}$ is column $j$ of $V_c$. Sum only over $w_c > 0$, and put $\bar{r}_j = (\sum_c a_{c,j})/(\sum_c w_c)$. If all $w_c$ vanish, both exact-utility floors are zero. Otherwise,

$$\begin{aligned} s_Z S(\tau_0) &= \sum_j \left( \sum_c \frac{a_{c,j}^2}{w_c} - \frac{(\sum_c a_{c,j})^2}{\sum_c w_c} \right) \\ &= \sum_j \sum_c w_c \left( \frac{a_{c,j}}{w_c} - \bar{r}_j \right)^2 \geq 0. \end{aligned} \tag{30}$$

To see the weighted-variance step explicitly, define $W_0 = \sum_c w_c > 0$ and $r_{c,j} = a_{c,j}/w_c$. Then $\sum_c w_c r_{c,j} = W_0 \bar{r}_j$, and

$$\begin{aligned} \sum_c w_c (r_{c,j} - \bar{r}_j)^2 &= \sum_c w_c r_{c,j}^2 - 2\bar{r}_j \sum_c w_c r_{c,j} + \bar{r}_j^2 W_0 \\ &= \sum_c \frac{a_{c,j}^2}{w_c} - W_0 \bar{r}_j^2 = \sum_c \frac{a_{c,j}^2}{w_c} - \frac{(\sum_c a_{c,j})^2}{W_0}. \end{aligned} \tag{31}$$

The centralized expression uses the stacked vector $u = (u_c)_c$, for which $\|u\|^2 = W_0$ and $u^\top v_j = \sum_c a_{c,j}$. The local expression uses the separate rank-one projectors $u_c u_c^\top / w_c$. This accounts for both terms in the first line of Eq. (30). Equivalently, each coordinate obeys

$$\left( \sum_c a_{c,j} \right)^2 = \left( \sum_c \frac{a_{c,j}}{\sqrt{w_c}} \sqrt{w_c} \right)^2 \leq \left( \sum_c \frac{a_{c,j}^2}{w_c} \right) \left( \sum_c w_c \right). \tag{32}$$

Equality holds exactly when $a_{c,j}/w_c$ agrees across the task-active blocks for each $j$. Blocks with $U_c = 0$ contribute zero to the floor and need no division by zero. □

Refining a fixed partition restricts the feasible edits and cannot lower the block floor; adding new features or changing the data distribution is not this comparison. For any independent edit $A^{\mathrm{ind}}$ satisfying the same utility constraint, its gap $G = \mathcal{L}(A^{\mathrm{ind}}) - \varepsilon^{\mathrm{B}}(\tau)$ is nonnegative. This does not order arbitrary approximate solvers.

### E.5 Derivation of the Coordinated Edit

Let $J(A) = \sum_c \|(A_c - I)\Sigma_{cc}^{1/2}\|_F^2$, $M = CC^\top$ and $N = BB^\top$. Freezing the inverse gives the surrogate functionals $\bar{\mathcal{L}}(A) = \mathrm{Tr}(A^\top\Sigma^{-1}AM)/s_Z$ and $\bar{\mathcal{U}}(A) = \mathrm{Tr}(A^\top\Sigma^{-1}AN)/s_y$. Absorb the two normalizers into the multipliers $\mu, \nu \geq 0$ and set $K = \mu M - \nu N$. The Lagrangian expands as

$$\begin{aligned} \mathcal{J}(A, \mu, \nu) &= J(A) + \mu s_Z(\bar{\mathcal{L}}(A) - \varepsilon) + \nu s_y(\tau - \bar{\mathcal{U}}(A)) \\ &= J(A) + \mathrm{Tr}(A^\top\Sigma^{-1}AK) - \mu s_Z\varepsilon + \nu s_y\tau. \end{aligned} \tag{33}$$

We use the Frobenius convention $\mathrm{d}f = \mathrm{Tr}((\nabla_A f)^\top \mathrm{d}A)$. Writing $S = \Sigma^{-1}$, symmetry of $S$ and $K$ gives

$$\begin{aligned} \mathrm{d}\,\mathrm{Tr}(A^\top SAK) &= \mathrm{Tr}((\mathrm{d}A)^\top SAK) + \mathrm{Tr}(A^\top S(\mathrm{d}A)K) \\ &= 2\,\mathrm{Tr}((SAK)^\top \mathrm{d}A), \\ \mathrm{d}J(A) &= 2\sum_c \mathrm{Tr}\left(((A_c - I)\Sigma_{cc})^\top \mathrm{d}A_c\right). \end{aligned} \tag{34}$$

Only diagonal blocks of $\mathrm{d}A$ are admissible. Since $A_{bb} = A_b$ and its off-diagonal blocks vanish, $(SAK)_{cc} = \sum_b S_{cb}A_bK_{bc}$. Restricting the differential accordingly gives

$$\begin{aligned} \tfrac{1}{2}\nabla_{A_c}\mathcal{J} &= (A_c - I)\Sigma_{cc} + (\Sigma^{-1}AK)_{cc} \\ &= (A_c - I)\Sigma_{cc} + \sum_b (\Sigma^{-1})_{cb}A_b(\mu M_{bc} - \nu N_{bc}) = 0. \end{aligned} \tag{35}$$

This is Eq. (7). Vectorizing each $A_c$ turns it into a block linear system with block $(c, b)$ equal to

$$\mathcal{H}_{cb} = \mathbf{1}_{\{c=b\}}(\Sigma_{cc} \otimes I_{d_c}) + K_{bc}^\top \otimes (\Sigma^{-1})_{cb}, \qquad b_c = \mathrm{vec}(\Sigma_{cc}). \tag{36}$$

Here vectorization stacks columns: $\mathrm{vec}(LXR) = (R^\top \otimes L)\,\mathrm{vec}(X)$. In particular, $\mathrm{vec}(A_c\Sigma_{cc}) = (\Sigma_{cc} \otimes I_{d_c})\,\mathrm{vec}(A_c)$ and

$$\mathrm{vec}(S_{cb}A_bK_{bc}) = (K_{bc}^\top \otimes S_{cb})\,\mathrm{vec}(A_b). \tag{37}$$

Thus $\mathcal{H}_{cb}$ has size $d_c^2 \times d_b^2$, and $\mathcal{H}_{bc}^\top = \mathcal{H}_{cb}$. Let $a$ stack the vectors $\mathrm{vec}(A_c)$ and let $b$ stack $b_c$. Up to constants independent of $a$, the Lagrangian is $a^\top \mathcal{H}a - 2b^\top a$. For a stationary solution $\mathcal{H}a_\star = b$, expansion gives

$$\mathcal{J}(a, \mu, \nu) - \mathcal{J}(a_\star, \mu, \nu) = (a - a_\star)^\top \mathcal{H}(a - a_\star). \tag{38}$$

Consequently positive semidefiniteness suffices for global minimization of this fixed-multiplier quadratic, and positive definiteness makes the minimizer unique. Let $A_\star$ additionally be surrogate-feasible and satisfy

$$\mu s_Z(\bar{\mathcal{L}}(A_\star) - \varepsilon) = 0, \qquad \nu s_y(\tau - \bar{\mathcal{U}}(A_\star)) = 0. \tag{39}$$

These equalities make $\mathcal{J}(A_\star, \mu, \nu) = J(A_\star)$. At any other feasible $A$, both constraint residuals in the Lagrangian are nonpositive and both multipliers are nonnegative, so $\mathcal{J}(A, \mu, \nu) \le J(A)$. Combining the two facts gives

$$\begin{aligned} J(A_\star) = \mathcal{J}(A_\star, \mu, \nu) &= \min_{\tilde{A} \in \mathcal{A}} \mathcal{J}(\tilde{A}, \mu, \nu) \\ &\le \mathcal{J}(A, \mu, \nu) \le J(A). \end{aligned} \tag{40}$$

Thus the positive-semidefinite quadratic, feasibility and complementary slackness together certify the surrogate optimum. The deployed candidate is accepted using the evaluated leakage and utility constraints in Eq. (6).

The true leakage still requires Eq. (4). With $\bar{Q}_A = A^\top \Sigma^{-1} A$, its discrepancy obeys

$$\begin{aligned} \left|\mathcal{L}(A) - \bar{\mathcal{L}}(A)\right| &= \frac{|\,\mathrm{Tr}(C^\top (Q_A - \bar{Q}_A)C)|}{s_Z} \\ &\le \frac{\|Q_A - \bar{Q}_A\|_{\mathrm{op}} \|C\|_F^2}{s_Z}. \end{aligned} \tag{41}$$

To verify the trace bound, set $D_A = Q_A - \bar{Q}_A$ and write $C = (c_1, \ldots, c_p)$. Symmetry gives $|c_j^\top D_A c_j| \le \|D_A\|_{\mathrm{op}} \|c_j\|^2$, whence

$$|\,\mathrm{Tr}(C^\top D_A C)| \le \sum_j |c_j^\top D_A c_j| \le \|D_A\|_{\mathrm{op}} \sum_j \|c_j\|^2. \tag{42}$$

Replacing $C, s_Z$ by $B, s_y$ gives the corresponding utility discrepancy. The sample version bounds $\Delta^{\mathrm{r}}$ when a surrogate is used as the certificate center; exact audit evaluation sets it to zero. There is no general sign for the discrepancy: $A = \alpha I$ with $\alpha \neq 0$ gives $\mathcal{L}(A) = \rho_k$ but $\bar{\mathcal{L}}(A) = \alpha^2 \rho_k$. Accordingly the protocol checks the actual leakage and utility before accepting a candidate, and reports an unmet target if its search fails. Under unrestricted linear decoding, reducing leakage requires discarding informative directions, not merely invertibly scaling them.

### E.6 Derivation of the Exact Read-Out Downdate

Here $\tilde{H}$ is a sample-by-feature matrix, $\lambda > 0$, $\Gamma = \tilde{H}^\top \tilde{H} + \lambda I$, and $F = \tilde{H}_{\mathcal{I}}$. The full-data objective is $\|\tilde{H}\theta - y\|_F^2 + \lambda\|\theta\|_F^2$, whose derivative is $2\Gamma\theta - 2\tilde{H}^\top y$. Its unique minimizer therefore satisfies $\Gamma\theta' = \tilde{H}^\top y$. Keep the edited features fixed while removing the rows in $\mathcal{I}$, as in Eq. (10). Write $E_{\mathcal{I}} = I - F\Gamma^{-1}F^\top$ and $\Gamma_- = \Gamma - F^\top F$. Since $\Gamma_- = \tilde{H}_{-\mathcal{I}}^\top \tilde{H}_{-\mathcal{I}} + \lambda I \succ 0$, conjugating by $\Gamma^{-1/2}$ gives

$$I - D^\top D = \Gamma^{-1/2}\Gamma_-\Gamma^{-1/2} \succ 0, \qquad D = F\Gamma^{-1/2}. \tag{43}$$

Every singular value of $D$ is strictly below one. Hence $E_{\mathcal{I}} = I - DD^\top \succ 0$, including any extra eigenvalues equal to one if $D$ is rectangular. The inverse update can be checked directly, without assuming it: multiplication of the claimed right-hand side by $\Gamma_-$ gives

$$\begin{aligned} &\Gamma_-\big(\Gamma^{-1} + \Gamma^{-1}F^\top E_{\mathcal{I}}^{-1} F\Gamma^{-1}\big) \\ &= I - F^\top F\Gamma^{-1} + F^\top (I - F\Gamma^{-1}F^\top) E_{\mathcal{I}}^{-1} F\Gamma^{-1} = I. \end{aligned} \tag{44}$$

This proves the Woodbury identity in the present setting:

$$(\Gamma - F^\top F)^{-1} = \Gamma^{-1} + \Gamma^{-1}F^\top E_{\mathcal{I}}^{-1} F\Gamma^{-1}. \tag{45}$$

For the last step of the downdate, the same multiplication gives the useful identity

$$\Gamma_-\Gamma^{-1}F^\top = F^\top(I - F\Gamma^{-1}F^\top) = F^\top E_{\mathcal{I}}, \qquad \Gamma_-^{-1}F^\top = \Gamma^{-1}F^\top E_{\mathcal{I}}^{-1}. \tag{46}$$

The retained cross-product is $\tilde{H}_{-\mathcal{I}}^\top y_{-\mathcal{I}} = \Gamma\theta' - F^\top y_{\mathcal{I}}$. Substitution into its normal equations now yields

$$\begin{aligned} \theta_{-\mathcal{I}} &= (\Gamma - F^\top F)^{-1}(\Gamma\theta' - F^\top y_{\mathcal{I}}) \\ &= \theta' + (\Gamma - F^\top F)^{-1} F^\top (F\theta' - y_{\mathcal{I}}) \\ &= \theta' + \Gamma^{-1}F^\top E_{\mathcal{I}}^{-1}(F\theta' - y_{\mathcal{I}}). \end{aligned} \tag{47}$$

This proves exact equality to refitting the ridge read-out with the same fixed adapters and the same $\lambda$. It does not assert equality to retraining the bottom models, or an influence-function approximation for a nonlinear head. The row normalization of the ridge objective must also remain unchanged.

### E.7 Proof of Lemma 4 (Sampling and Approximation Slacks)

Fix the deployed state and its audit feature map before drawing $n'$ independent audit records. Let $X_A \in \mathbb{R}^{d_A}$ be the declared feature representation of that state, with $\|X_A\| \le B_X$, $\|Z\| \le B_Z$, and $s_Z \ge s_0 > 0$. Centering parameters are also fixed before this draw. For the declared ball $\mathcal{F}_D = \{x \mapsto Wx : \|W\|_F \le \beta\}$, define

$$\mathcal{L}_D = 1 - \frac{R_D}{s_Z}, \qquad R_D = \inf_{f \in \mathcal{F}_D} \mathbb{E}\|Z - f(X_A)\|^2, \qquad \widehat{\mathcal{L}}_D = 1 - \frac{\widehat{R}_D}{\hat{s}_Z}. \tag{48}$$

Here $\widehat{R}_D$ and $\hat{s}_Z$ replace expectations by audit averages; the infimum is over the same fixed ball. Both class optima include the zero decoder. If the least-squares solutions are in the ball, Lemma 1 makes this definition exactly Eq. (4). When the constraint binds, the constrained risks must be used instead; any certified optimization error is charged to $\Delta^{\mathrm{r}}$.

**Lemma 4** (Sampling and approximation slacks). *For $0 < \xi \le \beta$, set $L_0 = (B_Z + \beta B_X)^2$ and*

$$\begin{aligned} a_{n'} &= L_0\sqrt{\frac{pd_A\log(1 + 2\beta/\xi) + \log(4/\delta_e)}{2n'}} + 4B_X(B_Z + \beta B_X)\xi, \\ b_{n'} &= B_Z^2\sqrt{\frac{\log(4/\delta_e)}{2n'}}. \end{aligned} \tag{49}$$

*If $b_{n'} \le s_0/2$, then with probability at least $1 - \delta_e$,*

$$|\mathcal{L}_D - \widehat{\mathcal{L}}_D| \le \Delta^e = \frac{2(a_{n'} + b_{n'})}{s_0}. \tag{50}$$

*For a certificate issued for this fixed dictionary, $\Delta^a = 0$. Extending it to a kernel class requires a separate approximation event as specified below.*

*Proof.* A Frobenius $\xi$-net of the radius-$\beta$ ball has at most $(1 + 2\beta/\xi)^{pd_A}$ elements. For two decoder matrices in the ball, the pointwise loss difference expands as

$$\begin{aligned}\left|\|Z - WX_A\|^2 - \|Z - W'X_A\|^2\right| &= \left|\langle (W' - W)X_A,\, 2Z - (W + W')X_A\rangle\right| \\ &\le 2B_X(B_Z + \beta B_X)\|W - W'\|_F.\end{aligned} \tag{51}$$

Write $\mathcal{N}_\xi$ for this net and $N_\xi = |\mathcal{N}_\xi|$. One construction is a maximal $\xi$-separated subset of the ball: maximality gives coverage, and disjoint balls of radius $\xi/2$ around its points lie in the ball of radius $\beta + \xi/2$. Comparing volumes in dimension $pd_A$ proves the stated cardinality bound. For each fixed net element the squared loss lies in $[0, L_0]$, so Hoeffding's inequality and a union bound give

$$\Pr\left\{\max_{W \in \mathcal{N}_\xi} |\widehat{R}(W) - R(W)| > t\right\} \le 2N_\xi \exp\left(-\frac{2n't^2}{L_0^2}\right). \tag{52}$$

Taking $t = L_0\sqrt{[pd_A \log(1 + 2\beta/\xi) + \log(4/\delta_e)]/(2n')}$ makes this at most $\delta_e/2$. For an arbitrary $W$, choose $W_0 \in \mathcal{N}_\xi$ within distance $\xi$. With $L_{\mathrm{lip}} = 2B_X(B_Z + \beta B_X)$, Eq. (51) gives

$$\begin{aligned}|\widehat{R}(W) - R(W)| &\le |\widehat{R}(W) - \widehat{R}(W_0)| + |\widehat{R}(W_0) - R(W_0)| + |R(W_0) - R(W)| \\ &\le L_{\mathrm{lip}}\xi + t + L_{\mathrm{lip}}\xi = a_{n'}.\end{aligned} \tag{53}$$

The bound is simultaneous over the entire ball, so it also controls the data-dependent empirical minimizer. More explicitly, for every $W$ we have $R(W) - a_{n'} \le \widehat{R}(W) \le R(W) + a_{n'}$; taking infima in these inequalities yields $R_D - a_{n'} \le \widehat{R}_D \le R_D + a_{n'}$. No assumption that the population and empirical minimizers coincide is needed.

For the energy denominator, $0 \le \|Z\|^2 \le B_Z^2$ gives

$$\Pr\{|\hat{s}_Z - s_Z| > b_{n'}\} \le 2\exp\left(-\frac{2n'b_{n'}^2}{B_Z^4}\right) = \delta_e/2. \tag{54}$$

On the intersection of the two events, whose probability is at least $1 - \delta_e$, $\hat{s}_Z \ge s_Z - b_{n'} \ge s_0/2$. The zero decoder belongs to the ball, hence $0 \le R_D \le s_Z$ and $0 \le \widehat{R}_D \le \hat{s}_Z$. Therefore both normalized leakages lie in $[0, 1]$, and their difference satisfies

$$\begin{aligned}|\mathcal{L}_D - \widehat{\mathcal{L}}_D| &= \left|\frac{\widehat{R}_D}{\hat{s}_Z} - \frac{R_D}{s_Z}\right| \\ &\le \frac{|\widehat{R}_D - R_D|}{\hat{s}_Z} + \frac{R_D}{s_Z}\frac{|s_Z - \hat{s}_Z|}{\hat{s}_Z} \le \frac{2(a_{n'} + b_{n'})}{s_0}.\end{aligned} \tag{55}$$

For fixed dimension and bounds, choosing $\xi = \beta/n'$ gives order $\sqrt{\log(n'/\delta_e)/n'}$; no minimax lower bound is needed for soundness. □

**Kernel extension.** Suppose an event of probability at least $1 - \delta_a$ pairs every decoder in the declared kernel ball with a decoder in $\mathcal{F}_D$, and conversely, with $L^2$ prediction distance at most $e_D$ and both prediction norms at most $B_f$. For a matched pair, Cauchy–Schwarz gives

$$\begin{aligned}|R(f) - R(g)| &= |\mathbb{E}\langle g - f,\, 2Z - f - g\rangle| \\ &\le \|f - g\|_{L^2}(2\|Z\|_{L^2} + \|f\|_{L^2} + \|g\|_{L^2}) \le 2(B_Z + B_f)e_D.\end{aligned} \tag{56}$$

To justify taking infima, write $R_K$ for the kernel-class optimum and $c_D = 2(B_Z + B_f)e_D$. For any $\eta > 0$, choose a kernel decoder $f_\eta$ with $R(f_\eta) \le R_K + \eta$ and its matched finite decoder $g_\eta$. Then $R_D \le R(g_\eta) \le R_K + \eta + c_D$. Reversing the matching gives $R_K \le R_D + \eta + c_D$. Letting $\eta \downarrow 0$ proves $|R_K - R_D| \le c_D$ even if an infimum is not attained. Dividing by the common $s_Z \ge s_0$ proves $|\mathcal{L} - \mathcal{L}_D| \le \Delta^{\mathrm{a}} = 2(B_Z + B_f)e_D/s_0$. The event must cover the selected edit, uniformly if that edit depends on the feature seed. Random-feature kernel approximation (Rahimi & Recht, 2007) alone does not establish this two-sided approximation of decoder balls. In its absence, the certificate names only the fixed finite dictionary.

### E.8 Proof of Theorem 5 (Two-Sided Soundness)

**Theorem 5** (Two-sided soundness)**.** *Under Lemma 4, let $\widehat{\mathcal{L}}$ be the reported certificate center and require $|\widehat{\mathcal{L}} - \widehat{\mathcal{L}}_D| \le \Delta^r$. Set $s = \Delta^e + \Delta^r + \Delta^a$ and $\hat{\varepsilon} = \widehat{\mathcal{L}} + s$. With $\delta_e + \delta_a \le \delta$, the declared population leakage satisfies, with probability at least $1 - \delta$,*

$$\max\{0, \hat{\varepsilon} - 2s\} \le \mathcal{L} \le \min\{1, \hat{\varepsilon}\}. \tag{57}$$

*Exact finite-dictionary auditing uses $\Delta^r = \Delta^a = 0$ and $\delta_e = \delta$.*

*Proof.* On the intersection of the sampling and approximation events,

$$\begin{aligned} |\mathcal{L} - \widehat{\mathcal{L}}| &\le |\mathcal{L} - \mathcal{L}_D| + |\mathcal{L}_D - \widehat{\mathcal{L}}_D| + |\widehat{\mathcal{L}}_D - \widehat{\mathcal{L}}| \\ &\le \Delta^{\mathrm{a}} + \Delta^{\mathrm{e}} + \Delta^{\mathrm{r}} = s. \end{aligned} \tag{58}$$

The union bound assigns this event probability at least $1 - \delta_e - \delta_a$; independence between the two events is not required. The absolute-value bound gives two separate inequalities,

$$\mathcal{L} \le \widehat{\mathcal{L}} + s = \hat{\varepsilon}, \qquad \mathcal{L} \ge \widehat{\mathcal{L}} - s = \hat{\varepsilon} - 2s. \tag{59}$$

Intersecting this interval with the deterministic range $[0, 1]$ proves both endpoints. To handle a random design, condition first on all pre-audit information, including the trained encoders and deployed edit. The sampling bound holds under this conditioning because the audit records are independent of that information; averaging its conditional failure probability leaves a failure probability at most $\delta_e$. The deployed edit, feature map, decoder radius and target must be fixed before the audit sample is drawn; selecting an edit using this sample requires a new audit or a bound uniform over the selection. In particular, this theorem does not replace the analytic slack by a bootstrap quantile or cover an attack class selected after looking at audit outcomes. □

**Audited state and solver excess.** The theorem applies to the declared decoder class acting on $X_A$, which collects the observable channels of the committed release. Reducing this interface to $AH$ requires same-class factorization: every admissible full-state decoder has an equivalent decoder of $AH$. Otherwise the read-out and surviving caches enter $X_A$ explicitly. This states the coverage condition independently of the purge operation. For a block-feasible edit in the same unrestricted class at the same $\tau$,

$$0 \le \mathcal{L}(A) - \varepsilon^{\mathrm{B}}(\tau) \le \mathcal{L}(A) - \varepsilon^{\mathrm{C}}(\tau). \tag{60}$$

The total excess decomposes into the minimum locality cost and the solver's residual:

$$\mathcal{L}(A) - \varepsilon^{\mathrm{C}}(\tau) = S(\tau) + g_A, \qquad g_A = \mathcal{L}(A) - \varepsilon^{\mathrm{B}}(\tau) \ge 0. \tag{61}$$

Eq. (27) evaluates the exact-utility benchmark for correlated blocks. At relaxed utility, feasible candidate values $\tilde{\varepsilon}^{\mathrm{B}} = \varepsilon^{\mathrm{B}} + g_B$ and $\tilde{\varepsilon}^{\mathrm{C}} = \varepsilon^{\mathrm{C}} + g_C$ give $\tilde{\varepsilon}^{\mathrm{B}} - \tilde{\varepsilon}^{\mathrm{C}} = S + g_B - g_C$. Their difference is a candidate comparison; an interval for the optimal surcharge requires bounds on both optimization residuals. Empirical covariance versions evaluate the corresponding sample benchmark.