FedCCA: Federated Canonical Correlation Analysis

Zhengquan Luo[1] **Kai Fong Ernest Chong**[2] **Pengfei Wei**[3]
Changyou Chen[4] Peilin Zhao[5] Renmin Han[6]
Chunlai Zhou[7] Yunlong Wang[8] Zhiqiang Xu[1,*]
[1] Mohamed bin Zayed University of Artificial Intelligence
[2] Singapore University of Technology and Design
[3] MTRI Tokyo Technology Research Institute
[4] State University of New York at Buffalo
[5] Shanghai Jiao Tong University [6] Shandong University
[7] Renmin University of China [8] Chinese Academy of Sciences
[*]Corresponding author: zhiqiang.xu@mbzuai.ac.ae

Abstract

Canonical Correlation Analysis (CCA) is a key tool for cross-modal learning, but centralized solutions are impractical due to the heavy cost of high-dimensional covariance operations and the privacy sensitivity of distributed data. To address these challenges, we propose **FedCCA**, a federated framework that replaces explicit inverses and inner least-squares solves with a truncated von Neumann series, reducing matrix inversions to lightweight matrix–vector multiplications while retaining provable convergence. This series formulation not only improves efficiency, but also provides explicit and tunable control of truncation error, and its structure naturally splits into client-side multiplications and a server-side projection step, making it particularly suitable for federated deployment. Building on this foundation, we incorporate Gaussian differential privacy and derive practical upper and lower bounds on the required noise variance, which yield end-to-end (ε, δ) guarantees together with convergence stability. Empirical results on five datasets confirm that FedCCA achieves accuracy comparable to centralized CCA and consistently outperforms ALS/TALS baselines in both sub-optimality gap and convergence speed, all while maintaining rigorous privacy protection.

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

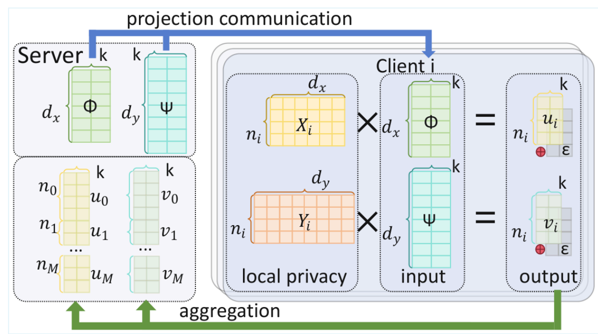

Figure 1: Federated CCA workflow. The server broadcasts current projections; clients return low-dimensional sketches. Optional Gaussian noise is added client-side for differential privacy.

1 INTRODUCTION

Canonical Correlation Analysis (CCA) is a fundamental technique in multi-modal learning, widely used to extract shared representations across different data modalities (Sanghavi and Verma, 2022; Karampatziakis and Mineiro, 2014). Despite its effectiveness, traditional CCA assumes centralized access to all raw data, leading to two critical challenges: (1) the prohibitive computational cost of forming and decomposing large covariance matrices, particularly in high-dimensional vision tasks, and (2) the privacy risks associated with sharing raw multi-modal data across different devices.

To formalize the CCA problem, we begin with two data matrices $X \in \mathbb{R}^{d_x \times n}$ and $Y \in \mathbb{R}^{d_y \times n}$ representing different modalities (e.g. visual features and text embeddings). The goal is to determine projection matrices $\Phi \in \mathbb{R}^{d_x \times k}$ and $\Psi \in \mathbb{R}^{d_y \times k}$ that maximize the correlation between the projected representations $X^\top \Phi$ and $Y^\top \Psi$. The empirical cross-covariance and auto-covariances are

$$C_{xy} := \frac{1}{n} XY^\top, C_{xx} := \frac{1}{n} XX^\top + r_x I, C_{yy} := \frac{1}{n} YY^\top + r_y I, \tag{1}$$

where r_x and r_y are small regularizers to ensure numerical stability. The block (or k-)CCA optimization problem is:

$$\begin{aligned} \text{maximize} \quad & \operatorname{tr}(\Phi^\top C_{xy} \Psi) \\ \text{subject to} \quad & \Phi^\top C_{xx} \Phi = \Psi^\top C_{yy} \Psi = I; \end{aligned} \tag{2}$$

whose classical solution can be obtained via singular value decomposition (SVD) on the whitened cross-covariance $C := C_{xx}^{-1/2} C_{xy} C_{yy}^{-1/2}$. However, for high-dimensional data, directly whitening the covariance matrices and performing SVD is computationally prohibitive. To mitigate this, various iterative methods based on stochastic gradient descent (SGD) (Bottou, 2010), stochastic variance reduced gradient (SVRG) (Johnson and Zhang, 2013), and alternating least squares (ALS) (Ge et al., 2016; Xu and Li, 2019) have been proposed. Improvements such as Faster ALS (FALS) (Xu and Li, 2019) and its simplified variant (Xu and Li, 2021) improve convergence rates but remain computationally expensive.

Advanced acceleration techniques such as shift-and-invert (Wang et al., 2016), iterative least squares (Lu and Foster, 2014), and streaming Oja methods (Bhatia et al., 2018) further reduce iteration complexity in the centralized setting, but they rely on global operators or full decompositions that do not translate into low-dimensional, per-client communication in horizontal federated learning (HFL). This gap motivates us to propose a design that is explicitly communication-aligned and privacy-aware.

To address these issues, we introduce **AMVM-CCA**, an alternating matrix–vector multiplication scheme that replaces explicit inverses and least-squares solves with a truncated von Neumann series. This transformation removes the need for gradient-based solvers and explicit matrix inversions, reducing heavy linear-algebra operations to lightweight matrix–vector products while preserving convergence guarantees. Importantly, the series order m provides direct geometric control of the residual error, turning an implicit approximation into an explicit and tunable accuracy–efficiency trade-off.

The AMVM formulation lends itself naturally to federated learning. In our design, each client only multiplies its local data with the current projection vectors, and the server aggregates these low-dimensional results and re-orthogonalizes the global projections (see Fig. 1). This protocol, which we call **FedCCA**, avoids any exchange of raw data and communicates only k-dimensional outputs, with $k \ll d_x, d_y, n$. As a result, the method is both communication-efficient and resistant to direct data reconstruction.

However, low-dimensional outputs alone do not provide formal privacy. While a single query produces an underdetermined system that cannot recover the client's raw data, a curious server could repeatedly issue carefully chosen queries to the same client and gradually infer sensitive information. To defend against this stronger threat, we extend FedCCA with client-side Gaussian noise injection, which achieves end-to-end (ε, δ)-differential privacy (DP) guarantees even under adaptive queries. We refer to this extension as **FedCCA-DP**. It allows practitioners to tune the series order m and the noise variance in a principled way, balancing convergence accuracy against the strength of privacy protection.

We validate FedCCA and FedCCA-DP on five datasets: Mediamill (Snoek et al., 2006), JW11 (Westbury et al., 1994), MNIST (LeCun et al., 1998), MFEAT (Dua et al., 2017), and Caltech101 (Fei-Fei et al., 2004). Results show that FedCCA achieves over three orders of magnitude lower sub-optimality gap and more than 20% faster convergence than ALS/FALS baselines, while FedCCA-DP maintains rigorous privacy guarantees.

In summary, our contributions are threefold:

- **AMVM-CCA**: *a new CCA paradigm based on alternating matrix-vector multiplications.* We use von Neumann series approximation to circumvent costly least-squares gradient computations in existing CCA methods, thereby improving computational efficiency and numerical stability, while concurrently allowing a fine-grained control of residual errors.

- **FedCCA**: *a new distributed, privacy-preserving CCA framework.* We seamlessly integrate our iterative CCA paradigm with federated learning, enabling large-scale multi-modal analysis across clients without data-sharing.

- **FedCCA-DP**: *an extension of FedCCA with differential privacy, coupled with theoretical guarantees on privacy-convergence tradeoffs.* We prove lower and upper bounds on noise variance, thereby providing a rigorous analysis of the privacy-convergence tradeoff.

2 RELATED WORK

CCA solvers. Since Hotelling's original formulation (Hotelling, 1935), many algorithms have been developed to scale CCA. Alternating least-squares (ALS/FALS) (Ge et al., 2016; Xu and Li, 2019, 2021) and stochastic methods (Arora et al., 2017) reduce per-iteration arithmetic but still require data-dependent inner least-squares or multi-step CG solves. Recent work has proposed stronger accelerations: Lazy-CCA (Allen-Zhu and Li, 2017) combines polynomial approximation with variance reduction for faster convergence, shift-and-invert preconditioning (Wang et al., 2016) converts the problem into solving regularized linear systems, ILS (Lu and Foster, 2014) iteratively approximates inverses via least-squares, and Gen-Oja (Bhatia et al., 2018) gives a streaming solution based on the generalized Oja's rule. These methods achieve provable speedups in centralized settings, but they require access to global covariance matrices or repeated high-dimensional SVD/linear system solves, which makes them impractical for horizontal FL where only local computations and low-dimensional communication are allowed.

Distributed and privacy-preserving CCA. In the context of distributed computation, DCCA (Bertrand, 2015) and subsequent variants rely on centralized high-dimensional SVD, incurring heavy computation and communication. More recent works (Fu et al., 2016; Chen and Zhu, 2024) propose distributed estimators to reduce cost, but these works focus on statistical estimation and do not provide end-to-end protocols under privacy constraints. For privacy, DP-based CCA (Imtiaz and Sarwate, 2017) and homomorphic-encryption-based methods (Wang, 2017) exist but introduce large utility or efficiency trade-offs. The distributed DP-CCA method of Imtiaz and Sarwate (2019) integrates Gaussian noise but still requires centralized SVD, leading to high overhead. Very recently, Shrestha and Fu (2023) proposed a federated CCA framework, but it is limited to the vertical setting where each client holds different views of the same samples; the more practical horizontal setting remains unaddressed. Here, horizontal FL refers to the scenario where each client holds samples of the same feature space (different rows), whereas vertical FL splits the feature space across parties for the same samples.

Our contribution. FedCCA is the first framework for horizontal federated CCA with rigorous privacy guarantees. By reformulating CCA through a truncated von Neumann series, we avoid explicit inversions and full SVD, reduce updates to lightweight local matrix–vector multiplications, and obtain explicit and tunable error control. This decomposition makes the algorithm naturally compatible with FL protocols, and by integrating Gaussian DP with tight variance bounds, we achieve end-to-end (ε, δ) guarantees. Compared to prior centralized accelerations, distributed approximations, or vertical FL approaches, FedCCA simultaneously addresses *scalability, privacy, and federated deployment*, which none of the existing methods achieve.

3 PRELIMINARIES

The goal of CCA is to solve equation (2), which involves whitening the cross-covariance matrix $C = C_{xx}^{-1/2} C_{xy} C_{yy}^{-1/2}$ and then performing a truncated SVD. Specifically, the top k singular vectors and values are given by $(U, \Sigma, V) = \text{rank-k SVD}(C)$, where $U \in \mathbb{R}^{d_x \times k}$ and $V \in \mathbb{R}^{d_y \times k}$ represent the top k left and right singular vectors of C, and Σ contains the top k singular values. The final projection matrices are: $(\Phi, \Psi) := (C_{xx}^{-1/2} U, C_{yy}^{-1/2} V)$.

The maximum correlation between the projected variables is given by $\mathrm{tr}(\Sigma)$, the sum of the top k singular values. In practice, directly computing covariance matrices and performing k-SVD on C can be computationally expensive, especially in high-dimensional settings.

Connection to alternating regressions. Under Tikhonov regularization and whitening, the block CCA objective is *approximately recast* as alternating ridge regressions that minimize $\|X^\top \Phi - Y^\top \Psi\|_{\mathrm{F}}^2$ subject to the canonical constraints $\Phi^\top C_{xx} \Phi = I, \Psi^\top C_{yy} \Psi = I$. This equivalence holds up to the regularization terms, which vanish as $r_x, r_y \to 0$; see Hardoon et al. (Hardoon et al., 2004) and Witten et al. (Witten et al., 2009) for precise conditions. Hence, equation (2) can be *approximately recast* as the following ridge-regularized alternating least-squares objective:

$$\begin{cases} \min_{\Phi \in \mathbb{R}^{d_x \times k}} \ell_\Psi(\Phi) := \frac{1}{2n} \|\mathbf{X}^\top \Phi - \mathbf{Y}^\top \Psi_t\|_F^2 + \frac{r_x}{2} \|\Phi\|_F^2; \\ \min_{\Psi \in \mathbb{R}^{d_y \times k}} \ell_\Phi(\Psi) := \frac{1}{2n} \|\mathbf{X}^\top \Phi_t - \mathbf{Y}^\top \Psi\|_F^2 + \frac{r_y}{2} \|\Psi\|_F^2. \end{cases} \tag{3}$$

We then iteratively compute a sequence $\{(\Phi_t, \Psi_t)\}_{t=1}^T$ converging to an approximate solution of (3), via the following updates:

$$\begin{cases} \tilde{\Phi}_{t+1} \leftarrow C_{xx}^{-1} C_{xy} \Psi_t + \xi_t; \\ \Phi_{t+1} \leftarrow \tilde{\Phi}_{t+1} \left(\tilde{\Phi}_{t+1}^\top C_{xx} \tilde{\Phi}_{t+1} \right)^{-\frac{1}{2}}; \\ \tilde{\Psi}_{t+1} \leftarrow C_{yy}^{-1} C_{xy}^\top \Phi_{t+1} + \eta_t; \\ \Psi_{t+1} \leftarrow \tilde{\Psi}_{t+1} \left(\tilde{\Psi}_{t+1}^\top C_{yy} \tilde{\Psi}_{t+1} \right)^{-\frac{1}{2}}. \end{cases} \tag{4}$$

Here each $\tilde{\Phi}_{t+1}$ (resp. $\tilde{\Psi}_{t+1}$) is an intermediate ridge regression solution with residual ξ_t (resp. η_t), which is then normalized to satisfy the canonical constraints. Although this reduces cost relative to direct SVD, the residual terms ξ_t, η_t can accumulate, leading to approximation error.

4 METHOD

4.1 Alternating matrix–vector multiplication

Gradient-based inner solvers (e.g., Nesterov's acceleration) introduce residuals ξ_t, η_t because their iterates deviate from the exact products $C_{xx}^{-1} C_{xy} \Psi_t$ and $C_{yy}^{-1} C_{xy}^\top \Phi_t$. These residuals are hard to quantify and can hurt robustness: small steps slow convergence, whereas large steps can cause oscillations or divergence (Nesterov, 1983; Montavon et al., 2012). To avoid *explicit inverses and inner least-squares/CG solves*, we introduce an alternating matrix–vector multiplication (AMVM) scheme based on the *von Neumann* series, which yields closed-form, quantifiable truncation error.

von Neumann series realization and shape consistency. To avoid explicit matrix inversions in the regularized CCA problem, we rewrite the inverses of the auto-covariance matrices using a von Neumann series. For any square matrix A with spectral radius $\rho(A) < 1$, it holds that

$$(I + A)^{-1} = \sum_{k=0}^{\infty} (-A)^k.$$

We apply this expansion to the regularized auto-covariances. To ensure scale-invariance and numerical stability across clients in the federated setting, we first normalize the input matrices:

$$\bar{X} := \frac{X}{N_{xy}}, \qquad \bar{Y} := \frac{Y}{N_{xy}},$$

where the normalization factor $N_{xy} > 0$ is computed once before training using secure aggregation (see Section 4.1).

The regularized auto-covariance matrices then take the form

$$\begin{aligned} \bar{C}_{xx} &= \frac{1}{n} \bar{X} \bar{X}^\top + r_x I = r_x \left(I + \frac{1}{n r_x} \bar{X} \bar{X}^\top \right), \\ \bar{C}_{yy} &= r_y \left(I + \frac{1}{n r_y} \bar{Y} \bar{Y}^\top \right), \end{aligned} \tag{5}$$

where $r_x, r_y > 0$ are small regularization constants that guarantee numerical stability.

Let

$$A := \frac{1}{n r_x} \bar{X} \bar{X}^\top \in \mathbb{R}^{d_x \times d_x}, \qquad B := \frac{1}{n r_y} \bar{Y} \bar{Y}^\top \in \mathbb{R}^{d_y \times d_y}.$$

Applying a truncated von Neumann series, the AMVM updates can be written as

$$\begin{aligned} (I + A)^{-1} C_{xy} \Psi_t &\approx \sum_{k=0}^{m} (-A)^k C_{xy} \Psi_t, \\ (I + B)^{-1} C_{yx} \Phi_t &\approx \sum_{k=0}^{m} (-B)^k C_{yx} \Phi_t. \end{aligned} \tag{6}$$

This formulation ensures shape consistency: the left-hand products $(\sum_{k=0}^{m} (-A)^k) C_{xy} \Psi_t \in \mathbb{R}^{d_x \times k}$ and symmetrically $(\sum_{k=0}^{m} (-B)^k) C_{yx} \Phi_t \in \mathbb{R}^{d_y \times k}$, matching the dimensions required by the AMVM updates. Moreover, this representation reduces matrix inversion to repeated matrix–vector multiplications, which can be computed efficiently and independently on each client in the federated setting.

Scale-invariance under joint normalization. For *regularized* CCA, scaling both views by $s > 0$ while rescaling the regularizers $(X, Y, r_x, r_y) \mapsto (sX, sY, s^2 r_x, s^2 r_y)$ preserves the solution. Our normalized form factors out r_x, r_y, so applying the same normalization to both views does not alter the regularized solution (Thompson, 1984; Hardoon et al., 2004).

A strict spectral-radius condition via operator norm. Choose N_{xy} so that

$$\|\bar{X}\|_2^2 \leq \alpha \, n r_x, \qquad \|\bar{Y}\|_2^2 \leq \alpha \, n r_y \quad (\alpha \in (0, 1)),$$

which implies $\|A\|_2 \leq \alpha$ and $\|B\|_2 \leq \alpha$. A one-shot choice satisfying these is

$$N_{xy} := \frac{\max\{\max_j \|x_j\|_2, \ \max_j \|y_j\|_2\}}{\sqrt{\alpha} \ \min\{\sqrt{r_x}, \sqrt{r_y}\}},$$

since $\|\bar{X}\|_2^2 \leq \sum_{j=1}^{n} \|\bar{x}_j\|_2^2 \leq n \max_j \|\bar{x}_j\|_2^2 \leq \alpha n r_x$ (and similarly for $\bar{Y}$). Using the series,

$$\begin{aligned} \bar{C}_{xx}^{-1} &= \tfrac{1}{r_x} (I + A)^{-1} = \tfrac{1}{r_x} \sum_{k=0}^{\infty} (-A)^k, \\ \bar{C}_{yy}^{-1} &= \tfrac{1}{r_y} (I + B)^{-1} = \tfrac{1}{r_y} \sum_{k=0}^{\infty} (-B)^k. \end{aligned} \tag{7}$$

Substituting (7) into (4) gives

$$\begin{cases} \tilde{\Phi}_{t+1} \leftarrow \frac{1}{r_x} \sum_{k=0}^{\infty} (-A)^k \bar{C}_{xy} \Psi_t, \\ \Phi_{t+1} \leftarrow \tilde{\Phi}_{t+1} \big(\tilde{\Phi}_{t+1}^\top \bar{C}_{xx} \tilde{\Phi}_{t+1} \big)^{-1/2}, \\ \tilde{\Psi}_{t+1} \leftarrow \frac{1}{r_y} \sum_{k=0}^{\infty} (-B)^k \bar{C}_{xy}^\top \Phi_{t+1}, \\ \Psi_{t+1} \leftarrow \tilde{\Psi}_{t+1} \big(\tilde{\Psi}_{t+1}^\top \bar{C}_{yy} \tilde{\Psi}_{t+1} \big)^{-1/2}. \end{cases} \tag{8}$$

Truncation and privacy composition. We shall truncate the series at order m and regard the tail as a residual:

$$\xi_t = -\tfrac{1}{r_x} \sum_{k=m+1}^{\infty} (-A)^k \bar{C}_{xy} \Psi_t,$$

$$\eta_t = -\tfrac{1}{r_y} \sum_{k=m+1}^{\infty} (-B)^k \bar{C}_{xy}^\top \Phi_{t+1}.$$

Because $\|A\|_2, \|B\|_2 \le \alpha < 1$, the truncation bias decays geometrically as $O(\alpha^{m+1}/(1-\alpha))$. When Gaussian DP noise is added per matrix–vector product, the *per-query* noise scale σ does not depend on m, but the *composed privacy loss* grows linearly with the number of noisy queries, which increases with m. Thus m tunes the accuracy–privacy trade-off without changing the convergence mechanism.

Principal-angle notation. Let

$$\begin{aligned} \theta'_T &:= \cos^{-1}\big(\sigma_k(U^\top C_{xx}\Phi_T)\big), \\ \theta''_T &:= \cos^{-1}\big(\sigma_k(V^\top C_{yy}\Psi_T)\big), \\ \theta_T &:= \max\{\theta'_T, \theta''_T\}. \end{aligned}$$

Theorem 1. *Let X, Y and $C_{xx}, C_{xy}, C_{yy}, \bar{C}_{xx}, \bar{C}_{yy}$ be as above. Suppose $\|A\|_2, \|B\|_2 \le \alpha < 1$. If the* von Neumann *order m satisfies*

$$m \ge \frac{\log\left(\frac{\min\{\sqrt{\lambda_{\max}(\bar{C}_{xx})},\ \sqrt{\lambda_{\max}(\bar{C}_{yy})}\}\ \max\{\sin\theta_T,\ \cos\theta_T\}}{\max\{\sqrt{r_y/r_x}\,\|\Phi_T\|_F,\ \sqrt{r_x/r_y}\,\|\Psi_T\|_F\}}\right)}{\log(1/\alpha)} + C,$$

$$C := \frac{\log\left(\frac{\sigma_k^2-\sigma_{k+1}^2}{12(1-\alpha)}\right)}{\log(1/\alpha)} - 2,$$

then $\tan(\theta_T) \le \varsigma$ can be reached in

$$T = O\left(\frac{\sigma_k^2}{\sigma_k^2 - \sigma_{k+1}^2}\ \log\frac{1}{\varsigma\cos\theta_0}\right),$$

where $\varsigma > 0$ is a target accuracy and θ_0 is the initial principal-angle quantity.

4.2 FedCCA

Building on the AMVM formulation in §4.1, we next consider scenarios where the two views of data are distributed across different institutions or devices—for example, multi-modal signals collected on mobile platforms or biomedical datasets held by separate hospitals. In such cases, centralizing raw samples is often infeasible due to regulatory and bandwidth constraints, making classical distributed linear-algebra solvers unsuitable since they typically assume unrestricted sharing of raw data or sufficient statistics across nodes. Federated learning (FL) offers a more realistic paradigm: clients retain local data and communicate only low-dimensional, privacy-preserving intermediate features to a central server, enabling joint CCA without violating privacy requirements.

The AMVM structure aligns naturally with this setting. Each von Neumann term decomposes into block-separable bilinear forms (e.g., $XX^\top W = \sum_i X_i X_i^\top W$), enabling clients to perform lightweight local multiplications while the server simply aggregates the results and re-orthogonalizes the projections.

Algorithm 1 FedCCA client-i:

Input: Operators Op, # local data $X_i \in \mathbb{R}^{d_x \times n_i}$ and $Y_i \in \mathbb{R}^{d_y \times n_i}$, # operation matrix W_{op}.
$Op = 1$: $W_1 = W_{op} \in \mathbb{R}^{n_i \times k}$, $Out = X \cdot W_1 \in \mathbb{R}^{d_x \times k}$.
$Op = 2$: $W_2 = W_{op} \in \mathbb{R}^{d_x \times k}$, $Out = X^\top \cdot W_2 \in \mathbb{R}^{n_i \times k}$.
$Op = 3$: $W_3 = W_{op} \in \mathbb{R}^{n_i \times k}$, $Out = Y \cdot W_3 \in \mathbb{R}^{d_y \times k}$.
$Op = 4$: $W_4 = W_{op} \in \mathbb{R}^{d_y \times k}$, $Out = Y^\top \cdot W_4 \in \mathbb{R}^{n_i \times k}$.
return Out

Compared to ALS or CG-based solvers that require repeated global reductions and data-dependent inner solves, this design minimizes synchronization cost and admits straightforward differential privacy calibration, making FedCCA a natural extension of AMVM to the federated regime.

FedCCA realizes the AMVM updates in equation (8) by orchestrating local sketches and server aggregation (Alg. 1–2). In each communication round, the server first broadcasts the current projection matrices. Each client then computes and uploads a set of low-dimensional sketches corresponding to its local data. Specifically, four client-side sketching operators are used to implement the components of $\bar{C}_{xy}\Psi_t$ and $A^j\bar{C}_{xy}\Psi_t$ (and symmetrically for Y):

- **Op 1:** $X_i W_1 \in \mathbb{R}^{d_x \times k}$, projection of the client's X features.
- **Op 2:** $X_i^\top W_2 \in \mathbb{R}^{n_i \times k}$, transposed sketch for aggregation.
- **Op 3:** $Y_i W_3 \in \mathbb{R}^{d_y \times k}$, projection of the client's Y features.
- **Op 4:** $Y_i^\top W_4 \in \mathbb{R}^{n_i \times k}$, transposed sketch for aggregation.

The server then aggregates these sketches, accumulates the von Neumann terms up to order m, and performs a $k \times k$ Gram orthogonalization to update Φ. The Ψ-update is carried out analogously with $X \leftrightarrow Y$. Because all transmitted variables are k-dimensional projections ($k \ll d_x, d_y, n_i$), this procedure preserves client privacy while enabling efficient federated computation.

Complexity. Each Φ-update on a client requires $(2{+}2m)$ local sketching operations (one $Op3 \to Op2$ and m repetitions of $Op1 \to Op2$). The resulting per-round communication can be separated as: server $\to$ clients: $O(M \cdot d_2 \cdot k)$, and clients $\to$ server: $O(M \cdot \max\{n_i, d_2\} \cdot k)$, with $d_2 = \max\{d_x, d_y\}$. Thus the total per-round communication is $O\big((m{+}1)\,M\,d_2\,k\big)$ messages per side due to the m Neumann terms. The actual transmitted bytes scale linearly with $(m{+}1)Md_2k$ (assuming single precision).

Algorithm 2 FedCCA Server for Updating Φ:

Input: Series order m, # client number M, # projection matrices Φ_t, Ψ_t,# regularization parameters r_x, r_y

Federated least-square solver of ϕ (In: ψ_t, Out:$\tilde{\Phi}_{t+1}$)

Initialize $\mathbf{P}_0 = \mathbf{0} \in \mathbb{R}^{d_x \times k}$, where $n = \sum_{i=1}^{M} n_i$.

for $i = 1$ **to** M **do**

  Receive $\text{Out}_i = \text{client-}i(Op = 4,\ W_{op} = \Psi_t)$;

  Receive $\text{Out}_i = \text{client-}i(Op = 1,\ W_{op} = \text{Out}_i)$

  Update $\mathbf{P}_0 = \mathbf{P}_0 + \text{Out}_i$

Compute $\mathbf{P}_0 = \frac{\mathbf{P}_0}{n \cdot r_x}$. Initialize $\tilde{\Phi}_{t+1} = \mathbf{P}_0$.

for $j = 1$ **to** m **do**

  Initialize $\mathbf{P}_j = \mathbf{0} \in \mathbb{R}^{d_x \times k}$.

  for $i = 1$ **to** M **do**

    Receive $\text{Out}_i = \text{client-}i(Op = 2,\ W_{op} = \mathbf{P}_{j-1})$

    Receive $\text{Out}_i = \text{client-}i(Op = 1,\ W_{op} = \text{Out}_i)$

    Update $\mathbf{P}_j = \mathbf{P}_j + \text{Out}_i$

  $\mathbf{P}_j = \frac{\mathbf{P}_j}{n \cdot r_x}$. $\tilde{\Phi}_{t+1} = \tilde{\Phi}_{t+1} + (-1)^j \mathbf{P}_j$.

Return $\tilde{\Phi}_{t+1}$.

Federated orthogonalization of ϕ

Initialize $\text{Norm} = \mathbf{0} \in \mathbb{R}^{k \times k}$.

for $i = 1$ **to** M **do**

  Receive $\text{Out}_i = \text{client-}i(Op = 2,\ W_{op} = \tilde{\Phi}_{t+1})$

  Receive $\text{Out}_i = \text{client-}i(Op = 1,\ W_{op} = \text{Out}_i)$

  Update $\text{Norm} = \text{Norm} + \tilde{\Phi}_{t+1}^{\top} \cdot \text{Out}_i$

Compute $\text{Norm} = \frac{1}{n} \cdot \text{Norm}$. Compute $\Phi_{t+1} = \tilde{\Phi}_{t+1} \cdot \text{Norm}^{-1/2}$. **Return** Φ_{t+1}.

In contrast, ALS/TALS solvers with ridge regularization require solving $(d_x \times d_x)$ or $(d_y \times d_y)$ systems or running multiple conjugate-gradient steps, leading to per-iteration cost at least $\Omega(d_x^{\omega} k)$ or $\Omega(\text{nnz}(X) \cdot \#\text{CG})$, together with repeated global synchronizations. By comparison, AMVM requires only $(m+1)$ forward multiplications with $XX^{\top}$ and $YY^{\top}$ per side plus a server-side $O(k^3)$ orthogonalization (via an EVD or QR of the $k \times k$ Gram), yielding lower and more predictable per-round cost that aligns naturally with federated settings.

Privacy considerations. In FedCCA, clients never transmit raw samples. Instead, they send only low-dimensional linear sketches of the form $X_i^{\top} W$ or $Y_i^{\top} W$, where the multiplier W is public. The sensitivity of such sketches is determined solely by the norm of W, which makes it straightforward to calibrate Gaussian noise for differential privacy. Moreover, the number of sketches sent in each round is fixed by the chosen von Neumann series order m and does not depend on data conditioning. By contrast, LS or CG solvers require a data-dependent number of iterations and reveal intermediate residuals whose sensitivity is harder to control. FedCCA therefore yields a clean tradeoff: increasing m reduces approximation bias geometrically, while the privacy cost grows only linearly with m, enabling predictable accuracy–privacy accounting across rounds.

4.3 FedCCA with DP guarantee

The privacy of FedCCA is partially protected during communication because the feature dimension k is deliberately chosen to be much smaller than both the data dimension d and the sample size n_i, making the system underdetermined: a single transmission provides only $n_i \times k$ equations versus $d \times n_i$ unknowns, and is therefore insufficient to reconstruct the original data X_i or Y_i. Nevertheless, if a malicious server repeatedly queries the same client with sufficiently many distinct multipliers (at least $t \geq \lceil d/k \rceil$) and the stacked query matrix $\bar{A}_{\text{attack}} \in \mathbb{R}^{(tk) \times d}$ attains full rank, then full reconstruction becomes theoretically possible. Even without meeting these conditions, carefully designed queries may still expose partial information—for instance, choosing $A_i = [1, 0, 0, 0]^{\top}$ directly reveals x_1. Hence, FedCCA's native underdetermination substantially hinders, but does not entirely eliminate, the risk of reconstruction attacks, motivating the integration of formal differential privacy mechanisms.

To rigorously mitigate such risks, we enforce *record-level differential privacy (DP)*. We adopt record-level adjacency, where two datasets differ in exactly one row of X_i or Y_i. Importantly, the server may adaptively choose the public multipliers W_{op} based on past messages, and the DP guarantee continues to hold under such adaptivity by standard composition. Each Φ-update on a client involves $(2+2m)$ primitive operator calls (and symmetrically for Ψ), so across T outer rounds with M clients, the total number of private queries is

$$Q = 4(2+2m)TM.$$

We instantiate DP by applying the Gaussian mechanism to each of the four client-side sketches (Ops 1–4). Let Δ_j denote the ℓ_2-sensitivity of the j-th sketch, as determined by $\|W_{\text{op}}\|$ and the normalization. Using Rényi DP (or the moments accountant), the per-query RDP at order $q > 1$ is

$$\rho_j(q) = \frac{q\,\Delta_j^2}{2\sigma_j^2},$$

when Gaussian noise $\mathcal{N}(0, \sigma_j^2)$ is added to the j-th sketch. By additivity, the total RDP is

$$\rho_{\text{tot}}(q) = \sum_{\ell=1}^{Q} \rho_{j_\ell}(q),$$

and converting RDP to (ε, δ)-DP yields

$$\varepsilon(\delta) = \min_{q>1} \left\{ \rho_{\text{tot}}(q) + \frac{\log(1/\delta)}{q-1} \right\}.$$

A formal statement is given in Appendix F.

Theorem 2 (Noise lower bound for DP). *In the T-th iteration of the bi-level optimization, FedCCA achieves (ε, δ)-differential privacy if each client adds Gaussian noise to the four local multiplications with variances satisfying*

$$\sigma_1 \geq \frac{2\sqrt{2}\alpha\,\|\Psi_t\|_2}{\varepsilon}\sqrt{\frac{r_y}{nr_x}\,\log\Big(\frac{1.25}{\delta}\Big)}, \quad \sigma_2 \geq \frac{2\sqrt{2}\alpha\,\|\Psi_t\|_2}{\varepsilon}\sqrt{\alpha r_y\,\log\Big(\frac{1.25}{\delta}\Big)},$$
$$\sigma_3 \geq \frac{2\sqrt{2}\alpha\,\|\Phi_{t+1}\|_2}{\varepsilon}\sqrt{\frac{r_x}{nr_y}\,\log\Big(\frac{1.25}{\delta}\Big)}, \quad \sigma_4 \geq \frac{2\sqrt{2}\alpha\,\|\Phi_{t+1}\|_2}{\varepsilon}\sqrt{\alpha r_x\,\log\Big(\frac{1.25}{\delta}\Big)}.$$

Theorem 2 establishes the necessary noise variance for achieving DP. However, excessively large noise may hinder convergence. To ensure both privacy and optimization guarantees, we next derive an upper bound on the admissible noise variance.

Theorem 3 (Noise upper bound for convergence). *Assuming a common variance σ for the four Gaussian noise terms (cf. Theorem 2), FedCCA converges with probability at least $1-\mu$ (cf. Theorem 1) if σ satisfies*

$$\sigma \leq \frac{C_1 \cdot \min(\sin\theta_T, \cos\theta_T) - C_2}{\Phi^{-1}\big(1-\frac{\mu}{2}\big)\,C_3},$$

where
$$C_1 = \frac{\sigma_k^2 - \sigma_{k+1}^2}{12}\min\left\{\lambda_{\min}^{1/2}(C_{xx}), \lambda_{\min}^{1/2}(C_{yy})\right\},$$
$$C_2 = \frac{\alpha^{m+2}}{1-\alpha}\max\left\{\sqrt{\tfrac{r_y}{r_x}}\|\Psi_T\|, \sqrt{\tfrac{r_x}{r_y}}\|\Phi_T\|\right\},$$
$$C_3 = \frac{m+1}{1-\alpha}\max\left\{\tfrac{1+\sqrt{n\alpha r_x}}{nr_x}, \tfrac{1+\sqrt{n\alpha r_y}}{nr_y}\right\}.$$

Here $\Phi^{-1}(\cdot)$ denotes the upper quantile of the standard normal cumulative distribution function.

The two bounds together reveal an explicit trade-off between privacy and convergence. Holding the per-op noise level σ fixed, the end-to-end privacy loss $\varepsilon(\delta)$ grows linearly with the total number of queries $Q = 4(2{+}2m)TM$ (since RDP is additive). Hence, increasing the series order m directly increases the privacy cost. Conversely, to maintain a target (ε, δ) while Q grows, the required per-op noise must scale as $\sigma = \Theta(\sqrt{Q})$ (up to constants depending on α). Larger m reduces the AMVM truncation bias but either increases privacy loss or requires larger σ, which tightens the convergence upper bound in Theorem 3.

5 EXPERIMENTS

Datasets and evaluation. We compare FedCCA (with and without DP noise) against existing distributed solvers, including ALS (Ge et al., 2016) and the truly alternating least squares (TALS) (Xu and Li, 2019). Five real-world datasets are used: Mediamill (Snoek et al., 2006), JW11 (Westbury et al., 1994), MNIST (LeCun et al., 1998), MFEAT (Dua et al., 2017), and Caltech101 (Fei-Fei et al., 2004). Statistics are summarized in Tab. 2 (Appendix). These benchmarks are widely used in the CCA literature and cover diverse modalities. We report relative sub-optimality as (Xu and Li, 2021) and adopt the metric

$$\frac{f^* - f}{f^*} \;=\; \frac{\mathrm{tr}(\Sigma) - \mathrm{tr}(\Phi_t^\top C_{xy}\Psi_t)}{\mathrm{tr}(\Sigma)},$$

where f^* is the optimal CCA objective, $\mathrm{tr}(\Sigma)$ is the sum of the top-k canonical correlations, and $\mathrm{tr}(\Phi_t^\top C_{xy}\Psi_t)$ is the value attained at iteration t. This metric represents the *relative gap to the top-k canonical correlations*: it monotonically decreases as the estimated subspaces align with the true ones, and smaller values indicate more accurate solutions.

Scope and Baselines. We restrict comparisons to solvers that are directly feasible in horizontal FL with DP. Centralized accelerations such as LazyCCA, shift-and-invert, ILS, Gen-Oja, or global SVD either require direct access to global covariances or incur prohibitive communication, and are thus not applicable in our setting. Accordingly, we focus on ALS and TALS as strong distributed baselines. Efficiency is reported in standardized units: GFLOPs (C1, computation) and MB (C2, communication), both derived from our complexity analysis in Section 4.2. Reported MB assumes single-precision storage, and the values are directly consistent with the big-O formulas.

Comparison. Fig. 2 compares ALS-k, TALS, AMVM-CCA (ALS-k variant[1]), AMVM-CCA (TALS variant), and FedCCA across the five datasets. AMVM-CCA (TALS) shows clear advantages: up to several orders-of-magnitude lower gap on MNIST and $> 20\%$ fewer passes on average across datasets with notable runtime savings. FedCCA achieves almost identical curves to AMVM-CCA (TALS), confirming that federated deployment preserves both accuracy and efficiency.

Gaussian noise. Fig. 3 illustrates the effect of Gaussian noise. Noise injection slightly reduces accuracy but provides rigorous privacy. Importantly, FedCCA-DP ($\sigma \neq 0$) on JW11, MNIST, and MFEAT still surpasses ALS and TALS in accuracy, highlighting robustness to noise. As expected, larger series order m means more noisy queries are composed, leading to gradual accuracy degradation. Thus, moderate m yields the best balance between DP protection and accuracy.

Efficiency. Table 1 compares FedCCA against DPCCA Imtiaz and Sarwate (2019) and DIST Chen

[1]Apart from replacing the least-squares step with the von Neumann series AMVM update, all other settings remain identical to ALS-k. See (Ge et al., 2016) for details.

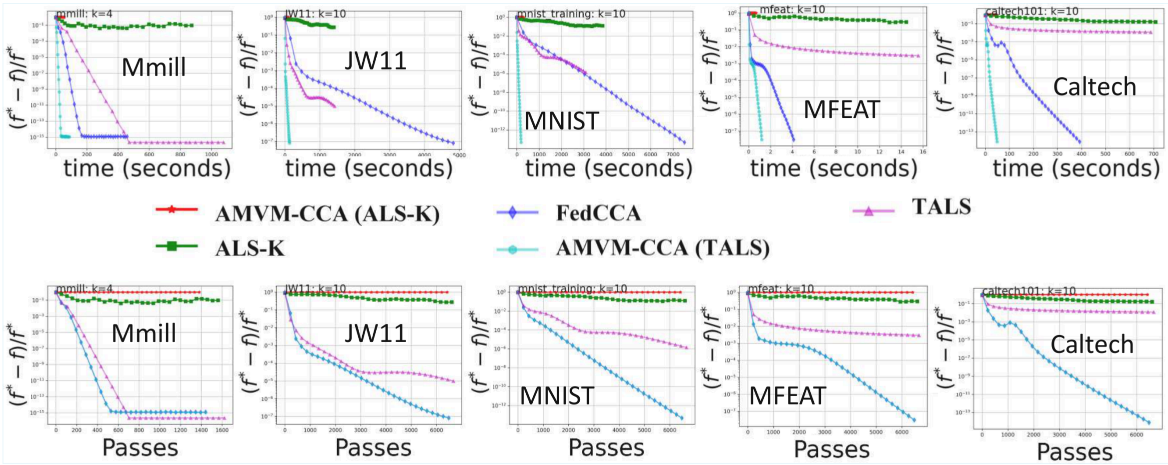

Figure 2: Comparison of ALS methods: AMVM-CCA improves accuracy on JW11, MNIST, MFEAT, and Caltech101 and significantly reduces passes and computation time. FedCCA maintains AMVM-CCA performance without loss. AMVM-CCA denotes the centralized algorithmic idea (alternating matrix–vector multiplications), while FedCCA is its federated realization with identical math but communication-aware orchestration.

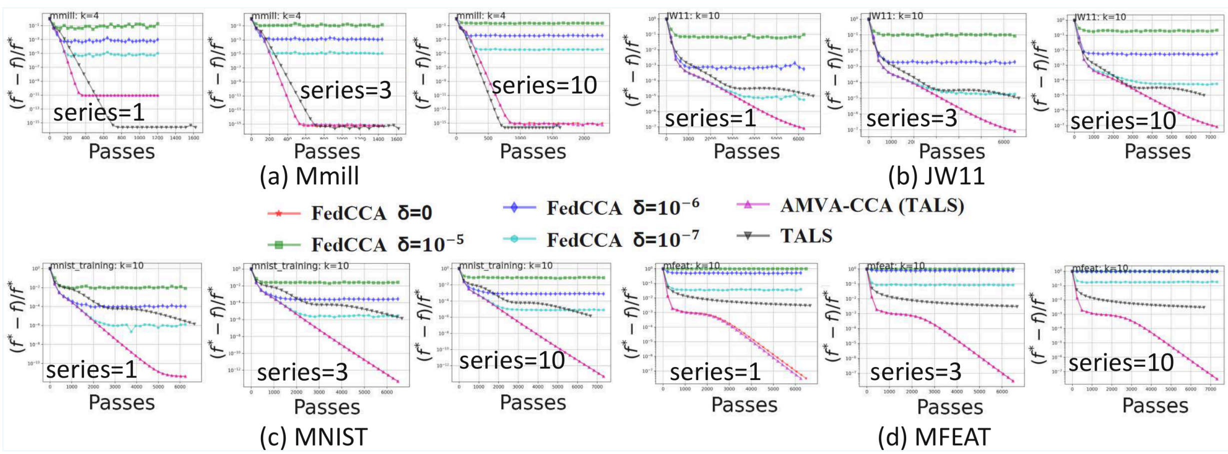

Figure 3: Impact of Gaussian noise on privacy and accuracy: While noise slightly degrades performance, accuracy with some noise on JW11, MNIST, and MFEAT still outperforms ALS. Higher series order adds more noise, further reducing accuracy.

and Zhu (2024). When $k \ll d$ and $k \ll TT'$, FedCCA achieves significantly lower communication and computation than both non-gradient distributed methods Bertrand (2015); Fu et al. (2016) and least-squares-based alternatives Chen and Zhu (2024). Results are reported in GFLOPs (C1) and MB (C2); the values confirm our derived theoretical complexity $O((n + Md_2)k)$.

Number of clients. Fig. 4(a) studies the effect of client count. As M increases, data per client decreases, reducing per-client cost. However, server aggregation overhead grows, so total time first rises and then drops, yielding an optimal range of M. Our experiments assume balanced i.i.d. splits; handling non-i.i.d. heterogeneity and much larger M remains future work.

von Neumann series order. Fig. 4(b) shows that increasing m improves accuracy but with diminishing returns and linearly higher cost. This confirms that moderate m values are most effective in practice.

Normalization. Fig. 4(c) analyzes the normalization hyperparameter α. Smaller α values accelerate convergence by smoothing gradients and reducing outlier sensitivity, but excessively small α risks overfitting. Hence, careful tuning of α is essential for balancing convergence speed and generalization.

Table 1: Computation (C1 GFLOPs) and communication (C2 MB) efficiency. Samples: n and data dimensions: d_x, d_y, where $d_1 = d_x + d_y$,$d_2 = max(d_x, d_y)$. Top-k = 10, client number M = 10, out-loop T = 40, and inner-loop $T' = 20$.

Methods	Communication complexity	MNIST		JW11	
		C1	C2	C1	C2
DPCCA-G (Imtiaz and Sarwate, 2019)	$O(Md_1^2)$	28.00	9.28	3.64	9.57
DIST (Chen and Zhu, 2024)	$O(TT'Md_1)$	1.86	6.27	2.25	3.08
FedCCA	$O((n + Md_2)k)$	**0.15**	**5.11**	**0.15**	**2.55**

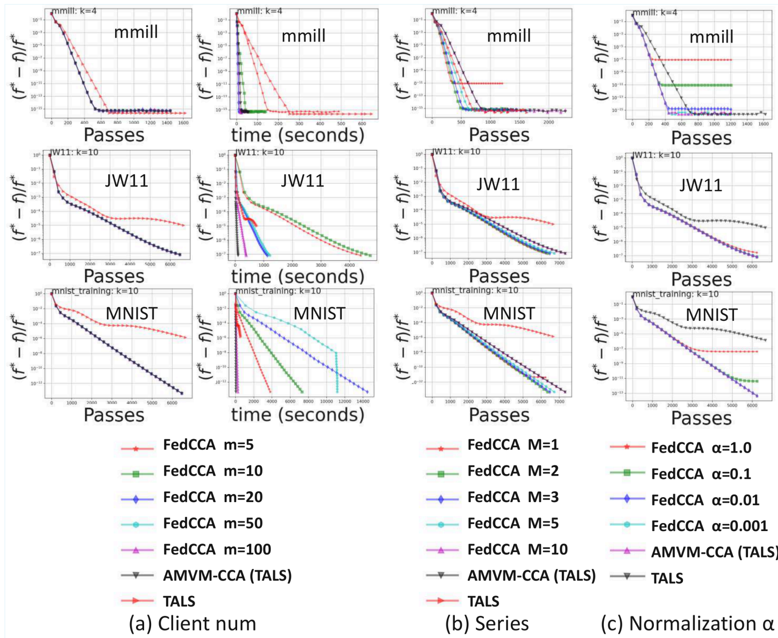

Figure 4: (a) Performance of FedCCA across different client numbers: As client count increases, operation time initially rises due to higher server workload, then falls as data per client decreases, reducing complexity per client. (b) Effect of von Neumann series order: Higher series order enhances accuracy with diminishing returns, while increasing the series order leads to linear growth in operation counts, underscoring the need for practical balance. (c) Impact of normalization hyperparameter α: Lower α values significantly improve performance.

6 CONCLUSION

We presented FedCCA, a federated CCA framework that replaces least-squares solves with von Neumann series–based alternating matrix–vector multiplications, yielding efficient computation with provable convergence. The federated design achieves privacy by sharing only low-dimensional sketches, and we further established rigorous differential privacy guarantees with tight variance bounds. Experiments show that FedCCA *outperforms least-squares baselines in our tested settings*, while maintaining privacy protection. Future work includes improving convergence rates and extending evaluation to larger, heterogeneous federated systems.

Limitations. FedCCA faces several limitations. First, the current theoretical and experimental analysis still requires more detailed treatment of client-side DP accounting and system-level implementation issues (e.g., communication cost, stopping criteria), which we plan to expand in follow-up work. Second, our experiments are restricted to balanced i.i.d. partitions with moderate client numbers; extending evaluation to non-i.i.d. heterogeneity and much larger scales is an important next step, orthogonal to the AMVM mechanics.

ACKNOWLEDGEMENTS

This work is supported in part by the Ministry of Education, Singapore, under its Tier 2 Research Fund (MOE-T2EP20125-0011), and in part by the SUTD Kickstarter Initiative (SKI 2021_03_01).

References

Allen-Zhu, Z. and Li, Y. (2017). Doubly accelerated methods for faster cca and generalized eigendecomposition. In *International Conference on Machine Learning*, pages 98–106. PMLR.

Arora, R., Marinov, T. V., Mianjy, P., and Srebro, N. (2017). Stochastic approximation for canonical correlation analysis. *Advances in neural information processing systems*, 30.

Bertrand, A. e. a. (2015). Distributed canonical correlation analysis in wireless sensor networks with application to distributed blind source separation. *IEEE TSP*, pages 4800–4813.

Bhatia, K., Pacchiano, A., Flammarion, N., Bartlett, P. L., and Jordan, M. I. (2018). Gen-oja: Simple & efficient algorithm for streaming generalized eigenvector computation. *Advances in neural information processing systems*, 31.

Bottou, L. (2010). Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010: 19th International Conference on Computational StatisticsParis France, August 22-27, 2010 Keynote, Invited and Contributed Papers*, pages 177–186. Springer.

Chen, C. and Zhu, L. (2024). Distributed estimation and gap-free analysis of canonical correlations. *arXiv:2412.17792*.

Dua, D., Graff, C., et al. (2017). Uci machine learning repository, 2017. *URL http://archive. ics. uci. edu/ml*, 7(1):62.

Fei-Fei, L., Fergus, R., and Perona, P. (2004). Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE.

Fu, X., Huang, K., Papalexakis, E. E., Song, H.-A., Talukdar, P. P., Sidiropoulos, N. D., Faloutsos, C., and Mitchell, T. (2016). Efficient and distributed algorithms for large-scale generalized canonical correlations analysis. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 871–876. IEEE.

Ge, R., Jin, C., Netrapalli, P., Sidford, A., et al. (2016). Efficient algorithms for large-scale generalized eigenvector computation and canonical correlation analysis. In *International Conference on Machine Learning*, pages 2741–2750. PMLR.

Hardoon, D. R., Szedmak, S., and Shawe-Taylor, J. (2004). Canonical correlation analysis: An overview with application to learning methods. *Neural computation*, 16(12):2639–2664.

Hotelling, H. (1935). The most predictable criterion. *Journal of educational Psychology*, 26(2):139.

Imtiaz, H. and Sarwate, A. D. (2017). Differentially-private canonical correlation analysis. In *IEEE GlobalSIP*, pages 283–287.

Imtiaz, H. and Sarwate, A. D. (2019). Distributed differentially-private canonical correlation analysis. In *IEEE ICASSP*, pages 3112–3116. IEEE.

Johnson, R. and Zhang, T. (2013). Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26.

Karampatziakis, N. and Mineiro, P. (2014). Discriminative features via generalized eigenvectors. In *International conference on machine learning*, pages 494–502. PMLR.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Lu, Y. and Foster, D. P. (2014). Large scale canonical correlation analysis with iterative least squares. *Advances in Neural Information Processing Systems*, 27.

Montavon, G., Orr, G., and Müller, K.-R. (2012). *Neural networks: tricks of the trade*, volume 7700. springer.

Nesterov, Y. (1983). A method for solving the convex programming problem with convergence rate. *Proceedings of the USSR Academy of Sciences*, 269:543–547.

Sanghavi, R. and Verma, Y. (2022). Multi-view multi-label canonical correlation analysis for cross-modal matching and retrieval. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4701–4710.

Shrestha, S. and Fu, X. (2023). Communication-efficient federated linear and deep generalized canonical correlation analysis. *IEEE Transactions on Signal Processing*, 71:1379–1394.

Snoek, C. G., Worring, M., Van Gemert, J. C., Geusebroek, J.-M., and Smeulders, A. W. (2006). The challenge problem for automated detection of 101 semantic concepts in multimedia. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 421–430.

Thompson, B. (1984). *Canonical correlation analysis: Uses and interpretation*, volume 47. Sage.

Wang, Q. e. a. (2017). Learning privately: Privacy-preserving canonical correlation analysis for cross-media retrieval. In *IEEE Conference on Computer Communications*, pages 1–9. IEEE.

Wang, W., Wang, J., Garber, D., and Srebro, N. (2016). Efficient globally convergent stochastic optimization for canonical correlation analysis. *Advances in Neural Information Processing Systems*, 29.

Westbury, J. R., Turner, G., and Dembowski, J. (1994). X-ray microbeam speech production database user's handbook. *University of Wisconsin*.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation analysis. *Biostatistics*, 10(3):515–534.

Xu, Z. and Li, P. (2019). Towards practical alternating least-squares for cca. *Advances in Neural Information Processing Systems*, 32.

Xu, Z. and Li, P. (2021). On the faster alternating least-squares for cca. In *International Conference on Artificial Intelligence and Statistics*, pages 1621–1629. PMLR.

A Reproducibility Checklist

1. This paper:
   - Includes a conceptual outline and/or pseudocode description of AI methods introduced (**yes**/partial/no/NA)
   - Clearly delineates statements that are opinions, hypotheses, and speculation from objective facts and results (**yes**/no)
   - Provides well-marked pedagogical references for less-familiar readers to gain background necessary to replicate the paper (**yes**/no)

2. Does this paper make theoretical contributions? (**yes**/no)
   If yes, please complete the list below:
   - All assumptions and restrictions are stated clearly and formally. (**yes**/partial/no)
   - All novel claims are stated formally (e.g., in theorem statements). (**yes**/partial/no)
   - Proofs of all novel claims are included. (**yes**/partial/no)
   - Proof sketches or intuitions are given for complex and/or novel results. (**yes**/partial/no)
   - Appropriate citations to theoretical tools used are given. (**yes**/partial/no)
   - All theoretical claims are demonstrated empirically to hold. (yes/**partial**/no/NA)
   - All experimental code used to eliminate or disprove claims is included. (yes/**no**/NA)

3. Does this paper rely on one or more datasets? (**yes**/no)
   If yes, please complete the list below:
   - A motivation is given for why the experiments are conducted on the selected datasets. (**yes**/partial/no/NA)
   - All novel datasets introduced in this paper are included in a data appendix. (yes/partial/no/**NA**)
   - All novel datasets introduced in this paper will be made publicly available upon publication with a license allowing free research use. (yes/partial/no/**NA**)
   - All datasets drawn from the existing literature are accompanied by appropriate citations. (**yes**/no/NA)
   - All datasets drawn from the existing literature are publicly available. (yes/**partial**/no/NA)
   - Datasets that are not publicly available are described in detail, with justification. (yes/partial/no/**NA**)

4. Does this paper include computational experiments? (**yes**/no)
   If yes, please complete the list below:
   - Number/range of values tried per (hyper-)parameter and selection criteria are reported. (yes/**partial**/no/NA)
   - Code for data preprocessing is included in the appendix. (yes/partial/**no**)
   - Source code for conducting and analyzing experiments is included. (**yes**/partial/no)
   - Code will be released publicly upon publication with a permissive license. (yes/**partial**/no)
   - Code includes comments with implementation details and paper references. (yes/**partial**/no)
   - Seed setting methods for stochastic algorithms are described. (yes/partial/**no**/NA)
   - Computing infrastructure (hardware/software specs) is reported. (**yes**/partial/no)
   - Evaluation metrics are formally described with motivations. (**yes**/partial/no)
   - Number of runs per result is specified. (yes/**no**)
   - Performance analysis includes variation, confidence, or distributions. (yes/**no**)
   - Significance of performance differences is assessed with statistical tests. (yes/partial/**no**)
   - Final (hyper-)parameter settings are listed. (**yes**/partial/no/NA)

FedCCA: Federated Canonical Correlation Analysis Supplementary Materials

B ALGORITHMS OF FEDCCA

Algorithm 3 FedCCA-DP client-i:

Input: Operators Op, # local data $X_i \in \mathbb{R}^{d_x \times n_i}$ and $Y_i \in \mathbb{R}^{d_y \times n_i}$, # operation matrix W_{op}.
$Op = 1 : Out = X \cdot W_1 + \varepsilon_1$, $\varepsilon_1 \sim N(0, \sigma_1^2) \times \mathbb{R}^{d_x \times k}$
$Op = 2 : Out = X^\top \cdot W_2 + \varepsilon_2$, $\varepsilon_2 \sim N(0, \sigma_2^2) \times \mathbb{R}^{n_i \times k}$
$Op = 3 : Out = Y \cdot W_3 + \varepsilon_3$, $\varepsilon_3 \sim N(0, \sigma_3^2) \times \mathbb{R}^{d_y \times k}$
$Op = 4 : Out = Y^\top \cdot W_4 + \varepsilon_4$, $\varepsilon_4 \sim N(0, \sigma_4^2) \times \mathbb{R}^{n_i \times k}$
return Out

C DETAILS OF EXPERIMENTS

Datasets: In our experiments, we compare the performance of existing methods, including ALS (Ge et al., 2016) and the truly alternating least squares (TALS) (Xu and Li, 2019) algorithm, with our AMVM-CCA and FedCCA, both with and without differential privacy noise. Five real-world datasets are used: Mediamill (Snoek et al., 2006), JW11 (Westbury et al., 1994), MNIST (LeCun et al., 1998), MFEAT (Dua et al., 2017), and Caltech101 (Fei-Fei et al., 2004).

Table 2: Statistics of Datasets

Data	Description	dx	dy	n
Mediamill	images and its labels	100	120	30000
JW11	acoustic and articulation	273	112	30000
MNIST	left and right image halves	392	392	60000
MFEAT	mfeat-fac and mfeat-pix	216	240	2000
Caltech101	left and right image halves	6144	6144	9144

Hyperparameters. Unless otherwise specified, the hyperparameter settings in this work are as follows: normalization hyperparameter $\alpha = 0.1$, regularization parameters fixed at $r_x = r_y = 0.1$, series order fixed at $m = 3$, and number of clients $M = 10$. Client data partitioning uses random splits with the same scale. All algorithms are implemented in Python and executed on a server equipped with a single Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz and 256 GB of RAM, with no GPU involvement.

D Proof of Convergence (harmonized with Theorem 1)

Theorem 4 (Convergence of AMVM-CCA with truncated von Neumann series)**.** *Let $C_{xx} = \frac{1}{n} XX^\top + r_x I$ and $C_{yy} = \frac{1}{n} YY^\top + r_y I$ with $r_x, r_y > 0$, and let (U, V, Σ) be the top-k CCA factors with canonical correlations $\sigma_1 \geq \cdots \geq \sigma_k > \sigma_{k+1}$. Let (Φ_T, Ψ_T) be the subspaces produced after T AMVM iterations, and define the maximum principal-angle quantity*

$$\theta_T := \max\{\theta'_T, \theta''_T\}, \quad \theta'_T := \cos^{-1}\big(\sigma_k(U^\top C_{xx} \Phi_T)\big),\ \theta''_T := \cos^{-1}\big(\sigma_k(V^\top C_{yy} \Psi_T)\big).$$

Normalize both views once as

$$\bar{X} = \tfrac{1}{N_{xy}} X, \quad \bar{Y} = \tfrac{1}{N_{xy}} Y, \quad N_{xy} := \frac{\max\{\max_j \|x_j\|_2,\ \max_j \|y_j\|_2\}}{\sqrt{\alpha}\,\min\{\sqrt{r_x}, \sqrt{r_y}\}},$$

and set

$$A := \frac{1}{nr_x}\bar{X}\bar{X}^\top, \qquad B := \frac{1}{nr_y}\bar{Y}\bar{Y}^\top.$$

Then $\rho(A) \le \alpha < 1$ *and* $\rho(B) \le \alpha < 1$. *Suppose the Neumann order* m *satisfies*

$$m \;\ge\; \frac{\log\left(\frac{\min\{\sqrt{\lambda_{\max}(\bar{C}_{xx})},\,\sqrt{\lambda_{\max}(\bar{C}_{yy})}\}\;\;\max\{\sin\theta_T,\cos\theta_T\}}{\max\{\sqrt{\frac{r_y}{r_x}}\|\Psi_T\|_F,\;\sqrt{\frac{r_x}{r_y}}\|\Phi_T\|_F\}}\right)}{\log(1/\alpha)} \;+\; \frac{\log\left(\frac{\sigma_k^2-\sigma_{k+1}^2}{12(1-\alpha)}\right)}{\log(1/\alpha)} \;-2. \tag{9}$$

Then there exists an absolute constant $c > 0$ *such that the principal angles contract as*

$$\tan\theta_{t+1} \;\le\; \left(1 - c\cdot\frac{\sigma_k^2-\sigma_{k+1}^2}{\sigma_k^2}\right)\tan\theta_t,$$

and consequently

$$T \;=\; O\left(\frac{\sigma_k^2}{\sigma_k^2-\sigma_{k+1}^2}\;\log\frac{1}{\varsigma\cos\theta_0}\right)$$

iterations suffice to reach $\tan\theta_T \le \varsigma$.

Proof. **(i) One-shot normalization $\Rightarrow$ spectral radius control.** By construction, each column $\bar{x}_j$ of $\bar{X}$ obeys $\|\bar{x}_j\|_2 \le \sqrt{\alpha r_x}$, and similarly $\|\bar{y}_j\|_2 \le \sqrt{\alpha r_y}$. Hence

$$\|A\| = \frac{1}{nr_x}\|\bar{X}\bar{X}^\top\| \le \frac{1}{nr_x}\|\bar{X}\|_F^2 = \frac{1}{nr_x}\sum_{j=1}^n \|\bar{x}_j\|_2^2 \le \frac{1}{nr_x}\cdot n\alpha r_x = \alpha,$$

and analogously $\|B\| \le \alpha$. Therefore $\rho(A), \rho(B) \le \alpha < 1$.

(ii) Truncation residuals of the von Neumann series. AMVM replaces C_{xx}^{-1} and C_{yy}^{-1} by their truncated series

$$C_{xx}^{-1} \approx \tfrac{1}{r_x}\sum_{k=0}^m (-A)^k, \qquad C_{yy}^{-1} \approx \tfrac{1}{r_y}\sum_{k=0}^m (-B)^k.$$

Let the residuals at iteration t be

$$\xi_t \;:=\; -\frac{1}{r_x}\sum_{k=m+1}^\infty (-A)^k\, C_{xy}\Psi_t, \qquad \eta_t \;:=\; -\frac{1}{r_y}\sum_{k=m+1}^\infty (-B)^k\, C_{yx}\Phi_t.$$

Using $\sum_{k=0}^\infty (-A)^k = (I+A)^{-1}$ (valid since $\rho(A) < 1$), we have

$$\xi_t = \frac{1}{r_x}A^{m+1}(I+A)^{-1}C_{xy}\Psi_t, \qquad \eta_t = \frac{1}{r_y}B^{m+1}(I+B)^{-1}C_{yx}\Phi_t.$$

Taking spectral norms and using $\|A^{m+1}\| \le \alpha^{m+1}$ and $\|(I+A)^{-1}\| \le 1/(1-\alpha)$, we get

$$\|\xi_t\| \;\le\; \frac{\alpha^{m+1}}{1-\alpha}\cdot\frac{1}{r_x}\,\|C_{xy}\|\,\|\Psi_t\|, \qquad \|\eta_t\| \;\le\; \frac{\alpha^{m+1}}{1-\alpha}\cdot\frac{1}{r_y}\,\|C_{yx}\|\,\|\Phi_t\|.$$

For the cross-covariance, with the above normalization,

$$\|C_{xy}\| = \left\|\frac{1}{n}\bar{X}\bar{Y}^\top\right\| \le \frac{1}{n}\sum_{j=1}^n \|\bar{x}_j\|_2\,\|\bar{y}_j\|_2 \le \alpha\sqrt{r_x r_y}.$$

Therefore

$$\|\xi_t\| \;\le\; \frac{\alpha^{m+2}}{1-\alpha}\sqrt{\frac{r_y}{r_x}}\,\|\Psi_t\|, \qquad \|\eta_t\| \;\le\; \frac{\alpha^{m+2}}{1-\alpha}\sqrt{\frac{r_x}{r_y}}\,\|\Phi_t\|. \tag{10}$$

Passing to the C_{xx}- and C_{yy}-weighted norms using $\|z\|_{C_{xx}} := \sqrt{z^\top C_{xx} z} \le \sqrt{\lambda_{\max}(C_{xx})}\|z\|$ (and symmetrically for C_{yy}), we have

$$\|\xi_t\|_{C_{xx}} \ \le\ \sqrt{\lambda_{\max}(C_{xx})} \cdot \frac{\alpha^{m+2}}{1-\alpha}\sqrt{\frac{r_y}{r_x}}\,\|\Psi_t\|, \quad \|\eta_t\|_{C_{yy}} \ \le\ \sqrt{\lambda_{\max}(C_{yy})} \cdot \frac{\alpha^{m+2}}{1-\alpha}\sqrt{\frac{r_x}{r_y}}\,\|\Phi_t\|. \tag{11}$$

(iii) From residual control to angle contraction. Consider the (ideal) exact AMVM updates $\tilde{\Phi}_{t+1} \propto C_{xx}^{-1}C_{xy}\Psi_t$ and $\tilde{\Psi}_{t+1} \propto C_{yy}^{-1}C_{yx}\Phi_{t+1}$, which reduce to a block power method on the operator $M := C_{xx}^{-1}C_{xy}C_{yy}^{-1}C_{yx}$ with rate governed by the gap $\sigma_k^2 - \sigma_{k+1}^2$. With truncation, the one-step updates acquire additive perturbations ξ_t and η_t. A standard principal-angle perturbation argument (Davis–Kahan/Wedin style) implies that a sufficient condition for monotone contraction is

$$\max\{\|\xi_t\|_{C_{xx}},\ \|\eta_t\|_{C_{yy}}\} \ \le\ \frac{\sigma_k^2 - \sigma_{k+1}^2}{12}\,\min\{\sin\theta_t,\ \cos\theta_t\}. \tag{12}$$

Combining (11) and (12) and using $\lambda_{\max}(\bar{C}_{xx}) \le \lambda_{\max}(C_{xx})$ and $\lambda_{\max}(\bar{C}_{yy}) \le \lambda_{\max}(C_{yy})$, we obtain the requirement

$$\frac{\alpha^{m+2}}{1-\alpha}\cdot\max\left\{\sqrt{\lambda_{\max}(\bar{C}_{xx})}\sqrt{\tfrac{r_y}{r_x}}\|\Psi_t\|,\ \sqrt{\lambda_{\max}(\bar{C}_{yy})}\sqrt{\tfrac{r_x}{r_y}}\|\Phi_t\|\right\} \ \le\ \frac{\sigma_k^2-\sigma_{k+1}^2}{12}\,\min\{\sin\theta_t,\cos\theta_t\}.$$

To guarantee this simultaneously for $t = 0,\dots,T-1$, it suffices to replace θ_t by θ_T on the right-hand side and $\|\Phi_t\|, \|\Psi_t\|$ by $\|\Phi_T\|_F, \|\Psi_T\|_F$ on the left, yielding (9) after taking logarithms with $\log(1/\alpha) > 0$. Under (12), the block power contraction

$$\tan\theta_{t+1} \ \le\ \left(1 - c\cdot\frac{\sigma_k^2-\sigma_{k+1}^2}{\sigma_k^2}\right)\tan\theta_t$$

follows, and the stated iteration bound is immediate. □

E CLIENT OPERATIONS OF FEDCCA

E.1 Federated Operations

When the operation $\mathrm{Op} = 1$, the server's goal is to compute the matrix product $O_1 = XW_1$, where $X \in \mathbb{R}^{d_x\times n}$ and $W_1 \in \mathbb{R}^{n\times k}$. In a federated learning environment, the data matrix X has already been partitioned among M clients based on their respective data samples:

$$X = [X_1, X_2, \dots, X_M],$$

where $X_i \in \mathbb{R}^{d_x\times n_i}$, and $n = n_1 + n_2 + \dots + n_M$.

Correspondingly, the weight matrix W_1 is partitioned into M submatrices $W_1^1, W_1^2, \dots, W_1^M$, with $W_1^i \in \mathbb{R}^{n_i\times k}$, so that:

$$W_1 = \begin{bmatrix} W_1^1 \\ W_1^2 \\ \vdots \\ W_1^M \end{bmatrix}.$$

Therefore, the matrix multiplication can be expressed as:

$$O_1 = XW_1 = [X_1, X_2, \dots, X_M]\begin{bmatrix} W_1^1 \\ W_1^2 \\ \vdots \\ W_1^M \end{bmatrix} = \sum_{i=1}^{M} X_i W_1^i.$$

Since X_i is already located on its respective client, each client can independently compute $X_i W_1^i$. The server then collects the computed results from all clients and sums them to obtain the final output O_1. This approach ensures that each client only processes its own data and computations, adhering to the principles of federated learning.

Similarly, when the operation Op $=$ 2, the server's goal is to compute the matrix product $O_2 = X^\top W_2$. The weight matrix $W_2 \in \mathbb{R}^{d_x \times k}$ is not partitioned and is sent directly to all clients. Therefore, the matrix multiplication can be expressed as:

$$O_2 = X^\top W_2 = \begin{bmatrix} X_1^\top \\ X_2^\top \\ \vdots \\ X_M^\top \end{bmatrix} W_2 = \begin{bmatrix} X_1^\top W_2 \\ X_2^\top W_2 \\ \vdots \\ X_M^\top W_2 \end{bmatrix} = \begin{bmatrix} O_2^1 \\ O_2^2 \\ \vdots \\ O_2^M \end{bmatrix}.$$

Since X_i is already located on its respective client, each client performs matrix multiplication independently:

$$O_2^i = X_i^\top W_2$$

The server then aggregates the results by concatenating them vertically to obtain the final output:

$$O_2 = \begin{bmatrix} O_2^1 \\ O_2^2 \\ \vdots \\ O_2^M \end{bmatrix}.$$

Due to symmetry, similar federated operations can be performed for operations $Op = 3$ and $Op = 4$ on the matrix data Y as follows.

$$O_3 = Y \cdot W_3 = [Y_1, Y_2, \cdots, Y_M] \begin{bmatrix} W_3^1 \\ W_3^2 \\ \cdots \\ W_3^M \end{bmatrix} = \sum_{i=1}^{M} Y_i \cdot W_3^i,$$

and

$$O_4 = Y^\top W_4 = \begin{bmatrix} Y_1^\top \\ Y_2^\top \\ \vdots \\ Y_M^\top \end{bmatrix} W_4 = \begin{bmatrix} Y_1^\top W_4 \\ Y_2^\top W_4 \\ \vdots \\ Y_M^\top W_4 \end{bmatrix} = \begin{bmatrix} O_4^1 \\ O_4^2 \\ \vdots \\ O_4^M \end{bmatrix}.$$

The update equation of AMVM-CCA proposed in this paper is as follows:

$$\begin{cases} \tilde{\Phi}_{t+1} = \frac{1}{r_x} \sum_{k=0}^{\infty} (-A)^k \bar{C}_{xy} \Psi_t, \quad \Phi_{t+1} = \tilde{\Phi}_{t+1} \left(\tilde{\Phi}_{t+1}^\top \bar{C}_{xx} \tilde{\Phi}_{t+1} \right)^{-\frac{1}{2}}, \\ \tilde{\Psi}_{t+1} = \frac{1}{r_y} \sum_{k=0}^{\infty} (-B)^k \bar{C}_{xy}^\top \Phi_t, \quad \Psi_{t+1} = \tilde{\Psi}_{t+1} \left(\tilde{\Psi}_{t+1}^\top \bar{C}_{yy} \tilde{\Psi}_{t+1} \right)^{-\frac{1}{2}}. \end{cases} \tag{13}$$

For computational efficiency, we approximate the infinite sums by considering only the first m terms, treating the remaining terms as residual errors. The residual terms are quantified as:

$$\begin{cases} \tilde{\Phi}_{t+1} = \bar{C}_{xx}^{-1} \bar{C}_{xy} \Psi_t + \xi_t, \quad \xi_t = -\frac{1}{r_x} \sum_{k=m+1}^{\infty} (-A)^k \bar{C}_{xy} \Psi_t \\ \tilde{\Psi}_{t+1} = \bar{C}_{yy}^{-1} \bar{C}_{xy}^\top \Phi_t + \eta_t, \quad \eta_t = -\frac{1}{r_y} \sum_{k=m+1}^{\infty} (-B)^k \bar{C}_{xy}^\top \Phi_t. \end{cases} \tag{14}$$

Thus, the complete update formula is shown by combining (13) with (14) as follows:

$$\begin{cases} \tilde{\Phi}_{t+1} = \frac{1}{r_x} \sum_{k=0}^{m} (-A)^k \bar{C}_{xy} \Psi_t + \xi_t, \quad \xi_t = -\frac{1}{r_x} \sum_{k=m+1}^{\infty} (-A)^k \bar{C}_{xy} \Psi_t \\ \Phi_{t+1} = \tilde{\Phi}_{t+1} \left(\tilde{\Phi}_{t+1}^\top \bar{C}_{xx} \tilde{\Phi}_{t+1} \right)^{-\frac{1}{2}}, \\ \tilde{\Psi}_{t+1} = \frac{1}{r_y} \sum_{k=0}^{m} (-B)^k \bar{C}_{xy}^\top \Phi_t + \eta_t, \quad \eta_t = -\frac{1}{r_y} \sum_{k=m+1}^{\infty} (-B)^k \bar{C}_{xy}^\top \Phi_t. \\ \Psi_{t+1} = \tilde{\Psi}_{t+1} \left(\tilde{\Psi}_{t+1}^\top \bar{C}_{yy} \tilde{\Psi}_{t+1} \right)^{-\frac{1}{2}}. \end{cases}$$

Next, we will introduce how the complete update process can be accomplished using only the above four operations. Ignoring the residual terms, the update of $\tilde{\Phi}_{t+1}$ can be presented as follows:

$$\tilde{\Phi}_{t+1} = \frac{1}{nr_x} \sum_{k=0}^{m} (-A)^k XY^\top \Psi_t.$$

To facilitate the computation, we define a sequence of intermediate matrices H_k as:

$$\begin{aligned}
H_0 &= \frac{1}{nr_x} XY^\top \Psi_t, \\
H_1 &= \frac{1}{nr_x} XX^\top H_0 = \frac{1}{(nr_x)^2} XX^\top XY^\top \Psi_t, \\
H_2 &= \frac{1}{nr_x} XX^\top H_1 = \frac{1}{(nr_x)^3} (XX^\top)^2 XY^\top \Psi_t, \\
&\vdots \\
H_m &= \frac{1}{nr_x} XX^\top H_{m-1} = \frac{1}{(nr_x)^{m+1}} (XX^\top)^m XY^\top \Psi_t.
\end{aligned}$$

Therefore, the update formula becomes:

$$\tilde{\Phi}_{t+1} = \sum_{k=0}^{m} (-1)^k H_k.$$

To express the computation using the four federated operations, we rewrite the definitions of H_k in terms of these operations:

$$\begin{aligned}
H_0 &= \frac{1}{nr_x} O_1(O_4(\Psi_t)), \\
H_1 &= \frac{1}{nr_x} O_1(O_2(H_0)), \\
H_2 &= \frac{1}{nr_x} O_1(O_2(H_1)), \\
&\vdots \\
H_m &= \frac{1}{nr_x} O_1(O_2(H_{m-1})).
\end{aligned} \tag{15}$$

By iteratively applying these operations, each H_k is computed using only the four federated operations, without requiring clients to share their raw data. Due to the symmetry in the update rules, the updating of $\tilde{\Psi}_{t+1}$ can be performed similarly by appropriately applying the federated operations as follows:

$$\begin{aligned}
H'_0 &= \frac{1}{nr_y} O_3(O_2(\Phi_t)), \\
H'_1 &= \frac{1}{nr_y} O_3(O_4(H'_0)), \\
H'_2 &= \frac{1}{nr_y} O_3(O_4(H'_1)), \\
&\vdots \\
H'_m &= \frac{1}{nr_y} O_3(O_4(H'_{m-1})).
\end{aligned} \tag{16}$$

Algorithm 4 FedCCA Server for Updating Ψ:

Input: Series order m, # client number M, # projection matrices Φ_t, Ψ_t,# regularization parameters r_x, r_y

Federated least-square solver of ψ (In: ϕ_{t+1}, Out:$\tilde{\Psi}_{t+1}$)

Initialize $\mathbf{P}_0 = \mathbf{0} \in \mathbb{R}^{d_y \times k}$, where $n = \sum_{i=1}^{M} n_i$.

for $i = 1$ **to** M **do**

Receive $\text{Out}_i = \text{client-}i(Op = 2,\, W_{op} = \Phi_{t+1})$

Receive $\text{Out}_i = \text{client-}i(Op = 3,\, W_{op} = \text{Out}_i)$

Update $\mathbf{P}_0 = \mathbf{P}_0 + \text{Out}_i$

end

Compute $\mathbf{P}_0 = \frac{\mathbf{P}_0}{n \cdot r_y}$.

Initialize $\tilde{\Psi}_{t+1} = \mathbf{P}_0$.

for $j = 1$ **to** m **do**

Initialize $\mathbf{P}_j = \mathbf{0} \in \mathbb{R}^{d_y \times k}$.

for $i = 1$ **to** M **do**

Receive $\text{Out}_i = \text{client-}i(Op = 4,\, W_{op} = \mathbf{P}_{j-1})$

Receive $\text{Out}_i = \text{client-}i(Op = 3,\, W_{op} = \text{Out}_i)$

Update $\mathbf{P}_j = \mathbf{P}_j + \text{Out}_i$

end

Compute $\mathbf{P}_j = \frac{\mathbf{P}_j}{n \cdot r_y}$.

Update $\tilde{\Psi}_{t+1} = \tilde{\Psi}_{t+1} + (-1)^j \mathbf{P}_j$.

end

Return $\tilde{\Psi}_{t+1}$.

Federated orthogonalization of ψ (In: $\tilde{\Psi}_{t+1}$, Out: Ψ_{t+1})

Initialize $\text{Norm} = \mathbf{0} \in \mathbb{R}^{k \times k}$.

for $i = 1$ **to** M **do**

Receive $\text{Out}_i = \text{client-}i(Op = 4,\, W_{op} = \tilde{\Psi}_{t+1})$

Receive $\text{Out}_i = \text{client-}i(Op = 3,\, W_{op} = \text{Out}_i)$

Update $\text{Norm} = \text{Norm} + \tilde{\Psi}_{t+1}^{\top} \cdot \text{Out}_i$

end

Compute $\text{Norm} = \frac{1}{n} \cdot \text{Norm}$.

Compute $\Psi_{t+1} = \tilde{\Psi}_{t+1} \cdot \text{Norm}^{-1/2}$.

Return Ψ_{t+1}.

Therefore, the update formula becomes:

$$\tilde{\Psi}_{t+1} = \sum_{k=0}^{m} (-1)^k H'_k.$$

Besides, the orthogonalization process can also be performed by appropriately applying the federated operations easily as follows:

$$\Phi_{t+1} = \tilde{\Phi}_{t+1} \left(\frac{1}{n \cdot r_x} \tilde{\Phi}_{t+1}^{\top} O_1(O_2(\tilde{\Phi}_{t+1})) \right)^{-\frac{1}{2}},$$

and

$$\Psi_{t+1} = \tilde{\Psi}_{t+1} \left(\frac{1}{n \cdot r_y} \tilde{\Psi}_{t+1}^{\top} O_3(O_4(\tilde{\Psi}_{t+1})) \right)^{-\frac{1}{2}}$$

To summarize, due to the convenience brought by the von Neumann series decomposition, the update process of AMVM-CCA can be fully expanded into matrix multiplication and addition operations. By mapping these operations into the above four federated operations, we can easily implement Federated CCA.

In addition, we present the FedCCA Server algorithm for Updating Ψ as follows:

E.2 Client Complexity

Then we want to demonstrate the computational efficiency of FedCCA that the entire update process can be efficiently carried out using only the four federated operations.

Computation complexity: Each client in FedCCA performs one of the four matrix multiplication operations when invoked. The computational complexities for these operations on a client are as follows:

- **Op1** ($X_i W_1^i$): Multiplying $X_i \in \mathbb{R}^{d_x \times n_i}$ by $W_1^i \in \mathbb{R}^{n_i \times k}$,
  resulting in complexity: $O(d_x n_i k)$.
- **Op2** ($X_i^\top W$): Multiplying $X_i^\top \in \mathbb{R}^{n_i \times d_x}$ by $W \in \mathbb{R}^{d_x \times k}$,
  resulting in complexity: $O(n_i d_x k)$.
- **Op3** ($Y_i W_3^i$): Multiplying $Y_i \in \mathbb{R}^{d_y \times n_i}$ by $W_3^i \in \mathbb{R}^{n_i \times k}$,
  resulting in complexity: $O(d_y n_i k)$.
- **Op4** ($Y_i^\top W$): Multiplying $Y_i^\top \in \mathbb{R}^{n_i \times d_y}$ by $W \in \mathbb{R}^{d_y \times k}$,
  resulting in complexity: $O(n_i d_y k)$.

Assuming that $n_i \approx \dfrac{n}{M}$ and $d_2 = \max(d_x, d_y)$, and considering that $k \ll d_x \ll n_i$, all operations have similar computational complexities. Therefore, the per-client computational complexity is dominated by:

$$O(n_i d_2 k) = O\left(\frac{n d_2 k}{M}\right).$$

Storage complexity: Each client in FedCCA needs to store its local data $X_i \in \mathbb{R}^{d_x \times n_i}$ and $Y_i \in \mathbb{R}^{d_y \times n_i}$, requiring permanent storage space of $O\left((d_x + d_y) n_i\right)$. During computation, clients only need temporary storage for input and output matrices of size up to $\max(d_x k,\, d_y k,\, n_i k)$. Given the assumptions $n_i \approx \dfrac{n}{M}$ and $k \ll d_x \ll n_i$, the temporary storage $O(n_i k)$ is significantly less than the permanent storage and can be considered negligible. Therefore, the overall space complexity at each client is dominated by the storage of local data, resulting in a space complexity of $O\left((d_x + d_y) n_i\right) \approx O\left(\dfrac{n d_1}{M}\right), d_1 = d_x + d_y$.

Communication complexity: In each round of FedCCA, a client communicates with the server by receiving an input matrix W and sending back the result of a matrix multiplication. The communication complexity per client per round is determined by the sizes of these matrices. Specifically, the input matrix W has dimensions up to $\max(n_i \times k,\, d_2 \times k)$, where $n_i \approx \dfrac{n}{M}$ and $d_2 = \max(d_x,\, d_y)$. The output matrix, resulting from the matrix multiplication, also has dimensions up to $\max(n_i \times k,\, d_2 \times k)$. The total communication between clients and server complexity per round is $O(M(n_i + d_2)k)$, which simplifies to $O\left((n + M d_2)k\right)$. This accounts for both the data sent from the server to the client and the results returned from the client to the server.

F Detailed Proofs of Differential Privacy

We give the detailed proof using Rényi DP / moments accountant, which is equivalent to the direct Gaussian-mechanism analysis in the main text. The final noise lower bounds coincide with Theorem 2.

F.1 Setting, adjacency, and query counting

Record-level adjacency. Two datasets are adjacent if they differ in exactly one row of X or Y on a given client. We add Gaussian noise to each of the four client-side sketches (Ops 1–4) per outer round, on both Φ- and Ψ-sides.

One-shot normalization and operator bounds. Following Sec. 4.1, define $\bar{X} = X/N_{xy}$, $\bar{Y} = Y/N_{xy}$ with $N_{xy} := \frac{\max\{\max_j \|x_j\|_2, \max_j \|y_j\|_2\}}{\sqrt{\alpha}\min\{\sqrt{r_x}, \sqrt{r_y}\}}$, so that $\|\bar{x}_j\|_2 \le \sqrt{\alpha r_x}$ and $\|\bar{y}_j\|_2 \le \sqrt{\alpha r_y}$. We use the standard covariances $C_{xx} = \frac{1}{n} X X^\top + r_x I$, $C_{yy} = \frac{1}{n} Y Y^\top + r_y I$.

Query count. Each Φ-side update uses $(2{+}2m)$ primitive ops (one Op3 $\to$ Op2 and m repetitions of Op1 $\to$ Op2); symmetrically for Ψ-side. With T rounds and M clients,

$$Q \;=\; 4(2{+}2m)TM$$

total (private) queries are issued. We also denote $N_1 = N_2 = N_3 = N_4 = (2{+}2m)TM$ as the number of calls of each op type.

F.2 Exact ℓ_2-sensitivity calculations (Ops 1–4)

We work with neighboring datasets (X, Y) and (X', Y) that differ in one row of X (the Y case is symmetric). We carefully bound the *operator norm* effects and keep constants explicit.

Lemma 1 (Sensitivity of Op1: $X \cdot A$)**.** *Let* $\mathrm{Op1}(X) = XA$ *with* $A \in \mathbb{R}^{n\times k}$ *of the form*

$$A_j \;=\; \frac{(X^\top X)^j\, Y^\top \Psi_t}{(nr_x)^{j+1}}, \qquad j = 0, 1, \ldots, m.$$

For adjacent X, X' differing in the first row,

$$\Delta_1 \;:=\; \|XA - X'A\|_2 \;\le\; 2\alpha \sqrt{\tfrac{r_y}{nr_x}}\; \|\Psi_t\|_2.$$

Proof. *Write* $XA{-}X'A = (x_1{-}x_1')\,A_{(1,:)}$ *(only the first row contributes). Hence* $\|XA{-}X'A\|_2 \le \|x_1{-}x_1'\|_2\,\|A\|_2$*. By normalization,* $\|x_1 - x_1'\|_2 \le 2\sqrt{\alpha r_x}$*. For* A_j*,*

$$\|A_j\|_2 \le \left\|\tfrac{X^\top X}{nr_x}\right\|_2^j \cdot \left\|\tfrac{Y^\top}{nr_x}\right\|_2 \cdot \|\Psi_t\|_2 \le \alpha^j \cdot \tfrac{\|Y^\top\|_2}{nr_x} \cdot \|\Psi_t\|_2.$$

Moreover $\|Y^\top\|_2 = \|Y\|_2 \le \sqrt{n\,\alpha r_y}$*, so* $\|A_j\|_2 \le \alpha^j \cdot \frac{\sqrt{n\,\alpha r_y}}{nr_x}\|\Psi_t\|_2 = \alpha^j \cdot \frac{\sqrt{\alpha r_y}}{r_x\sqrt{n}}\|\Psi_t\|_2$*. Since* $\alpha \in (0,1)$*,* $\max_{0\le j\le m}\alpha^j = 1$ *at* $j = 0$*, thus* $\|A\|_2 \le \frac{\sqrt{\alpha r_y}}{r_x\sqrt{n}}\|\Psi_t\|_2$*. Combining gives* $\Delta_1 \le 2\sqrt{\alpha r_x} \cdot \frac{\sqrt{\alpha r_y}}{r_x\sqrt{n}}\|\Psi_t\|_2 = 2\alpha\sqrt{\frac{r_y}{nr_x}}\,\|\Psi_t\|_2$*.* □

Lemma 2 (Sensitivity of Op2: $X^\top \cdot A'$)**.** *Let* $\mathrm{Op2}(X) = X^\top A'$ *with* $A' \in \mathbb{R}^{d_x\times k}$ *of the form*

$$A'_j \;=\; \frac{(XX^\top)^j\, XY^\top \Psi_t}{(nr_x)^{j+1}}, \qquad j = 0, 1, \ldots, m.$$

Then

$$\Delta_2 \;:=\; \|X^\top A' - X'^\top A'\|_2 \;\le\; 2\alpha\,\sqrt{\alpha r_y}\;\|\Psi_t\|_2.$$

Proof. $\|X^\top A' - X'^\top A'\|_2 \le \|X - X'\|_2 \cdot \|A'\|_2$ *and* $\|X - X'\|_2 \le \|x_1 - x_1'\|_2 \le 2\sqrt{\alpha r_x}$*. For* A'_j*,*

$$\|A'_j\|_2 \le \left\|\tfrac{XX^\top}{nr_x}\right\|_2^j \cdot \left\|\tfrac{XY^\top}{nr_x}\right\|_2 \cdot \|\Psi_t\|_2.$$

Use $\left\|\frac{XX^\top}{nr_x}\right\|_2 \le \alpha$ *and* $\|XY^\top\|_2 \le \|X\|_2\|Y\|_2 \le \sqrt{n\alpha r_x}\sqrt{n\alpha r_y} = n\alpha\sqrt{r_x r_y}$*, hence* $\|A'_j\|_2 \le \alpha^j \cdot \frac{n\alpha\sqrt{r_x r_y}}{nr_x}\|\Psi_t\|_2 = \alpha^{j+1}\sqrt{\frac{r_y}{r_x}}\,\|\Psi_t\|_2$*. Maximizing over j at $j = 0$ gives* $\|A'\|_2 \le \alpha\sqrt{\frac{r_y}{r_x}}\|\Psi_t\|_2$*, so* $\Delta_2 \le 2\sqrt{\alpha r_x} \cdot \alpha\sqrt{\frac{r_y}{r_x}}\|\Psi_t\|_2 = 2\alpha\sqrt{\alpha r_y}\,\|\Psi_t\|_2$*.* □

Lemma 3 (Sensitivity of Op3: $Y \cdot B$)**.** *For* $\mathrm{Op3}(Y) = YB$ *with* $B_j = \frac{(Y^\top Y)^j\, X^\top \Phi_{t+1}}{(nr_y)^{j+1}}$*, we have*

$$\Delta_3 \;:=\; \|YB - Y'B\|_2 \;\le\; 2\alpha\sqrt{\tfrac{r_x}{nr_y}}\;\|\Phi_{t+1}\|_2.$$

Proof. *Symmetric to Lemma 1 with* $X \leftrightarrow Y$*,* $r_x \leftrightarrow r_y$*, and* $\Psi_t \leftrightarrow \Phi_{t+1}$*.* □

Lemma 4 (Sensitivity of Op4: $Y^\top \cdot B'$)**.** *For* $\mathrm{Op4}(Y) = Y^\top B'$ *with* $B'_j = \frac{(YY^\top)^j\, YX^\top \Phi_{t+1}}{(nr_y)^{j+1}}$*, we have*

$$\Delta_4 \;:=\; \|Y^\top B' - Y'^\top B'\|_2 \;\le\; 2\alpha\,\sqrt{\alpha r_x}\;\|\Phi_{t+1}\|_2.$$

Proof. *Symmetric to Lemma 2.* □

F.3 Rényi DP (RDP) for Gaussian, adaptive composition, and conversion

Lemma 5 (Per-query RDP of Gaussian mechanism)**.** *For a function f with ℓ_2-sensitivity Δ and Gaussian mechanism $\mathcal{M}(x) = f(x) + \mathcal{N}(0, \sigma^2 I)$, the RDP at order $q > 1$ is*

$$\rho(q) \;=\; \frac{q\,\Delta^2}{2\sigma^2}.$$

Proof. *Let $P = \mathcal{N}(f(x), \sigma^2 I)$ and $Q = \mathcal{N}(f(x'), \sigma^2 I)$. The order-$q$ Rényi divergence is $D_q(P\|Q) = \frac{q}{2\sigma^2}\|f(x) - f(x')\|_2^2 \le \frac{q\Delta^2}{2\sigma^2}$.* □

Lemma 6 (Adaptive composition for RDP)**.** *Let mechanisms $\{\mathcal{M}_\ell\}_{\ell=1}^L$ be applied adaptively (the ℓ-th mechanism may depend on previous outputs). If $\mathcal{M}_\ell$ is $\rho_\ell(q)$-RDP at order $q > 1$, then the composition is $\big(\sum_{\ell=1}^L \rho_\ell(q)\big)$-RDP at the same order.* Proof. *Standard; follows from the additivity of Rényi divergence under product measures and the post-processing invariance.* □

Lemma 7 (RDP→ (ε, δ) conversion)**.** *If a mechanism is $\rho(q)$-RDP for some $q > 1$, then for any $\delta \in (0, 1)$ it is (ε, δ)-DP with*

$$\varepsilon(\delta) \;=\; \rho(q) + \frac{\log(1/\delta)}{q-1}.$$

Moreover, minimizing over $q > 1$ yields the tightest bound: $\varepsilon(\delta) = \min_{q>1}\big\{\rho(q) + \frac{\log(1/\delta)}{q-1}\big\}$. Proof. *Standard; see RDP literature / moments accountant.* □

F.4 Total RDP for FedCCA and closed-form sufficient noise bounds

By Lemma 5 and Lemma 6, after Q queries with possibly mixed types $j \in \{1, 2, 3, 4\}$ and per-op noises σ_j,

$$\rho_{\text{tot}}(q) \;=\; \sum_{j=1}^{4} N_j \cdot \frac{q\,\Delta_j^2}{2\sigma_j^2}, \qquad N_j = (2{+}2m)TM.$$

By Lemma 7,

$$\varepsilon(\delta) \;=\; \min_{q>1}\Big\{\sum_{j=1}^{4} \frac{qN_j\Delta_j^2}{2\sigma_j^2} \;+\; \frac{\log(1/\delta)}{q-1}\Big\}. \tag{$\star$}$$

Sufficient bounds by fixing q. Fix any $q > 1$ such that $\varepsilon > \frac{\log(1/\delta)}{q-1}$ (otherwise increase q). A sufficient condition for ($\star$) is to enforce each term to be *individually* below $\varepsilon - \frac{\log(1/\delta)}{q-1}$:

$$\frac{qN_j\Delta_j^2}{2\sigma_j^2} \;\le\; \varepsilon - \frac{\log(1/\delta)}{q-1}, \qquad j = 1, 2, 3, 4.$$

Solving for σ_j yields

$$\sigma_j \;\ge\; \Delta_j \cdot \sqrt{\frac{qN_j}{2\left(\varepsilon - \frac{\log(1/\delta)}{q-1}\right)}}. \tag{17}$$

Plugging the sensitivities from Lemmas 1–4 and $N_j = (2{+}2m)TM$ gives the explicit four inequalities:

$$\sigma_1 \geq \sqrt{\frac{q(2{+}2m)TM}{2\left(\varepsilon - \frac{\log(1/\delta)}{q-1}\right)}} \cdot \left(2\alpha\sqrt{\tfrac{r_y}{nr_x}}\,\|\Psi_t\|_2\right),$$

$$\sigma_2 \geq \sqrt{\frac{q(2{+}2m)TM}{2\left(\varepsilon - \frac{\log(1/\delta)}{q-1}\right)}} \cdot \left(2\alpha\sqrt{\alpha r_y}\,\|\Psi_t\|_2\right),$$

$$\sigma_3 \geq \sqrt{\frac{q(2{+}2m)TM}{2\left(\varepsilon - \frac{\log(1/\delta)}{q-1}\right)}} \cdot \left(2\alpha\sqrt{\tfrac{r_x}{nr_y}}\,\|\Phi_{t+1}\|_2\right),$$

$$\sigma_4 \geq \sqrt{\frac{q(2{+}2m)TM}{2\left(\varepsilon - \frac{\log(1/\delta)}{q-1}\right)}} \cdot \left(2\alpha\sqrt{\alpha r_x}\,\|\Phi_{t+1}\|_2\right).$$

Relation to the textbook (ε, δ)-Gaussian bound. If one prefers to avoid the optimization over q, upper-bound $(\star)$ by taking a crude $\min_q$ upper envelope yields the familiar sufficient condition

$$\sigma_j \;\geq\; \frac{\Delta_j}{\varepsilon}\sqrt{2N_j \log\frac{1.25}{\delta}},$$

which is the closed form used in the main-text DP lower-bound theorem (after grouping constants). We emphasize that our (17) with a small grid over q typically gives a tighter (smaller) noise for the same (ε, δ).

Theorem 5 (DP noise lower bounds, RDP form)**.** *Under record-level adjacency and the one-shot normalization, the FedCCA protocol with per-op Gaussian noises $\sigma_1, \sigma_2, \sigma_3, \sigma_4$ is (ε, δ)-DP provided there exists $q > 1$ such that the four inequalities in* (17) *hold with $N_j = (2{+}2m)TM$ and sensitivities* $(\Delta_1, \Delta_2, \Delta_3, \Delta_4)$ *given by Lemmas 1–4. Equivalently, using the textbook Gaussian bound yields the sufficient conditions*

$$\sigma_1 \geq \frac{2\sqrt{2}\,\alpha\,\|\Psi_t\|_2}{\varepsilon}\sqrt{\tfrac{r_y}{nr_x}\,\log\tfrac{1.25}{\delta}}, \qquad \sigma_2 \geq \frac{2\sqrt{2}\,\alpha\,\|\Psi_t\|_2}{\varepsilon}\sqrt{\alpha r_y\,\log\tfrac{1.25}{\delta}},$$
$$\sigma_3 \geq \frac{2\sqrt{2}\,\alpha\,\|\Phi_{t+1}\|_2}{\varepsilon}\sqrt{\tfrac{r_x}{nr_y}\,\log\tfrac{1.25}{\delta}}, \qquad \sigma_4 \geq \frac{2\sqrt{2}\,\alpha\,\|\Phi_{t+1}\|_2}{\varepsilon}\sqrt{\alpha r_x\,\log\tfrac{1.25}{\delta}}.$$

F.5 Consistency with convergence-side noise upper bounds

The RDP analysis above shows the *privacy-side* dependence of σ_j on the total query count Q and the series order m (via N_j). As noted in Sec. 4.3, holding σ_j fixed implies $\varepsilon(\delta)$ grows linearly in Q; keeping (ε, δ) fixed while Q increases forces $\sigma_j = \Theta(\sqrt{Q})$, which tightens the convergence upper bound in Theorem 3. This establishes the stated privacy–accuracy trade-off.

F.6 Adaptivity and post-processing

All multipliers W_{op} broadcast by the server may be adaptive functions of past messages. RDP composition (Lemma 6) is valid in this fully adaptive setting. Post-processing by the server (e.g., $k \times k$ re-orthogonalization) does not weaken privacy.

F.7 Notation remarks

We consistently use *von Neumann* for the series in Sec. 4.1 and keep the RDP order symbol q distinct from the spectral-radius parameter α.

G CONVERGENCE OF FEDCCA WITH DIFFERENTIAL PRIVACY

The introduction of noise significantly impacts the convergence of FedCCA. This impact is reflected in the addition of noise to the update formula.

The noisy update formula for H'_{m+1} is given by:

$$H'_{m+1} = \frac{1}{nr_x} \left(X \left(X^\top H'_m + \varepsilon_2 \right) + \varepsilon_1 \right), \tag{18}$$

where H'_m is the perturbed version of H_m, and ε_1, ε_2 are Gaussian noise matrices introduced to guarantee differential privacy.

We define H'_m as:

$$H'_m = H_m + \beta_m^1 \varepsilon_1 + \beta_m^2 \varepsilon_2 + \beta_m^3 \varepsilon_3 + \beta_m^4 \varepsilon_4, \tag{19}$$

where β_m^j are coefficients to be determined, and ε_j are Gaussian noise terms.

Substituting (19) into (18), we expand H'_{m+1}:

$$\begin{aligned} H'_{m+1} &= \frac{1}{nr_x} \left(X \left(X^\top \left(H_m + \sum_{j=1}^{4} \beta_m^j \varepsilon_j \right) + \varepsilon_2 \right) + \varepsilon_1 \right) \\ &= \frac{1}{nr_x} \left(XX^\top H_m + XX^\top \sum_{j=1}^{4} \beta_m^j \varepsilon_j + X\varepsilon_2 + \varepsilon_1 \right). \end{aligned} \tag{20}$$

Recognizing that the original (noise-free) update is:

$$H_{m+1} = \frac{1}{nr_x} XX^\top H_m,$$

we rewrite (20) as:

$$H'_{m+1} = H_{m+1} + \frac{1}{nr_x} \left(XX^\top \sum_{j=1}^{4} \beta_m^j \varepsilon_j + X\varepsilon_2 + \varepsilon_1 \right). \tag{21}$$

Our goal is to express the perturbations in terms of the noise terms ε_j. We initialize the coefficients as:

$$\beta_0^1 = \frac{1}{nr_x}, \quad \beta_0^2 = 0, \quad \beta_0^3 = 0, \quad \beta_0^4 = \frac{X}{nr_x}.$$

By iteratively applying (21), we derive the expressions for β_m^j:

$$\begin{cases} \beta_m^1 = \frac{1}{nr_x} \sum_{k=0}^{m} \left(\frac{XX^\top}{nr_x} \right)^k, \\ \beta_m^2 = \frac{X}{nr_x} \sum_{k=0}^{m-1} \left(\frac{XX^\top}{nr_x} \right)^k, \\ \beta_m^3 = 0, \\ \beta_m^4 = \left(\frac{XX^\top}{nr_x} \right)^m \frac{X}{nr_x}. \end{cases}$$

The total noise can be considered as part of the residual term in the update of $\tilde{\Phi}_{t+1}$:

$$\tilde{\Phi}_{t+1} = \sum_{k=0}^{m} (-1)^k H_k + \xi'_t,$$

where the modified residual ξ'_t is:

$$\xi'_t = \xi_t + \sum_{k=0}^{m} (-1)^k \left(\sum_{j=1}^{4} \beta_k^j \varepsilon_j \right).$$

Assuming that all noise terms are equal, $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = \varepsilon$, we can bound the norm of ξ'_t:

$$\|\xi'_t\| \le \|\xi_t\| + \left\| \sum_{k=0}^{m} \sum_{j=1}^{4} \beta_k^j \varepsilon \right\| \le \|\xi_t\| + \left(\sum_{k=0}^{m} \sum_{j=1}^{4} \|\beta_k^j\| \right) \|\varepsilon\|.$$

Using the property $\left\|\frac{XX^\top}{nr_x}\right\|_2 \leq \alpha \ll 1$, and applying geometric series approximations, we have:

$$\sum_{k=0}^{m} \|\beta_k^1\| \leq \frac{1}{nr_x}\sum_{k=0}^{m}\alpha^k \leq \frac{1}{nr_x}\frac{1}{1-\alpha},$$
$$\sum_{k=0}^{m} \|\beta_k^2\| \leq \frac{\|X\|}{nr_x}\sum_{k=0}^{m-1}\alpha^k \leq \frac{\|X\|}{nr_x}\frac{1}{1-\alpha}.$$

Assuming $\|X\| \leq \sqrt{nr_x}$, we obtain:

$$\|\xi_t'\| \leq \|\xi_t\| + \frac{m+1}{1-\alpha}\frac{1+\sqrt{n\alpha r_x}}{nr_x}\|\varepsilon\|.$$

From previous results, we have:

$$\|\xi_t\| \leq \frac{\alpha^{m+2}}{1-\alpha}\sqrt{\frac{r_y}{r_x}}\|\Psi_t\|.$$

Since $\alpha \ll 1$, the term α^{m+2} becomes negligible, and we can approximate $\|\xi_t\| \approx 0$. Therefore, the bound simplifies to:

$$\|\xi_t'\| \leq \frac{\alpha^{m+2}}{1-\alpha}\sqrt{\frac{r_y}{r_x}} \cdot \|\Psi_t\| + \frac{m+1}{1-\alpha}\frac{\sqrt{\alpha n r_x}+1}{nr_x}\|\varepsilon\|$$

For FedCCA to converge, the residual ξ_t' must satisfy:

$$\|\xi_t'\|\lambda_{\min}^{-1/2}(C_{xx}) \leq \frac{\sigma_k^2 - \sigma_{k+1}^2}{12}\min\{\sin\theta_t, \cos\theta_t\}, \tag{22}$$

and similarly for the Y side with C_{yy}.

Assuming Gaussian noise with variance σ^2, and utilizing concentration inequalities for the Gaussian distribution, we have with probability at least $1-\mu$:

$$\|\varepsilon\| \leq \sigma\Phi^{-1}\left(\frac{\mu}{2}\right), \tag{23}$$

where Φ^{-1} is the inverse cumulative distribution function of the standard normal distribution. Substituting (23) into (22), we derive an upper bound on σ:

$$\sigma \leq \frac{C_1\min\{\sin\theta_T, \cos\theta_T\} - C_2}{C_3\Phi^{-1}\left(\frac{\mu}{2}\right)}, \tag{24}$$

where:

$$C_1 = \frac{\sigma_k^2 - \sigma_{k+1}^2}{12}\min\left\{\lambda_{\min}^{1/2}(C_{xx}), \lambda_{\min}^{1/2}(C_{yy})\right\},$$
$$C_2 = \frac{\alpha^{(m+2)}}{1-\alpha}\max\left(\sqrt{\frac{r_y}{r_x}}\|\Psi_T\|, \sqrt{\frac{r_x}{r_y}}\|\Phi_T\|\right)$$
$$C_3 = \frac{m+1}{1-\alpha}\max\left\{\frac{1+\sqrt{n\alpha r_x}}{nr_x}, \frac{1+\sqrt{n\alpha r_y}}{nr_y}\right\}.$$

Theorem 6. *Assuming consistent variance σ for the Gaussian noise terms, FedCCA converges with at least $1-\mu$ probability if σ satisfies the upper bound in* (24).

By ensuring that the noise variance σ^2 is below the derived upper bound, we guarantee that the introduction of Gaussian noise for differential privacy does not prevent the convergence of FedCCA. This balance allows us to achieve both privacy and utility in federated learning.