FEDWNNM: A FEDERATED FRAMEWORK FOR MATRIX COMPLETION WITH PROVABLE PRIVACY-ACCURACY TRADE-OFFS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated learning prevents raw data aggregation on a central server, posing challenges for matrix completion. While matrix factorization methods are suitable for federated settings, Weighted Nuclear Norm Minimization (WNNM) outperforms them in centralized contexts by recovering low-rank and nearly low-rank matrices with rapidly decaying singular values. However, WNNM relies on Singular Value Decomposition (SVD), a global operation requiring full data access, which violates federated learning's privacy principles. To address this, we propose Federated WNNM (FedWNNM), a privacy-preserving solver for federated matrix completion. Our key contribution is a Privacy-Preserving Federated SVD (PPF-SVD) subroutine, where clients generate low-dimensional, privacy-preserving sketches of local data using structured random projections. These sketches are aggregated on a central server to approximate global singular values and right singular vectors without accessing raw data. Theoretical analysis provides upper and lower bounds on approximation error, quantifying the privacy-accuracy trade-off. Experiments demonstrate that FedWNNM achieves state-of-the-art recovery performance with formal privacy guarantees.

1 INTRODUCTION

Data is increasingly generated and stored in a decentralized manner across numerous clients, such as mobile devices, hospitals, and financial institutions (McMahan et al., 2017; Kairouz et al., 2021; Saif et al., 2025). This has led to the development of a new collaborative learning framework known as Federated Learning (FL). FL facilitates model training on distributed data without centralizing raw, and often sensitive, information (Gabrielli et al., 2023; Siniosoglou et al., 2024; Yurdem et al., 2024). A challenge in these real-world settings is data incompleteness, which severely hinders the potential of distributed data for critical applications. For example, user rating matrices in federated recommender systems are naturally sparse (Flanagan et al., 2020; Harasic et al., 2024), and patient records across distributed healthcare networks frequently contain missing entries (Brisimi et al., 2018; Xu et al., 2021). Consequently, developing methods to accurately recover a complete data matrix from sparse, decentralized observations is a task of paramount importance (Li et al., 2025). It not only enables subsequent analysis but also ensures the reliability of models trained in a federated, privacy-preserving context.

This paper addresses the problem of federated low-rank matrix completion. In this setting, a ground-truth matrix $\boldsymbol{Y} \in \mathbb{R}^{m \times n}$ is assumed to be low-rank and is partitioned horizontally across p clients. Each client i holds its local submatrix $\boldsymbol{Y}_i \in \mathbb{R}^{m_i \times n}$ but observes only a sparse subset of its entries, indexed by the set $\boldsymbol{\Omega}_i$. Let $\mathcal{P}_{\boldsymbol{\Omega}_i}$ denote the projection operator that preserves entries with indices in $\boldsymbol{\Omega}_i$ and sets others to zero. The goal is to recover a matrix $\boldsymbol{X}_i \in \mathbb{R}^{m_i \times n}$ that approximates $\boldsymbol{Y}_i$ by fitting the observed data across all clients. Ideally, this can be formulated as the following global rank minimization problem:

$$\min_{\boldsymbol{X} \in \mathbb{R}^{m \times n}} \operatorname{rank}(\boldsymbol{X}) \quad \text{s.t.} \quad \mathcal{P}_{\boldsymbol{\Omega}_i}(\boldsymbol{X}_i) = \mathcal{P}_{\boldsymbol{\Omega}_i}(\boldsymbol{Y}_i), \quad \forall i \in \{1, \ldots, p\}, \tag{1}$$

where $\boldsymbol{X}_i$ denotes the submatrix of $\boldsymbol{X}$ corresponding to the rows associated with client i. The critical challenge, however, arises from the federated learning setting, which mandates that each client's raw data $\boldsymbol{Y}_i$ must remain localized and cannot be shared with a central server or other clients.

Existing federated matrix completion methods include matrix factorization approaches (Cherapanamjeri et al., 2017; Li et al., 2021; Nayer & Vaswani, 2022; Dalleiger & Gionis, 2025). These methods decompose a target matrix and iteratively update low-rank factors, a scheme suited for communication efficiency in a federated setting. While this scheme is communication-efficient, its performance is critically dependent on a pre-specified matrix rank. This parameter is not only difficult to estimate in practice, but its underlying assumption is also problematic. Real-world matrices are often not strictly low-rank; instead, they exhibit a spectrum of singular values that decays gradually. For such matrices, any fixed-rank approximation is inherently inaccurate and can lead to either underfitting or overfitting (Montesinos López et al., 2022; Fogel et al., 2023). The disadvantage of matrix factorization methods, therefore, lies not only in the challenge of rank estimation but also in their inapplicability when the matrix is not strictly low-rank. Alternatively, methods based on nuclear norm minimization provide a

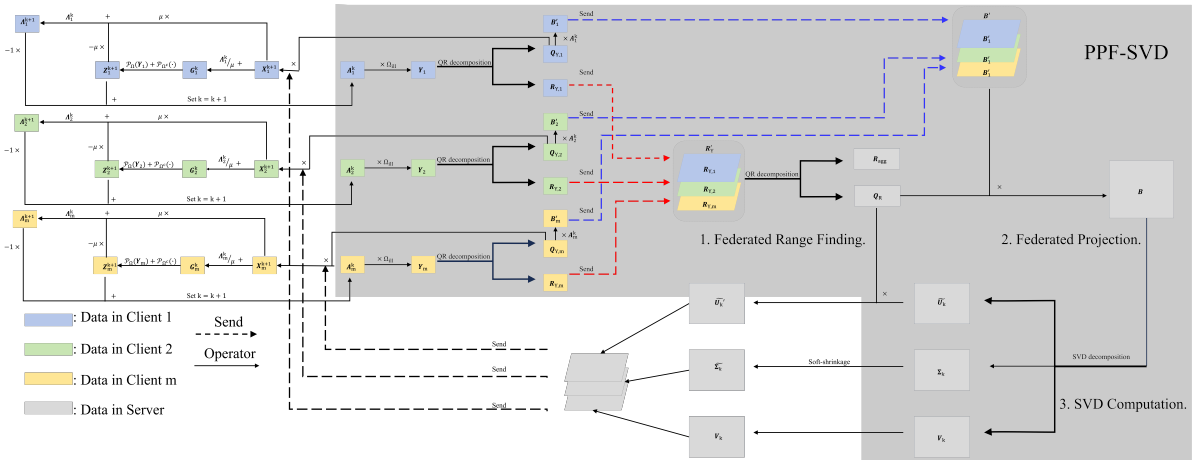


Figure 1: A schematic of a single ADMM iteration in the FedWNNM framework. The iteration begins with clients and the server collaboratively computing a global low-rank matrix estimate using our privacy-preserving SVD. This global estimate is then broadcast back to clients, where each performs local updates to enforce its data observations and adjust its dual variable, preparing for the next cycle.

convex surrogate for an explicit rank constraint (Candes & Recht, 2012; Gu et al., 2017). These methods employ a convex surrogate for the rank constraint and can effectively handle matrices with many small singular values. The core computational step in these approaches is the proximal operator, which necessitates a Singular Value Decomposition (SVD) of the global data matrix. As SVD is a centralized operation, its use directly conflicts with the data localization constraints of federated learning. This conflict has posed a fundamental barrier to deploying these robust models in a federated setting.

To resolve this dilemma, we propose the **Federated Weighted Nuclear Norm Minimization (FedWNNM)** framework, an algorithm to deploy the WNNM model in a privacy-preserving, federated setting. FedWNNM is based on the Alternating Direction Method of Multipliers (ADMM), a solver for such optimization problems. The core of this framework is a component we designed: the **Privacy-Preserving Federated SVD (PPF-SVD)**. This algorithm approximates the required SVD within each ADMM iteration without centralizing client data. This approach overcomes a key barrier to using WNNM in a federated setting. By integrating PPF-SVD, our FedWNNM framework applies matrix completion without a predefined rank under the constraints of federated setting. See Figure 1.

We propose FedWNNM, a framework for federated matrix completion based on our PPF-SVD algorithm. Unlike existing federated matrix factorization methods, FedWNNM does not require a pre-specified matrix rank. This feature enhances its practicality and robustness. The PPF-SVD algorithm is also distinct from previous federated SVD methods. While other methods incur high communication overhead (Dalleiger & Gionis, 2025) or require a trusted third party (Chai et al., 2022), PPF-SVD is designed with a theoretical error bound and offers a tunable trade-off between accuracy and privacy.

The FedWNNM framework provides control over the accuracy-privacy trade-off through a tunable decay factor, ρ, within its PPF-SVD component. To evaluate the framework, we conduct experiments on synthetic and real-world datasets. The results show that: (i) FedWNNM achieves higher recovery accuracy compared to existing federated factorization baselines, reducing recovery error by up to 96%; (ii) Experiments on ρ demonstrate that this parameter balances model fidelity with data privacy.

Contributions. The main contributions of this paper are summarized as follows:

- We propose **FedWNNM**, a federated framework for matrix completion based on WNNM. Motivated by the observation that real-world data are approximately, not strictly, low-rank, our framework does not require pre-specification of the matrix rank. This overcomes a primary limitation of matrix factorization methods, whose efficacy is diminished by the imposition of a hard rank constraint on such data.
- We develop **PPF-SVD**, a privacy-preserving algorithm for approximating the Singular Value Decomposition. This algorithm enables the computation of the WNNM proximal operator within FedWNNM without exposing client data. This design allows for accurate matrix recovery while ensuring client privacy, a feature not present in centralized WNNM solvers.
- We establish a theoretical error bound for the PPF-SVD algorithm. This analysis provides a controllable trade-off between model accuracy and data privacy.
- We demonstrate through experiments that FedWNNM achieves higher recovery accuracy and efficiency compared to existing federated matrix completion methods.

2 Related Works

Our work relates to federated matrix completion and federated optimization.

Federated Matrix Completion: Most approaches to federated matrix completion are based on matrix factorization (Cherapanamjeri et al., 2017; Li et al., 2021; Nayer & Vaswani, 2022; Dalleiger & Gionis, 2025). These methods decompose the global matrix $\boldsymbol{X}$ into two low-rank factors, $\boldsymbol{U}$ and $\boldsymbol{V}$, which are learned through an iterative client-server communication scheme. This design is well-suited for the federated setting, as sharing only the factor matrices provides data privacy and communication efficiency (Jain et al., 2013; Yi et al., 2016; Abbasi & Vaswani, 2025). Building on this principle, methods from alternating minimization (Jain et al., 2013) to gradient-based approaches (Abbasi & Vaswani, 2025) have been adapted for the federated setting. However, this family of algorithms shares a limitation: the requirement of a pre-specified matrix rank r, which is often impractical. In applications where the true rank is unknown, an incorrect choice can lead to performance degradation through underfitting or overfitting (Montesinos López et al., 2022; Fogel et al., 2023). FedWNNM addresses this limitation by using a framework based on a non-convex relaxation of the rank function.

Nuclear Norm Minimization: In centralized settings, a standard approach to matrix completion involves replacing the intractable rank function with convex surrogates like the nuclear norm (Candes & Recht, 2012; Sun & Luo, 2016) or the weighted nuclear norm (WNNM) (Gu et al., 2017). The success of these models relies on solving a proximal operator via SVD (Cai et al., 2010; Hsieh & Olsen, 2014). This reliance on a global SVD has rendered this class of models inaccessible to federated learning. Our FedWNNM framework addresses this challenge by providing a privacy-preserving mechanism to compute the required updates.

Federated SVD: The challenge of computing SVD on decentralized data is a known problem. Proposed solutions include hierarchical merging (Cherapanamjeri et al., 2017), which can require high communication bandwidth, and iterative methods like subspace iteration (Dalleiger & Gionis, 2025). Another direction involves cryptography, such as FedSVD's use of matrix masking, but this requires a trusted third-party, which is an often impractical assumption (Chai et al., 2022). In contrast, PPF-SVD is a specialized algorithm designed for the federated WNNM problem, not for general federated SVD. Our method provides a balance between model accuracy and data privacy, supported by theoretical analysis.

3 Proposed Method

In this section, we develop Federated WNNM (FedWNNM), a novel algorithm for applying Weighted Nuclear Norm Minimization (WNNM) in a federated setting. WNNM is a non-convex approach for matrix completion, however, its federated application is hindered by a core optimization step that requires a global SVD. To overcome this barrier, we first formulate the federated WNNM problem within the ADMM framework. This formulation reveals that a key subproblem is the need for a distributed and privacy-preserving SVD computation. We then introduce the Privacy-Preserving Federated SVD (PPF-SVD) algorithm as a subroutine specifically designed to solve this subproblem. By integrating PPF-SVD into the ADMM framework, FedWNNM solves the matrix completion problem across distributed clients and provides quantifiable privacy guarantees without centralizing data.

3.1 WNNM for Matrix Completion and its ADMM Framework

The WNNM model (Gu et al., 2017) formulates the matrix completion problem (1) as

$$\min_{\boldsymbol{X}} \|\boldsymbol{X}\|_{w,*} \quad \text{s.t.} \quad \mathcal{P}_{\boldsymbol{\Omega}_i}(\boldsymbol{X}_i) = \mathcal{P}_{\boldsymbol{\Omega}_i}(\boldsymbol{Y}_i), \quad \forall i = 1, \ldots, p, \tag{2}$$

where $\boldsymbol{X}_i$ is the i-th row block of the global matrix $\boldsymbol{X}$, and $\|\boldsymbol{X}\|_{w,*} = \sum_{j=1}^{\min(m,n)} w_j \sigma_j(\boldsymbol{X})$.

We solve Eq. (2) using ADMM (Boyd et al., 2011). The method utilizes an auxiliary variable $\boldsymbol{Z}$, a dual variable $\boldsymbol{\Lambda}$, and a penalty parameter $\mu > 0$. The iterative updates at step k are:

$$\begin{aligned}
\boldsymbol{X}^{k+1} &= \arg\min_{\boldsymbol{X}} \left(\|\boldsymbol{X}\|_{w,*} + \frac{\mu}{2} \|\boldsymbol{X} - (\boldsymbol{Z}^k - \boldsymbol{\Lambda}^k/\mu)\|_F^2 \right), \\
\boldsymbol{Z}_i^{k+1} &= \mathcal{P}_{\boldsymbol{\Omega}_i}(\boldsymbol{Y}_i) + \mathcal{P}_{\boldsymbol{\Omega}_i^c}(\boldsymbol{X}_i^{k+1} + \frac{\boldsymbol{\Lambda}_i^k}{\mu}), \\
\boldsymbol{\Lambda}_i^{k+1} &= \boldsymbol{\Lambda}_i^k + \mu(\boldsymbol{X}_i^{k+1} - \boldsymbol{Z}_i^{k+1}).
\end{aligned}$$

A comprehensive derivation of these update steps is provided in Appendix B. Critically, this ADMM framework decomposes the problem into local and global subproblems. The updates for variables $\boldsymbol{Z}$ and $\boldsymbol{\Lambda}$ are executed locally on each client. The global $\boldsymbol{X}$-update, however, which involves all clients, presents the primary computational and privacy bottleneck.

3.2 PPF-SVD: A PRIVACY-PRESERVING ALGORITHM FOR THE $\boldsymbol{X}$-UPDATE

The $\boldsymbol{X}$-update step requires the computation of the Singular Value Decomposition (SVD) of a matrix $\boldsymbol{A}^k \in \mathbb{R}^{m\times n}$, defined as $\boldsymbol{A}^k = \boldsymbol{Z}^k - \boldsymbol{\Lambda}^k/\mu$. In our federated framework, $\boldsymbol{A}^k$ is constituted by the vertical concatenation of local data matrices $\boldsymbol{A}_i^k \in \mathbb{R}^{m_i\times n}$ from p distinct clients, such that $\boldsymbol{A}^k = [(\boldsymbol{A}_1^k)^T, \ldots, (\boldsymbol{A}_p^k)^T]^T$. A direct computation of the SVD would mandate the aggregation of all local matrices $\{\boldsymbol{A}_i^k\}$ at a central server, a procedure that is explicitly forbidden by the privacy constraints inherent to federated learning.

To surmount this obstacle, we propose a method founded upon the principles of randomized SVD (Halko et al., 2011). The central strategy is to construct a low-dimensional orthonormal matrix $\boldsymbol{Q} \in \mathbb{R}^{m\times d}$, where the rank $d \ll \min(m, n)$, whose columns form a basis that approximates the range of $\boldsymbol{A}$. Such a basis facilitates the low-rank approximation $\boldsymbol{A} \approx \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}$. Consequently, the SVD of the large, distributed matrix $\boldsymbol{A}$ can be accurately and efficiently approximated by computing the SVD of a much smaller projected matrix, $\boldsymbol{B} = \boldsymbol{Q}^T\boldsymbol{A} \in \mathbb{R}^{d\times n}$. Our method proceeds in three distinct stages:

1. Federated Range Finding. Our initial objective is to construct the orthonormal basis $\boldsymbol{Q}$ in a federated manner. To achieve this, we employ a sampling technique. The server generates a random test matrix, $\boldsymbol{\Omega} \in \mathbb{R}^{n\times d}$, and distributes it to each client. The purpose of $\boldsymbol{\Omega}$ is to create a low-dimensional sketch of the column space of $\boldsymbol{A}$. Each client i computes its local sketch, $\boldsymbol{Y}_i$, as:

$$\boldsymbol{Y}_i = \boldsymbol{A}_i\boldsymbol{\Omega}.$$

The global sketch, $\boldsymbol{Y} = [\boldsymbol{Y}_1^T, \ldots, \boldsymbol{Y}_p^T]^T$, is the conceptual vertical concatenation of these local sketches, though it is never explicitly formed in a central location. To find an orthonormal basis for the range of $\boldsymbol{Y}$, we perform a distributed QR decomposition. Each client first computes the thin QR factorization of its local sketch:

$$\boldsymbol{Y}_i = \boldsymbol{Q}_{Y,i}\boldsymbol{R}_{Y,i}.$$

The matrix $\boldsymbol{Q}_{Y,i}$ contains an orthonormal basis for the local sketch, while the small matrix $\boldsymbol{R}_{Y,i} \in \mathbb{R}^{d\times d}$ is transmitted to the server. The server then aggregates these by forming $\boldsymbol{R}_Y = [\boldsymbol{R}_{Y,1}^T, \ldots, \boldsymbol{R}_{Y,p}^T]^T$ and computes its own thin QR factorization:

$$\boldsymbol{R}_Y = \boldsymbol{Q}_R\boldsymbol{R}_{\text{agg}}.$$

This second QR factorization serves to combine the local orthonormal bases into a single, global basis. The final orthonormal basis for the range of $\boldsymbol{A}$ is given by $\boldsymbol{Q} = \text{diag}(\boldsymbol{Q}_{Y,1}, \ldots, \boldsymbol{Q}_{Y,p})\boldsymbol{Q}_R$. For practical computation, we consider $\boldsymbol{Q}$ partitioned into row blocks $\boldsymbol{Q} = [\boldsymbol{Q}_1^T, \ldots, \boldsymbol{Q}_p^T]^T$, where $\boldsymbol{Q}_i \in \mathbb{R}^{m_i\times d}$. The i-th block, which is the only part of $\boldsymbol{Q}$ that client i needs, is computed locally as:

$$\boldsymbol{Q}_i = \boldsymbol{Q}_{Y,i}(\boldsymbol{Q}_R)_i,$$

where $(\boldsymbol{Q}_R)_i$ is the corresponding i-th block of rows of $\boldsymbol{Q}_R$ sent from the server.

2. Federated Projection. Having constructed the basis $\boldsymbol{Q}$, we now project the original matrix $\boldsymbol{A}$ onto the subspace spanned by $\boldsymbol{Q}$ to obtain the smaller matrix $\boldsymbol{B} = \boldsymbol{Q}^T\boldsymbol{A}$. The federated structure of our problem allows this projection to be decomposed into a sum of local contributions, thereby avoiding data centralization:

$$\boldsymbol{B} = \sum_{i=1}^{p} \boldsymbol{Q}_i^T\boldsymbol{A}_i = \sum_{i=1}^{p} (\boldsymbol{Q}_{Y,i}(\boldsymbol{Q}_R)_i)^T\boldsymbol{A}_i = \sum_{i=1}^{p} (\boldsymbol{Q}_R)_i^T(\boldsymbol{Q}_{Y,i}^T\boldsymbol{A}_i).$$

This formulation is key to the federated computation. Each client computes a local partial projection, $\boldsymbol{B}_i' = \boldsymbol{Q}_{Y,i}^T\boldsymbol{A}_i$, and transmits this $d \times n$ matrix to the server. The server then computes the final projected matrix $\boldsymbol{B}$ by applying the appropriate weights from $\boldsymbol{Q}_R$ and summing the results. This step effectively reduces the dimensionality of the problem from $m \times n$ to $d \times n$ while preserving the essential spectral information of $\boldsymbol{A}$.

3. SVD Computation and Reconstruction. The server now holds the small matrix $\boldsymbol{B}$, which serves as a compressed representation of $\boldsymbol{A}$. The server computes the SVD of $\boldsymbol{B}$ directly: $\boldsymbol{B} \approx \tilde{\boldsymbol{U}}_k\hat{\boldsymbol{\Sigma}}_k\boldsymbol{V}_k^T$. The resulting singular values $\hat{\boldsymbol{\Sigma}}_k$ and right singular vectors $\boldsymbol{V}_k$ are direct approximations of those of $\boldsymbol{A}$. This follows from the approximation $\boldsymbol{A} \approx \boldsymbol{Q}\boldsymbol{B} = \boldsymbol{Q}(\tilde{\boldsymbol{U}}_k\hat{\boldsymbol{\Sigma}}_k\boldsymbol{V}_k^T) = (\boldsymbol{Q}\tilde{\boldsymbol{U}}_k)\hat{\boldsymbol{\Sigma}}_k\boldsymbol{V}_k^T$. From this, we identify the approximate left singular vectors of $\boldsymbol{A}$ as $\boldsymbol{U} \approx \boldsymbol{Q}\tilde{\boldsymbol{U}}_k$. To maintain data privacy, these are reconstructed locally at each client. The server disseminates $\tilde{\boldsymbol{U}}_k$ to all clients. Client i then computes its local partition of the left singular vectors using its local information and the relevant block from the global basis:

$$\boldsymbol{U}_i \approx \boldsymbol{Q}_i\tilde{\boldsymbol{U}}_k = \boldsymbol{Q}_{Y,i}(\boldsymbol{Q}_R)_i\tilde{\boldsymbol{U}}_k.$$

This procedure, which we designate as Privacy-Preserving Federated SVD (PPF-SVD), is delineated in Algorithm 1.

With the establishment of the PPF-SVD subroutine to address the global $\boldsymbol{X}$-update, we now introduce the complete FedWNNM algorithm. This algorithm is iterative, alternating between the federated $\boldsymbol{X}$-update, executed via PPF-SVD, and the local $\boldsymbol{Z}$- and $\boldsymbol{\Lambda}$-updates, which are performed independently by each client.

Algorithm 1 Privacy-Preserving Federated SVD (PPF-SVD)

Input: Client data $\{\boldsymbol{A}_i\}_{i=1}^p$, target rank k, oversampling parameter p_{over}, privacy-tuning decay factor $\rho \in (0,1)$.

Output: Server obtains $\hat{\boldsymbol{\Sigma}}, \boldsymbol{V}$; client i obtains $(\boldsymbol{U}_i)$.

// Server: Initialization

Set $d = k + p_{\text{over}}$.
Generate random orthogonal $\boldsymbol{U}_\Omega \in \mathbb{R}^{n\times n}$, $\boldsymbol{V}_\Omega \in \mathbb{R}^{d\times d}$.
Form $\boldsymbol{D}_\Omega = \text{diag}(1, \rho, \rho^2, \ldots, \rho^{d-1})$.
Compute $\boldsymbol{\Omega}_{\text{ill}} = \boldsymbol{U}_\Omega(:, 1:d)\boldsymbol{D}_\Omega \boldsymbol{V}_\Omega^T$.
Send $\boldsymbol{\Omega}$ to all clients.

// Clients: Federated Range Finding (Part 1)

for all client $i = 1$ to p (parallel) **do**
  Compute $\boldsymbol{Y}_i = \boldsymbol{A}_i\boldsymbol{\Omega}$.
  Decompose $\boldsymbol{Y}_i = \boldsymbol{Q}_{Y,i}\boldsymbol{R}_{Y,i}$.
  Send $\boldsymbol{R}_{Y,i}$ to server.
end for

// Server: Federated Range Finding (Part 2)

Aggregate $\boldsymbol{R}'_Y = [\boldsymbol{R}_{Y,1}^T, \ldots, \boldsymbol{R}_{Y,p}^T]^T$.
Decompose $\boldsymbol{R}'_Y = \boldsymbol{Q}_R\boldsymbol{R}_{\text{agg}}$.

// Clients: Federated Projection (Part 1)

for all client $i = 1$ to p (parallel) **do**
  Compute $\boldsymbol{B}'_i = \boldsymbol{Q}_{Y,i}^T\boldsymbol{A}_i$.
  Send $\boldsymbol{B}'_i$ to server.
end for

// Server: Federated Projection (Part 2)

Partition $\boldsymbol{Q}_R$ into blocks $\{(\boldsymbol{Q}_R)_i\}_{i=1}^p$.
Compute $\boldsymbol{B} = \sum_{i=1}^p (\boldsymbol{Q}_R)_i^T \boldsymbol{B}'_i$.
SVD of $\boldsymbol{B} \approx \tilde{\boldsymbol{U}}_k\hat{\boldsymbol{\Sigma}}_k\boldsymbol{V}_k^T$.

// Clients: Left Singular Vector Reconstruction

for all client $i = 1$ to p (parallel) **do**
  Server sends $\tilde{\boldsymbol{U}}$ and $(\boldsymbol{Q}_R)_i$ to client i.
  Compute $(\boldsymbol{U}_i) = \boldsymbol{Q}_{Y,i}(\boldsymbol{Q}_R)_i\tilde{\boldsymbol{U}}$.
end for

Algorithm 2 Federated WNNM for Matrix Completion (Fed-WNNM)

Input: Client data $\{(\boldsymbol{Y}_i, \Omega_i)\}_{i=1}^p$; ADMM penalty $\mu > 0$; penalty growth factor $\tau > 1$; tolerance $\theta > 0$; initial weights $\boldsymbol{w}$ *or* reweighting parameters (c, ϵ); PPF-SVD settings $(k, p_{\text{over}}, \rho)$ with $d = k + p_{\text{over}}$.

Output: Distributed estimate $\boldsymbol{X} = [\boldsymbol{X}_1^\top, \ldots, \boldsymbol{X}_p^\top]^\top$; (server obtains no raw data).

// Server: Initialization

Fix tolerance θ, penalty $\mu > 0$, growth factor $\tau > 1$, and PPF-SVD parameters $(k, p_{\text{over}}, \rho)$; set $d = k + p_{\text{over}}$.

// Clients: Initialization

Set $\boldsymbol{X}_i^0 = 0$, $\boldsymbol{Z}_i^0 = 0$, $\boldsymbol{\Lambda}_i^0 = 0$.
$iter \leftarrow 0$.
while not converged **do**

// Clients: Prepare for SVD

  for all client $i = 1$ to p (parallel) **do**
    Form proximal input $\boldsymbol{A}_i^{iter} = \boldsymbol{Z}_i^{iter} - \boldsymbol{\Lambda}_i^{iter}/\mu$.
  end for

// Server: Federated SVD and Weighting

  Invoke **PPF-SVD (Alg. 1)** on $\boldsymbol{A}^{iter}$ to get $\hat{\boldsymbol{\Sigma}}^{iter}, \boldsymbol{V}^{iter}$ and clients get $\boldsymbol{U}_i^{iter}$.
  Update weights $w_j^{iter+1} = c/(\hat{\sigma}_j^{iter} + \epsilon)$.
  Shrink singular values $\tilde{\sigma}_j^{iter+1} = \max(\hat{\sigma}_j^{iter} - w_j^{iter+1}/\mu,\, 0)$.
  Broadcast $\tilde{\boldsymbol{\Sigma}}^{iter+1} = \text{diag}(\tilde{\sigma}_j^{iter+1})$ and $\boldsymbol{V}^{iter}$.

// Clients: X-Update

  for all client $i = 1$ to p (parallel) **do**
    $\boldsymbol{X}_i^{iter+1} = \boldsymbol{U}_i^{iter}\,\tilde{\boldsymbol{\Sigma}}^{iter+1}\,(\boldsymbol{V}^{iter})^\top$.
  end for

// Clients: Z-Update

  for all client $i = 1$ to p (parallel) **do**
    $\boldsymbol{Z}_i^{iter+1} = \mathcal{P}_{\Omega_i}(\boldsymbol{Y}_i) + \mathcal{P}_{\Omega_i^c}(\boldsymbol{X}_i^{iter+1} + \frac{\boldsymbol{\Lambda}_i^{iter}}{\mu})$.
  end for

// Clients: Dual Ascent

  for all client $i = 1$ to p (parallel) **do**
    $\boldsymbol{\Lambda}_i^{iter+1} = \boldsymbol{\Lambda}_i^{iter} + \mu\big(\boldsymbol{X}_i^{iter+1} - \boldsymbol{Z}_i^{iter+1}\big)$.
    Compute and send residuals $r_i^2 = \|\boldsymbol{X}_i^{iter+1} - \boldsymbol{Z}_i^{iter+1}\|_F^2, s_i^2 = \|\boldsymbol{Y}_i\|_F^2$ to server.
  end for

// Server: Convergence Check

  Aggregate residuals $r^2 = \sum_{i=1}^p r_i^2$, $s^2 = \sum_{i=1}^p s_i^2$.
  if $\sqrt{r^2}/\sqrt{s^2} \le \theta$ **then**
    stop.
  end if
  Update penalty $\mu \leftarrow \tau\mu$.
  $iter \leftarrow iter + 1$.
end while
Return: distributed solution $\{\boldsymbol{X}_i\}_{i=1}^p$.

Algorithm 2 specifies the complete FedWNNM procedure, and Figure 1 illustrates the corresponding data flow. The practical viability of a federated algorithm is determined by its communication overhead. We therefore quantify the cost per ADMM iteration as follows:

$$C_{\text{total}} = p\big[2(k + p_{\text{over}})n + 2(k + p_{\text{over}})^2 + (n + k + p_{\text{over}} + 1)k + 2\big].$$

This cost is dominated by terms that scale linearly with the number of clients p and the feature dimension n. A full derivation is provided in Appendix D.

While the communication cost governs the practical feasibility of FedWNNM, its numerical accuracy depends on the approximation error from the PPF-SVD subroutine. A theoretical analysis of this error is necessary to establish performance guarantees for the algorithm. Our analysis begins with a general theorem that bounds the error of the federated range finding procedure. We introduce the following notation for the subsequent results.

Let the SVD of $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ be partitioned as $\boldsymbol{A} = \boldsymbol{U}_1\boldsymbol{\Sigma}_1\boldsymbol{V}_1^T + \boldsymbol{U}_2\boldsymbol{\Sigma}_2\boldsymbol{V}_2^T$, where $\boldsymbol{\Sigma}_1$ is a diagonal matrix containing the k largest singular values. The columns of $\boldsymbol{U}_1$ and $\boldsymbol{V}_1$ form orthonormal bases for the corresponding k-dimensional left and right singular subspaces, respectively. The remaining singular components comprise the term $\boldsymbol{U}_2\boldsymbol{\Sigma}_2\boldsymbol{V}_2^T$. Let $\boldsymbol{\Omega} \in \mathbb{R}^{n\times d}$, where $d \geq k$, be a random test matrix. Let $\boldsymbol{Q}$ be an orthonormal basis for the range of $\boldsymbol{A}\boldsymbol{\Omega}$, constructed according to Algorithm 1 as $\boldsymbol{Q} = \mathrm{diag}(\boldsymbol{Q}_{Y,1}, \ldots, \boldsymbol{Q}_{Y,p})\boldsymbol{Q}_R$.

Theorem 3.1. *Assume that $\boldsymbol{V}_1^T\boldsymbol{\Omega}$ has full row rank. The spectral norm of the approximation error satisfies*

$$\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}\|_2^2 \leq \|\boldsymbol{\Sigma}_2\|_2^2 + \|\boldsymbol{\Sigma}_2(\boldsymbol{V}_2^T\boldsymbol{\Omega})(\boldsymbol{V}_1^T\boldsymbol{\Omega})^\dagger\|_2^2, \tag{3}$$

where $(\cdot)^\dagger$ denotes the Moore-Penrose pseudoinverse.

Proof. The proof is provided in Appendix C.1. □

Theorem 3.1 establishes a general upper bound on the approximation error. However, this bound does not make explicit the dependence of the approximation accuracy on the choice of the test matrix $\boldsymbol{\Omega}$ in Algorithm 1. To analyze this dependence, we specialize the bound to a structured class of test matrices. Specifically, the following corollary considers the case where $\boldsymbol{\Omega}$ is constructed with singular values that exhibit a prescribed decay. The approximation quality is quantified in terms of the principal angles (see Definition C.1) between the target subspace $\mathrm{range}(\boldsymbol{V}_1)$ and the subspace spanned by the columns of $\boldsymbol{\Omega}$.

Corollary 3.2. *Consider the test matrix $\boldsymbol{\Omega} \in \mathbb{R}^{n\times d}$ from Algorithm 1, defined by $\boldsymbol{\Omega} = \boldsymbol{U}_\Omega\boldsymbol{D}_\Omega\boldsymbol{V}_\Omega^T$, where the columns of $\boldsymbol{U}_\Omega \in \mathbb{R}^{n\times d}$ are orthonormal, $\boldsymbol{V}_\Omega \in \mathbb{R}^{d\times d}$ is orthogonal, and $\boldsymbol{D}_\Omega = \mathrm{diag}(1, \rho, \ldots, \rho^{d-1})$ for a factor $\rho \in (0, 1)$. If $\boldsymbol{V}_1^T\boldsymbol{U}_\Omega$ has full row rank, the spectral norm of the approximation error satisfies the bound*

$$\sigma_{d+1}(\boldsymbol{A}) \leq \|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}\|_2 \leq \sigma_{k+1}(\boldsymbol{A})\sqrt{1 + \frac{\kappa(\boldsymbol{D}_\Omega)^2}{\cos^2\theta_k}} = \sigma_{k+1}(\boldsymbol{A})\sqrt{1 + \rho^{-2(d-1)}\sec^2\theta_k},$$

where $\sigma_{k+1}(\boldsymbol{A})$ is the $(k+1)$-th singular value of $\boldsymbol{A}$, and θ_k is the k-th principal angle between the subspaces $\mathrm{range}(\boldsymbol{V}_1)$ and $\mathrm{range}(\boldsymbol{U}_\Omega)$ (see Definition C.1).

Proof. The proof is provided in Appendix C.2. □

Remark 3.3. Corollary 3.2 delineates a performance interval for the approximation error. The lower bound, $\sigma_{d+1}(\boldsymbol{A})$, establishes the theoretical floor for the error. This inherent limitation is dictated by the spectral properties of $\boldsymbol{A}$ and the dimensionality d of the subspace used for the approximation. The upper bound, in contrast, quantifies how this baseline error is amplified by the specific construction of the test matrix $\boldsymbol{\Omega}$. This amplification depends on two main factors: the geometric alignment between the target subspace $\mathrm{range}(\boldsymbol{V}_1)$ and the sampling subspace $\mathrm{range}(\boldsymbol{U}_\Omega)$, captured by $\sec^2\theta_k$, and the condition number of the test matrix, $\kappa(\boldsymbol{D}_\Omega) = \rho^{-(d-1)}$. Since $\mathrm{range}(\boldsymbol{U}_\Omega)$ is chosen randomly, orthogonality to the fixed subspace $\mathrm{range}(\boldsymbol{V}_1)$ is improbable. The dominant term governing the error amplification is therefore the condition number, which is explicitly controlled by the decay factor ρ. This parameter thus provides a direct mechanism to manage the trade-off between approximation accuracy and other algorithmic considerations.

Having established the theoretical error bounds for the core PPF-SVD subroutine, we next evaluate the empirical performance of the complete FedWNNM algorithm.

4 Numerical Experiments

In this section, we present a comprehensive empirical evaluation of the proposed FedWNNM algorithm. Our investigation is designed to assess its performance across three critical dimensions: recovery accuracy in diverse settings, the efficacy of the embedded privacy mechanism, and algorithmic robustness. We begin by demonstrating the algorithm's superior performance on challenging recovery tasks, including phase transition analysis and image inpainting. Subsequently, we analyze the privacy-preserving SVD subroutine to verify its control over the privacy-utility trade-off. Finally, we provide detailed benchmarks on synthetic data to quantify convergence rates and examine the algorithm's sensitivity to hyperparameter variations. All experiments were conducted in MATLAB (R2023a) on a workstation with an Intel Xeon E5-2698 v4 CPU and 32 GB of RAM. The per-iteration computational complexity of FedWNNM is $\mathcal{O}(m_i nd + pd^2 + d^2 n)$, dominated by distributed QR factorizations on clients and a low-dimensional SVD on the server.

The algorithm is benchmarked against centralized and federated methods. Centralized baselines include Weighted Nuclear Norm Minimization (WNNM) (Gu et al., 2017), Alternating Gradient Descent (AltGD) (Nayer & Vaswani, 2022), and centralized alternating minimization variants (altGDMinCntrl, altMinCntrl) (Jain et al., 2013). Federated baselines are Federated Alternating Gradient Descent Minimization (altGDMin) (Abbasi & Vaswani, 2025) and Private Alternating Minimization (altMinPrvt) (Jain et al., 2013). Performance is quantified using Relative Error for synthetic data, and Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index

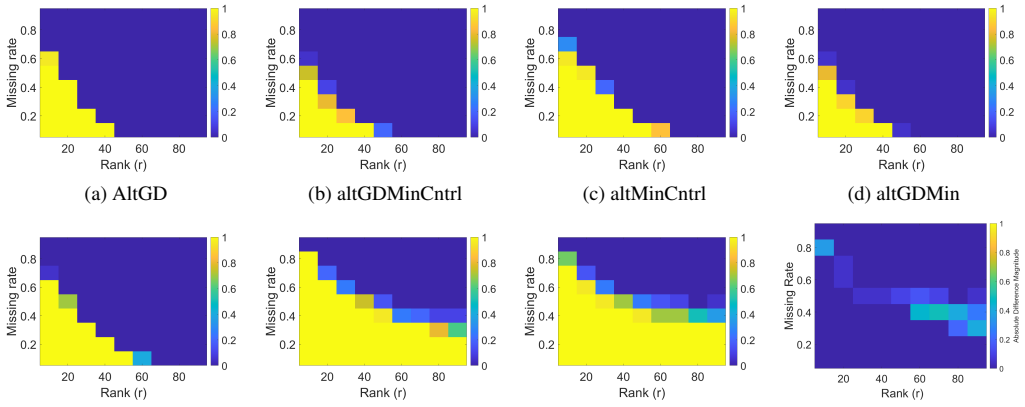
Measure (SSIM) for image data. Unless specified otherwise, hyperparameters for our model were set as follows: weight $C = 5.252\sqrt{mn}$ with $\epsilon = 10^{-6}$, PPF-SVD decay factor $\rho = 0.1$, oversampling parameter $p_{\text{over}} = 10$ and an initial ADMM penalty $\mu = 1$ with an adaptive factor $\tau = 1.05$. Iterations terminate when the relative residual is below 10^{-5} or after 350 iterations.

4.1 Performance of the FedWNNM Algorithm

We commence our numerical study by evaluating the overall recovery capabilities of the FedWNNM algorithm. This subsection establishes the algorithm's efficacy on both structural recovery limits and practical application, benchmarking it against state-of-the-art centralized and federated alternatives.

4.1.1 Phase Transition Analysis

To evaluate empirical recovery performance, a phase transition analysis is performed on 100×100 matrices by varying the rank r and the missing rate. A recovery is considered successful if the relative error is below 0.17. Figure 2 shows the success probability maps, averaged over 20 trials. The centralized WNNM (Figure 2f) and the proposed FedWNNM (Figure 2g) achieve successful recovery over a larger region of the parameter space compared to other federated methods. This performance is attributed to the flexibility of the WNNM framework. Whereas methods based on standard matrix factorization impose a hard rank constraint, which may discard information contained in smaller singular values, the WNNM's weighted norm penalizes singular values differentially. This enables a more effective model for data matrices whose singular values decay gradually, resulting in improved recovery accuracy. Figure 2h presents the absolute difference in the success rate between centralized WNNM and FedWNNM. The prevalence of dark blue areas indicates a negligible performance discrepancy across the parameter space. This result demonstrates that FedWNNM preserves the recovery capabilities of the WNNM model within a federated framework, making it a viable solution for collaborative matrix completion under privacy constraints.



(a) AltGD (b) altGDMinCntrl (c) altMinCntrl (d) altGDMin

(e) altMinPrvt (f) WNNM (g) **FedWNNM**(Ours) (h) |WNNM − FedWNNM|

Figure 2: Phase transition diagrams. Axes denote matrix rank and missing rate. Yellow indicates a success probability of 1; dark blue indicates 0. Plot (h) shows the absolute difference in success rates between WNNM and FedWNNM.

4.1.2 Application to Image Inpainting

To assess performance on real-world data, we apply FedWNNM to an image inpainting task on the CBSD68 dataset, with missing rates from 10% to 80%. Table 1 shows that the WNNM-based methods outperform those based on standard matrix factorization. These results are consistent with the understanding that real-world image data contain a spectrum of small but non-zero singular values that encode high-frequency details. The WNNM framework preserves these smaller singular values, whereas conventional matrix factorization methods truncate them, resulting in lower reconstruction quality.

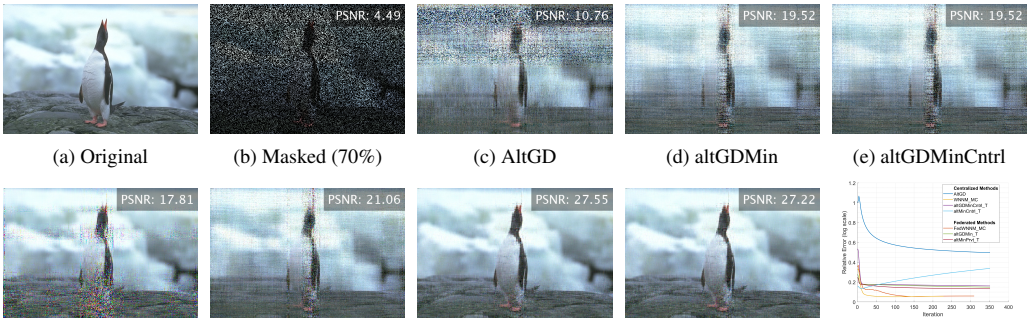
Among the federated algorithms, FedWNNM achieves the highest Peak Signal-to-Noise Ratio (PSNR), exceeding the second-best method (altMinPrvt) by 2.96 dB to 14.87 dB across the tested missing rates. Furthermore, the performance of FedWNNM is comparable to its centralized counterpart, WNNM, which serves as the non-private performance upper bound. The average PSNR difference is 0.19 dB, which demonstrates that the reconstruction fidelity of the WNNM model is maintained in a privacy-preserving framework.

A qualitative comparison is presented in Figure 3 for an image with 70% of pixels removed. The reconstruction from FedWNNM closely resembles that of the centralized WNNM, preserving texture and edge features. In contrast, other methods such as AltGD produce artifacts and oversmoothing, resulting in a loss of fine details. This

Table 1: Quantitative comparison for image inpainting on the CBSD68 dataset. We report average PSNR (dB) / SSIM values. For each missing rate (MR), the best result is in **bold** and the second best is underlined.

Algorithm		MR=10%	MR=20%	MR=30%	MR=40%	MR=50%	MR=60%	MR=70%	MR=80%
Centralized	AltGD	26.80/0.7061	25.32/0.6455	23.80/0.5851	21.92/0.5140	19.27/0.4259	15.82/0.3164	11.24/0.1861	9.30/0.0829
	altGDMinCntrl	25.02/0.6192	23.90/0.5652	23.08/0.5213	22.31/0.4746	21.36/0.4202	20.15/0.3486	17.92/0.2446	3.36/0.0660
	altMinCntrl	28.55/0.8408	24.70/0.8168	20.41/0.7791	18.22/0.7250	14.77/0.6417	12.75/0.5182	11.58/0.3396	4.65/0.0323
	WNNM	**37.32/0.9672**	**33.18/0.9228**	**30.47/0.8709**	**28.29/0.8142**	**26.59/0.7552**	**24.91/0.6845**	**23.31/0.6016**	**21.58/0.4983**
Federated	altGDMin	25.02/0.6192	23.90/0.5652	23.08/0.5213	22.31/0.4745	21.36/0.4201	20.15/0.3486	17.92/0.2446	5.36/0.0655
	altMinPrvt	26.78/0.7099	26.06/0.6768	25.39/0.6433	24.67/0.6050	23.60/0.5490	21.96/0.4598	18.44/0.3033	6.34/0.0801
	FedWNNM	37.01/0.9646	32.98/0.9193	30.29/0.8661	28.23/0.8121	26.56/0.7547	24.86/0.6861	23.18/0.5996	21.21/0.4855

visual evidence validates the quantitative results, demonstrating the algorithm's effectiveness in image restoration. The convergence plot in Figure 3j further confirms these findings, illustrating that although several baseline methods exhibit faster initial convergence, FedWNNM converges to a solution with a significantly lower relative error.



(a) Original (b) Masked (70%) (c) AltGD (d) altGDMin (e) altGDMinCntrl

(f) altMinCntrl (g) altMinPrvt (h) WNNM (i) **FedWNNM** (Ours) (j) Convergence

Figure 3: Qualitative comparison of image inpainting with a 70% pixel missing rate. The final subfigure shows the convergence curves for relative error.

4.2 Analysis of the Privacy-Preserving SVD Subroutine

The performance observed in the preceding subsection depends on the PPF-SVD subroutine. To assess this component's privacy trade-off, which is controlled by the privacy parameter ρ, we analyze the subroutine in isolation.

We construct a matrix $\boldsymbol{A} \in \mathbb{R}^{1000\times 500}$ with decaying singular values and partition it horizontally among $p = 5$ clients. The PPF-SVD algorithm is then applied with a target rank $k = 15$. Let $\tilde{\boldsymbol{A}}_k$ denote matrix reconstructed from the algorithm 1. The trade-off is quantified by the reconstruction error, which is measured using two relative error metrics: the spectral norm error ($\|\tilde{\boldsymbol{A}}_k - \boldsymbol{A}\|_2/\|\boldsymbol{A}\|_2$) and the Frobenius norm error ($\|\tilde{\boldsymbol{A}}_k - \boldsymbol{A}\|_F/\|\boldsymbol{A}\|_F$). The magnitude of this error serves as a metric for privacy. A larger error indicates a higher level of privacy protection, as it becomes more difficult to infer client data from the server.

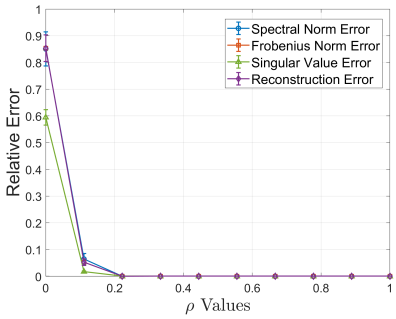
The numerical results are presented in Table 2. As ρ decreases from 0.5 to 0.001, the reconstruction error increases by more than eight orders of magnitude, indicating an enhancement in privacy. Concurrently, the spectral and Frobenius norm errors increase from approximately 10^{-8} to 0.9. These trends are illustrated in Figure 4a, where the curves plot the mean error over Monte Carlo runs. These results show that ρ serves as a control parameter for calibrating the privacy-utility trade-off.

Table 2: Quantitative analysis of the privacy-utility trade-off as a function of ρ. For the first three error metrics, lower values indicate higher utility. For the Reconstruction error (privacy proxy), a higher value is desirable.

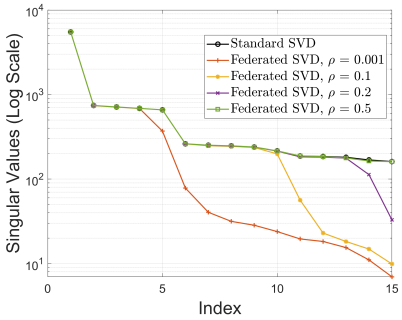
Parameter ρ	Relative Error Metrics (Mean)			
	Spectral Norm	Frobenius Norm	Singular Value	Reconstruction
0.001	8.842×10^{-1}	8.557×10^{-1}	6.187×10^{-1}	8.557×10^{-1}
0.1	7.341×10^{-2}	6.078×10^{-2}	1.931×10^{-2}	6.078×10^{-2}
0.2	2.713×10^{-3}	1.691×10^{-3}	3.440×10^{-5}	1.691×10^{-3}
0.5	3.910×10^{-8}	2.520×10^{-9}	7.230×10^{-13}	2.520×10^{-9}

Next, we compare the singular values computed by the PPF-SVD algorithm against those obtained from a centralized SVD, which serves as the ground truth. The experimental setup remains identical. Figure 4b plots the singular value spectra for various values of ρ on a semi-logarithmic scale. The plot shows that the algorithm captures the dominant singular values. For large ρ (e.g., $\rho = 0.5$), the computed spectrum approximates the ground

truth spectrum. As ρ decreases, the smaller singular values are estimated with a larger bias. Nevertheless, the spectral decay profile is preserved for all tested values of ρ. This result indicates that the algorithm retains the overall spectral structure of the data matrix, even in regimes corresponding to high privacy.



(a) Effect of ρ on relative error metrics.



(b) Singular value comparison for varying ρ.

Figure 4: Numerical analysis of the privacy parameter ρ. (a) The privacy-utility trade-off: as ρ decreases, the reconstruction error (privacy proxy) increases, while approximation errors (utility) also rise. (b) Comparison of singular values from our PPF-SVD against a centralized SVD (Ground Truth).

4.3 Numerical Validation on Synthetic Data

We conduct a quantitative comparison on a synthetic 1000×1000 matrix of rank 20, with 30% of its entries observed and distributed across 5 clients. To ensure statistical reliability, all reported results are averaged over 30 Monte Carlo trials.

The results are summarized in Table 3. The proposed FedWNNM algorithm outperforms other federated methods in both accuracy and computational efficiency. Its mean relative error is an order of magnitude lower than that of the second-best method, altMinPrvt, representing a 96.89% reduction in recovery error. Moreover, it requires significantly less computation time. Notably, the accuracy of FedWNNM also surpasses that of some centralized algorithms (e.g., AltGD). While the centralized WNNM algorithm serves as an ideal benchmark, our federated approach is competitive and demonstrates the ability to achieve high-fidelity matrix recovery without data centralization.

Table 3: Quantitative comparison of average relative error and total computation time (seconds) averaged over 30 Monte Carlo trials. For both metrics, lower values indicate superior performance. The **best** and second-best results among the federated methods are highlighted.

Algorithm		Average Relative Error ↓	Time (s) ↓
Centralized	AltGD	0.086183	0.7429
	altGDMinCntrl	0.143927	106.9561
	altMinCntrl	0.115372	0.6659
	WNNM	0.000020	0.4272
Federated	altGDMin	0.143996	124.0073
	altMinPrvt	0.088510	1186.3513
	FedWNNM	**0.002757**	54.8871

5 Conclusions and Limitations

In this work, we proposed FedWNNM, a framework that adapts the WNNM model to a federated learning setting while preserving data privacy. Our primary contribution is the Privacy-Preserving Federated SVD (PPF-SVD) subroutine. This method allows a central server to approximate the global SVD by aggregating low-dimensional sketches generated by clients via structured random projections. We derived a theoretical bound for the approximation error of PPF-SVD, which quantifies the trade-off between accuracy and a privacy parameter ρ. Experiments on synthetic data and image inpainting tasks demonstrate that FedWNNM achieves accuracy comparable to its centralized WNNM counterpart.

A limitation of our current work is that the error bound in Corollary 3.2 is deterministic for a given realization of the random projection matrix $\boldsymbol{U}_{\Omega,d}$, due to its dependence on the principal angle. A future theoretical direction is to derive an upper bound on the expectation of this error over the Haar measure on the Stiefel manifold. Obtaining a tractable analytical expression for this expectation is challenging due to the complexity of the Jacobian associated with the underlying eigenvalue distribution of the Jacobi ensemble.

Reproducibility Statement

To ensure the reproducibility of our research, we have provided a comprehensive supplementary material package. The source code for the proposed FedWNNM algorithm, all baseline methods, and the scripts to generate the figures and tables presented in this paper are included. The code is implemented in MATLAB. The supplementary material is organized to correspond with the experimental sections of the paper: phase transition analysis (Section 4.1.1), image inpainting on the CBSD68 dataset (Section 4.1.2), the analysis of the PPF-SVD subroutine (Section 4.2), and numerical validation on synthetic data (Section 4.3). Each experimental section in the supplement contains a `main.m` script, which allows for the direct reproduction of our results. Detailed instructions for setting up the environment and running the experiments are provided in the `README.md` file within the supplementary package. The theoretical proofs for our claims are available in Appendix C of the paper.

References

Ahmed Ali Abbasi and Namrata Vaswani. Efficient federated low rank matrix completion. *IEEE Transactions on Information Theory*, 2025.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Theodora S Brisimi, Ruidi Chen, Theofanie Mela, Alex Olshevsky, Ioannis Ch Paschalidis, and Wei Shi. Federated learning of predictive models from federated electronic health records. *International journal of medical informatics*, 112:59–67, 2018.

Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4):1956–1982, 2010.

Emmanuel Candes and Benjamin Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

Di Chai, Leye Wang, Junxue Zhang, Liu Yang, Shuowei Cai, Kai Chen, and Qiang Yang. Practical lossless federated singular vector decomposition over billion-scale data. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 46–55, 2022.

Yeshwanth Cherapanamjeri, Kartik Gupta, and Prateek Jain. Nearly optimal robust matrix completion. In *International Conference on Machine Learning*, pp. 797–805. PMLR, 2017.

Sebastian Dalleiger and Aristides Gionis. Creating coherence in federated non-negative matrix factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16135–16143, 2025.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3): 211–218, 1936.

Adrian Flanagan, Were Oyomno, Alexander Grigorievskiy, Kuan E Tan, Suleiman A Khan, and Muhammad Ammad-Ud-Din. Federated multi-view matrix factorization for personalized recommendations. In *Joint European conference on machine learning and knowledge discovery in databases*, pp. 324–347. Springer, 2020.

Paul Fogel, Christophe Geissler, Nicolas Morizet, and George Luta. On rank selection in non-negative matrix factorization using concordance. *Mathematics*, 11(22):4611, 2023.

Edoardo Gabrielli, Giovanni Pica, and Gabriele Tolomei. A survey on decentralized federated learning. *arXiv preprint arXiv:2308.04604*, 2023.

Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International journal of computer vision*, 121(2):183–208, 2017.

Nathan Halko, Per-Gunnar Martinsson, and Joel A Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM review*, 53(2):217–288, 2011.

Marko Harasic, Felix-Sebastian Keese, Denny Mattern, and Adrian Paschke. Recent advances and future challenges in federated recommender systems. *International Journal of Data Science and Analytics*, 17(4):337–357, 2024.

Cho-Jui Hsieh and Peder Olsen. Nuclear norm minimization via active subspace selection. In *International Conference on Machine Learning*, pp. 575–583. PMLR, 2014.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pp. 665–674, 2013.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

Andrew V Knyazev and Peizhen Zhu. Principal angles between subspaces and their tangents. *arXiv preprint arXiv:1209.0523*, 2012.

Siqi Li, Mengying Yan, Ruizhi Yuan, Molei Liu, Nan Liu, and Chuan Hong. Fedimpute: Privacy-preserving missing value imputation for multi-site heterogeneous electronic health records. *Journal of Biomedical Informatics*, 165:104780, 2025.

Zitao Li, Bolin Ding, Ce Zhang, Ninghui Li, and Jingren Zhou. Federated matrix factorization with privacy guarantee. *Proceedings of the VLDB Endowment*, 15(4), 2021.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pp. 1273–1282. PMLR, 2017.

Osval Antonio Montesinos López, Abelardo Montesinos López, and Jose Crossa. Overfitting, model tuning, and evaluation of prediction performance. In *Multivariate statistical machine learning methods for genomic prediction*, pp. 109–139. Springer, 2022.

Seyedehsara Nayer and Namrata Vaswani. Fast and sample-efficient federated low rank matrix recovery from column-wise linear and quadratic projections. *IEEE Transactions on Information Theory*, 69(2):1177–1202, 2022.

Sarwar Saif, Md Jahirul Islam, Md Zihad Bin Jahangir, Parag Biswas, Abdur Rashid, MD Nasim, and Kishor Datta Gupta. A comprehensive review on understanding the decentralized and collaborative approach in machine learning. *arXiv preprint arXiv:2503.09833*, 2025.

Ilias Siniosoglou, Stamatia Bibi, Konstantinos-Filippos Kollias, George Fragulis, Panagiotis Radoglou-Grammatikis, Thomas Lagkas, Vasileios Argyriou, Vasileios Vitsas, and Panagiotis Sarigiannidis. Federated learning models in decentralized critical infrastructure. *Shaping the Future of IoT with Edge Intelligence*, pp. 95–115, 2024.

Ruoyu Sun and Zhi-Quan Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Transactions on Information Theory*, 62(11):6535–6579, 2016.

Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of healthcare informatics research*, 5(1):1–19, 2021.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust pca via gradient descent. *Advances in neural information processing systems*, 29, 2016.

Betul Yurdem, Murat Kuzlu, Mehmet Kemal Gullu, Ferhat Ozgur Catak, and Maliha Tabassum. Federated learning: Overview, strategies, applications, tools and future directions. *Heliyon*, 10(19), 2024.

A USE OF LARGE LANGUAGE MODELS (LLMS)

In adherence to the conference guidelines, we clarify the role of Large Language Models (LLMs) in preparing this manuscript. We utilized an LLM as a general-purpose assistive tool, primarily for language refinement and grammatical correction. The core scientific contributions of this work—including the initial research ideation, the development of the FedWNNM framework, the formulation of the PPF-SVD subroutine, all theoretical analyses, the design and execution of experiments, and the interpretation of results—were conceived and carried out exclusively by the human authors. The LLM was employed to improve the clarity and readability of the text. The authors take full responsibility for all content presented in this paper, including its scientific accuracy and integrity. The LLM is not considered an author of this work.

B ADMM DERIVATION FOR WNNM

To solve the constrained optimization problem in Eq. (2) using the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011), we first introduce an auxiliary variable $\boldsymbol{Z} \in \mathbb{R}^{m\times n}$ with the constraint $\boldsymbol{X} = \boldsymbol{Z}$. The variable $\boldsymbol{Z}$ is partitioned row-wise, consistent with $\boldsymbol{X}$, as $\boldsymbol{Z} = [\boldsymbol{Z}_1^T, \dots, \boldsymbol{Z}_p^T]^T$. The problem is then formulated via the augmented Lagrangian:

$$\mathcal{L}_\mu(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{\Lambda}) = \|\boldsymbol{X}\|_{w,*} + \sum_{i=1}^{p} I_{\boldsymbol{\Omega}_i}(\boldsymbol{Z}_i) + \langle \boldsymbol{\Lambda}, \boldsymbol{X} - \boldsymbol{Z}\rangle + \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{Z}\|_F^2,$$

where $\boldsymbol{\Lambda}$ is the matrix of Lagrange multipliers (partitioned as $\boldsymbol{\Lambda}_i \in \mathbb{R}^{m_i\times n}$), $\mu > 0$ is a penalty parameter, and $I_{\boldsymbol{\Omega}_i}(\boldsymbol{Z}_i)$ is an indicator function defined as:

$$I_{\boldsymbol{\Omega}_i}(\boldsymbol{Z}_i) = \begin{cases} 0 & \text{if } \mathcal{P}_{\boldsymbol{\Omega}_i}(\boldsymbol{Z}_i) = \mathcal{P}_{\boldsymbol{\Omega}_i}(\boldsymbol{Y}_i), \\ \infty & \text{otherwise.} \end{cases}$$

The ADMM algorithm proceeds by iteratively minimizing $\mathcal{L}_\mu$ with respect to one variable at a time, holding the others fixed. This procedure results in three subproblems for each iteration k.

The $\boldsymbol{X}$-update The update for $\boldsymbol{X}$ is the solution to the minimization problem

$$\begin{aligned} \boldsymbol{X}^{k+1} &= \arg\min_{\boldsymbol{X}} \mathcal{L}_\mu(\boldsymbol{X}, \boldsymbol{Z}^k, \boldsymbol{\Lambda}^k) \\ &= \arg\min_{\boldsymbol{X}} \left(\|\boldsymbol{X}\|_{w,*} + \langle \boldsymbol{\Lambda}^k, \boldsymbol{X} - \boldsymbol{Z}^k\rangle + \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{Z}^k\|_F^2 \right). \end{aligned}$$

By completing the square, the expression inside the minimum is equivalent to

$$\|\boldsymbol{X}\|_{w,*} + \frac{\mu}{2}\left\|\boldsymbol{X} - \left(\boldsymbol{Z}^k - \frac{\boldsymbol{\Lambda}^k}{\mu}\right)\right\|_F^2 - \frac{1}{2\mu}\|\boldsymbol{\Lambda}^k\|_F^2.$$

Dropping the constant term yields the equivalent problem

$$\boldsymbol{X}^{k+1} = \arg\min_{\boldsymbol{X}} \left(\|\boldsymbol{X}\|_{w,*} + \frac{\mu}{2}\|\boldsymbol{X} - \boldsymbol{A}^k\|_F^2 \right),$$

where $\boldsymbol{A}^k = \boldsymbol{Z}^k - \boldsymbol{\Lambda}^k/\mu$. This problem is an instance of proximal mapping for the weighted nuclear norm. The solution is obtained by applying weighted soft-thresholding to the singular values of $\boldsymbol{A}^k$. Let the SVD of $\boldsymbol{A}^k$ be $\boldsymbol{U}\,\mathrm{diag}(\{\sigma_j\})\boldsymbol{V}^T$. The update is then given by $\boldsymbol{X}^{k+1} = \boldsymbol{U}\,\mathrm{diag}(\{\tilde{\sigma}_j\})\boldsymbol{V}^T$, where $\tilde{\sigma}_j = \max(\sigma_j - w_j/\mu, 0)$.

The $\boldsymbol{Z}$-update The update for $\boldsymbol{Z}$ is the solution to

$$\begin{aligned} \boldsymbol{Z}^{k+1} &= \arg\min_{\boldsymbol{Z}} \mathcal{L}_\mu(\boldsymbol{X}^{k+1}, \boldsymbol{Z}, \boldsymbol{\Lambda}^k) \\ &= \arg\min_{\boldsymbol{Z}} \left(\sum_{i=1}^{p} I_{\boldsymbol{\Omega}_i}(\boldsymbol{Z}_i) + \langle \boldsymbol{\Lambda}^k, \boldsymbol{X}^{k+1} - \boldsymbol{Z}\rangle + \frac{\mu}{2}\|\boldsymbol{X}^{k+1} - \boldsymbol{Z}\|_F^2 \right). \end{aligned}$$

This problem decouples across the client partitions $\boldsymbol{Z}_i$. For each client i, the problem is

$$\boldsymbol{Z}_i^{k+1} = \arg\min_{\boldsymbol{Z}_i} \left(I_{\boldsymbol{\Omega}_i}(\boldsymbol{Z}_i) + \frac{\mu}{2}\left\|\boldsymbol{Z}_i - \left(\boldsymbol{X}_i^{k+1} + \frac{\boldsymbol{\Lambda}_i^k}{\mu}\right)\right\|_F^2 \right).$$

This is a projection problem. The solution enforces the constraint imposed by the indicator function $I_{\boldsymbol{\Omega}_i}$. For entries $(j,l) \in \boldsymbol{\Omega}_i$, we must have $(\boldsymbol{Z}_i)_{j,l} = (\boldsymbol{Y}_i)_{j,l}$. For entries $(j,l) \notin \boldsymbol{\Omega}_i$, the quadratic term is minimized by setting $(\boldsymbol{Z}_i)_{j,l} = (\boldsymbol{X}_i^{k+1} + \boldsymbol{\Lambda}_i^k/\mu)_{j,l}$. This yields the update rule

$$\boldsymbol{Z}_i^{k+1} = \mathcal{P}_{\boldsymbol{\Omega}_i}(\boldsymbol{Y}_i) + \mathcal{P}_{\boldsymbol{\Omega}_i^c}(\boldsymbol{X}_i^{k+1} + \frac{\boldsymbol{\Lambda}_i^k}{\mu}).$$

This step is performed entirely locally at each client.

The Multiplier-update The Lagrange multipliers are updated using gradient ascent:

$$\mathbf{\Lambda}^{k+1} = \mathbf{\Lambda}^k + \mu(\boldsymbol{X}^{k+1} - \boldsymbol{Z}^{k+1}).$$

This update also decouples by client, and each client i computes its local update:

$$\mathbf{\Lambda}_i^{k+1} = \mathbf{\Lambda}_i^k + \mu(\boldsymbol{X}_i^{k+1} - \boldsymbol{Z}_i^{k+1}).$$

C Proofs of Theoretical Results

C.1 Proof of Theorem 3.1

Proof. Let $\boldsymbol{Y} = \boldsymbol{A}\mathbf{\Omega}$. By definition, $\boldsymbol{Y}$ is formed by horizontally stacking the local sketches $\boldsymbol{Y}_i = \boldsymbol{A}_i\mathbf{\Omega}$:

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Y}_1 \\ \boldsymbol{Y}_2 \\ \vdots \\ \boldsymbol{Y}_p \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_1\mathbf{\Omega} \\ \boldsymbol{A}_2\mathbf{\Omega} \\ \vdots \\ \boldsymbol{A}_p\mathbf{\Omega} \end{pmatrix}.$$

Substituting the local thin QR factorization $\boldsymbol{A}_i\mathbf{\Omega} = \boldsymbol{Q}_{Y,i}\boldsymbol{R}_{Y,i}$ for each block yields

$$\boldsymbol{Y} = \begin{pmatrix} \boldsymbol{Q}_{Y,1}\boldsymbol{R}_{Y,1} \\ \boldsymbol{Q}_{Y,2}\boldsymbol{R}_{Y,2} \\ \vdots \\ \boldsymbol{Q}_{Y,p}\boldsymbol{R}_{Y,p} \end{pmatrix}.$$

This can be expressed in factored form as

$$\boldsymbol{Y} = \operatorname{diag}(\boldsymbol{Q}_{Y,1}, \ldots, \boldsymbol{Q}_{Y,p}) \begin{pmatrix} \boldsymbol{R}_{Y,1} \\ \boldsymbol{R}_{Y,2} \\ \vdots \\ \boldsymbol{R}_{Y,p} \end{pmatrix} = \operatorname{diag}(\boldsymbol{Q}_{Y,1}, \ldots, \boldsymbol{Q}_{Y,p})\boldsymbol{R}_Y, \tag{4}$$

where $\boldsymbol{R}_Y = [\boldsymbol{R}_{Y,1}^T, \ldots, \boldsymbol{R}_{Y,p}^T]^T$.

Next, we substitute the thin QR factorization of $\boldsymbol{R}_Y = \boldsymbol{Q}_R\boldsymbol{R}_{\text{agg}}$ into Eq. (4):

$$\boldsymbol{Y} = \operatorname{diag}(\boldsymbol{Q}_{Y,1}, \ldots, \boldsymbol{Q}_{Y,p})(\boldsymbol{Q}_R\boldsymbol{R}_{\text{agg}}) = (\operatorname{diag}(\boldsymbol{Q}_{Y,1}, \ldots, \boldsymbol{Q}_{Y,p})\boldsymbol{Q}_R)\,\boldsymbol{R}_{\text{agg}}.$$

By the definition given in the theorem statement, $\boldsymbol{Q} = \operatorname{diag}(\boldsymbol{Q}_{Y,1}, \ldots, \boldsymbol{Q}_{Y,p})\boldsymbol{Q}_R$. Thus, we have the relationship $\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{R}_{\text{agg}}$.

First, we verify that $\boldsymbol{Q}$ has orthonormal columns. Let $\boldsymbol{Q}_{\text{diag}} = \operatorname{diag}(\boldsymbol{Q}_{Y,1}, \ldots, \boldsymbol{Q}_{Y,p})$. The matrices $\boldsymbol{Q}_{Y,i}$ and $\boldsymbol{Q}_R$ have orthonormal columns by definition of the QR factorization. Therefore,

$$\boldsymbol{Q}^T\boldsymbol{Q} = (\boldsymbol{Q}_{\text{diag}}\boldsymbol{Q}_R)^T(\boldsymbol{Q}_{\text{diag}}\boldsymbol{Q}_R) = \boldsymbol{Q}_R^T(\boldsymbol{Q}_{\text{diag}}^T\boldsymbol{Q}_{\text{diag}})\boldsymbol{Q}_R = \boldsymbol{Q}_R^T\boldsymbol{I}_{pd}\boldsymbol{Q}_R = \boldsymbol{Q}_R^T\boldsymbol{Q}_R = \boldsymbol{I}_d.$$

Since $\boldsymbol{Q}$ has orthonormal columns, the matrix $\boldsymbol{P}_{\boldsymbol{Q}} = \boldsymbol{Q}\boldsymbol{Q}^T$ is an orthogonal projector onto the subspace $\operatorname{range}(\boldsymbol{Q})$.

The expression $\boldsymbol{Y} = \boldsymbol{Q}\boldsymbol{R}_{\text{agg}}$ implies that any vector in the column space of $\boldsymbol{Y}$ can be written as a linear combination of the columns of $\boldsymbol{Q}$. This means $\operatorname{range}(\boldsymbol{Y}) \subseteq \operatorname{range}(\boldsymbol{Q})$.

Let $\boldsymbol{P}_Y$ be the orthogonal projector onto the subspace $\operatorname{range}(\boldsymbol{Y}) = \operatorname{range}(\boldsymbol{A}\mathbf{\Omega})$. The approximation error we seek to bound is $\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}\|_2$. Since $\operatorname{range}(\boldsymbol{Y}) \subseteq \operatorname{range}(\boldsymbol{Q})$, the projection of $\boldsymbol{A}$ onto the larger subspace $\operatorname{range}(\boldsymbol{Q})$ yields an approximation that is at least as good as the projection onto the smaller subspace $\operatorname{range}(\boldsymbol{Y})$. Thus, we have the inequality:

$$\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}\|_2 \leq \|\boldsymbol{A} - \boldsymbol{P}_Y\boldsymbol{A}\|_2.$$

The term on the right, $\|\boldsymbol{A} - \boldsymbol{P}_Y\boldsymbol{A}\|_2$, is the error of the optimal projection of $\boldsymbol{A}$ onto the subspace spanned by $\boldsymbol{A}\mathbf{\Omega}$, which is the standard setup in randomized numerical linear algebra. We can invoke Theorem 9.1 from Halko et al. (2011) to bound this term.

Let the SVD of $\boldsymbol{A}$ be partitioned as $\boldsymbol{A} = \boldsymbol{U}_1\mathbf{\Sigma}_1\boldsymbol{V}_1^T + \boldsymbol{U}_2\mathbf{\Sigma}_2\boldsymbol{V}_2^T$, where $\mathbf{\Sigma}_1$ contains the k largest singular values. Under the theorem's hypothesis that $\boldsymbol{V}_1^T\mathbf{\Omega}$ has full row rank, Theorem 9.1 of Halko et al. (2011) states that:

$$\|\boldsymbol{A} - \boldsymbol{P}_Y\boldsymbol{A}\|_2^2 \leq \|\mathbf{\Sigma}_2\|_2^2 + \|\mathbf{\Sigma}_2(\boldsymbol{V}_2^T\mathbf{\Omega})(\boldsymbol{V}_1^T\mathbf{\Omega})^\dagger\|_2^2.$$

By combining the inequality and the theorem, we arrive at the desired result:

$$\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}\|_2^2 \leq \|\mathbf{\Sigma}_2\|_2^2 + \|\mathbf{\Sigma}_2(\boldsymbol{V}_2^T\mathbf{\Omega})(\boldsymbol{V}_1^T\mathbf{\Omega})^\dagger\|_2^2.$$

This completes the proof. □

C.2 Proof of Corollary 3.2

Theorem 3.1 provides a general error bound for an arbitrary test matrix $\boldsymbol{\Omega}$. To illuminate the trade-off between approximation accuracy and numerical stability, we now specialize this result to a class of structured test matrices whose condition numbers can be controlled.

The fidelity of the approximation is fundamentally governed by the geometric relationship between the signal subspace, $\mathrm{range}(\boldsymbol{V}_1)$, and the test subspace, $\mathrm{range}(\boldsymbol{U}_\Omega)$. This relationship is canonically quantified by the principal angles between the two subspaces. For completeness, we recall the formal definition.

Definition C.1 (Principal Angles (Knyazev & Zhu, 2012))**.** *Let $\mathcal{U} = \mathrm{range}(\boldsymbol{U})$ and $\mathcal{V} = \mathrm{range}(\boldsymbol{V})$ be two subspaces of $\mathbb{R}^n$ of dimensions p and q, respectively, where $p, q \geq 1$. The $m = \min(p, q)$ principal angles $0 \leq \theta_1 \leq \cdots \leq \theta_m \leq \pi/2$ between $\mathcal{U}$ and $\mathcal{V}$ are defined recursively for $i = 1, \ldots, m$ by*

$$\cos\theta_i = \max_{\substack{\boldsymbol{u}\in\mathcal{U}, \boldsymbol{v}\in\mathcal{V} \\ \|\boldsymbol{u}\|_2=\|\boldsymbol{v}\|_2=1}} \boldsymbol{u}^T\boldsymbol{v} \quad \textit{subject to} \quad \boldsymbol{u}^T\boldsymbol{u}_j = 0, \boldsymbol{v}^T\boldsymbol{v}_j = 0 \textit{ for } j = 1, \ldots, i-1,$$

where $(\boldsymbol{u}_j, \boldsymbol{v}_j)$ is a pair of principal vectors corresponding to the principal angle θ_j. If the columns of $\boldsymbol{U}$ and $\boldsymbol{V}$ form orthonormal bases for $\mathcal{U}$ and $\mathcal{V}$, respectively, then the cosines of the principal angles, $\cos\theta_i$, are the singular values of the matrix $\boldsymbol{U}^T\boldsymbol{V}$.

Armed with the concept of principal angles, we are now positioned to establish a more explicit error bound. This bound will directly relate the approximation error to the geometric alignment of the signal and test subspaces, as measured by their smallest principal angle, and the condition number of the test matrix.

Proof. The proof proceeds by establishing the lower and upper bounds for the approximation error separately.

1. Lower Bound. The approximation of $\boldsymbol{A}$ generated by the procedure is $\hat{\boldsymbol{A}} = \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}$. The matrix $\boldsymbol{Q} \in \mathbb{R}^{m\times d}$ has, by construction, d orthonormal columns. The associated orthogonal projector $\boldsymbol{P}_{\boldsymbol{Q}} = \boldsymbol{Q}\boldsymbol{Q}^T$ is a matrix of rank d. The rank of the approximation $\hat{\boldsymbol{A}}$ is therefore bounded by the rank of $\boldsymbol{P}_{\boldsymbol{Q}}$:

$$\mathrm{rank}(\hat{\boldsymbol{A}}) = \mathrm{rank}(\boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}) \leq \mathrm{rank}(\boldsymbol{Q}\boldsymbol{Q}^T) = d.$$

The Eckart-Young-Mirsky theorem (Eckart & Young, 1936) states that the error of approximating a matrix $\boldsymbol{A}$ by any matrix $\boldsymbol{B}$ of rank at most d is bounded below by the error of the optimal rank-d approximation. This optimal error is the $(d+1)$-th singular value of $\boldsymbol{A}$. As $\hat{\boldsymbol{A}}$ is a matrix with $\mathrm{rank}(\hat{\boldsymbol{A}}) \leq d$, it must satisfy this bound:

$$\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}\|_2 \geq \sigma_{d+1}(\boldsymbol{A}).$$

This establishes the lower bound on the error.

2. Upper Bound. The upper bound is derived from the result of Theorem 3.1, which states

$$\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}\|_2^2 \leq \|\boldsymbol{\Sigma}_2\|_2^2 + \|\boldsymbol{\Sigma}_2(\boldsymbol{V}_2^T\boldsymbol{\Omega})(\boldsymbol{V}_1^T\boldsymbol{\Omega})^\dagger\|_2^2. \tag{5}$$

Let the term $\boldsymbol{\mathcal{E}} = \boldsymbol{\Sigma}_2(\boldsymbol{V}_2^T\boldsymbol{\Omega})(\boldsymbol{V}_1^T\boldsymbol{\Omega})^\dagger$. Given the decomposition $\boldsymbol{\Omega} = \boldsymbol{U}_\Omega\boldsymbol{D}_\Omega\boldsymbol{V}_\Omega^T$, the products with $\boldsymbol{V}_1^T$ and $\boldsymbol{V}_2^T$ are

$$\boldsymbol{V}_1^T\boldsymbol{\Omega} = (\boldsymbol{V}_1^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega\boldsymbol{V}_\Omega^T,$$
$$\boldsymbol{V}_2^T\boldsymbol{\Omega} = (\boldsymbol{V}_2^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega\boldsymbol{V}_\Omega^T.$$

Since $\boldsymbol{V}_\Omega$ is orthogonal and $\boldsymbol{D}_\Omega$ is invertible, the Moore-Penrose pseudoinverse of $\boldsymbol{V}_1^T\boldsymbol{\Omega}$ is $(\boldsymbol{V}_1^T\boldsymbol{\Omega})^\dagger = \boldsymbol{V}_\Omega((\boldsymbol{V}_1^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega)^\dagger$. Substituting this into the expression for $\boldsymbol{\mathcal{E}}$ yields

$$\begin{aligned}\boldsymbol{\mathcal{E}} &= \boldsymbol{\Sigma}_2(\boldsymbol{V}_2^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega\boldsymbol{V}_\Omega^T\left(\boldsymbol{V}_\Omega((\boldsymbol{V}_1^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega)^\dagger\right) \\ &= \boldsymbol{\Sigma}_2(\boldsymbol{V}_2^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega((\boldsymbol{V}_1^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega)^\dagger.\end{aligned}$$

By the submultiplicative property of the spectral norm,

$$\|\boldsymbol{\mathcal{E}}\|_2 \leq \|\boldsymbol{\Sigma}_2\|_2\|(\boldsymbol{V}_2^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega\|_2\|((\boldsymbol{V}_1^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega)^\dagger\|_2. \tag{6}$$

We bound each term on the right-hand side.

1. By definition, $\|\boldsymbol{\Sigma}_2\|_2 = \sigma_{k+1}(\boldsymbol{A})$.
2. Since the columns of $\boldsymbol{V}_2$ and $\boldsymbol{U}_\Omega$ are orthonormal and $\|\boldsymbol{D}_\Omega\|_2 = 1$, it follows that $\|(\boldsymbol{V}_2^T\boldsymbol{U}_\Omega)\boldsymbol{D}_\Omega\|_2 \leq \|\boldsymbol{V}_2^T\boldsymbol{U}_\Omega\|_2\|\boldsymbol{D}_\Omega\|_2 \leq 1$.

3. Let $\boldsymbol{G}_1 = \boldsymbol{V}_1^T \boldsymbol{U}_\Omega$. The norm of the pseudoinverse is the reciprocal of the smallest non-zero singular value. Since $\boldsymbol{G}_1$ has full row rank k,

$$\|(\boldsymbol{G}_1 \boldsymbol{D}_\Omega)^\dagger\|_2 = \frac{1}{\sigma_k(\boldsymbol{G}_1 \boldsymbol{D}_\Omega)}.$$

Using the inequality $\sigma_k(\boldsymbol{XY}) \geq \sigma_k(\boldsymbol{X})\sigma_{\min}(\boldsymbol{Y})$, we obtain

$$\sigma_k(\boldsymbol{G}_1 \boldsymbol{D}_\Omega) \geq \sigma_k(\boldsymbol{G}_1)\sigma_k(\boldsymbol{D}_\Omega).$$

The k-th singular value of $\boldsymbol{G}_1$ is $\sigma_k(\boldsymbol{G}_1) = \cos\theta_k$, and the minimum singular value of $\boldsymbol{D}_\Omega$ is $\sigma_{\min}(\boldsymbol{D}_\Omega) = \rho^{d-1}$. This gives $\sigma_k(\boldsymbol{G}_1 \boldsymbol{D}_\Omega) \geq (\cos\theta_k)(\rho^{d-1})$, which implies

$$\|(\boldsymbol{G}_1 \boldsymbol{D}_\Omega)^\dagger\|_2 \leq \frac{1}{(\cos\theta_k)(\rho^{d-1})}.$$

Substituting these bounds into Eq. (6) gives

$$\|\boldsymbol{\mathcal{E}}\|_2 \leq \sigma_{k+1}(\boldsymbol{A}) \cdot 1 \cdot \frac{1}{(\cos\theta_k)(\rho^{d-1})} = \sigma_{k+1}(\boldsymbol{A})\frac{\kappa(\boldsymbol{D}_\Omega)}{\cos\theta_k},$$

where $\kappa(\boldsymbol{D}_\Omega) = \sigma_{\max}(\boldsymbol{D}_\Omega)/\sigma_{\min}(\boldsymbol{D}_\Omega) = 1/\rho^{d-1}$. Finally, substituting this result into Eq. (5) yields

$$\begin{aligned}\|\boldsymbol{A} - \boldsymbol{Q}\boldsymbol{Q}^T\boldsymbol{A}\|_2^2 &\leq \sigma_{k+1}(\boldsymbol{A})^2 + \left(\sigma_{k+1}(\boldsymbol{A})\frac{\kappa(\boldsymbol{D}_\Omega)}{\cos\theta_k}\right)^2 \\ &= \sigma_{k+1}(\boldsymbol{A})^2\left(1 + \frac{\kappa(\boldsymbol{D}_\Omega)^2}{\cos^2\theta_k}\right).\end{aligned}$$

Taking the square root of both sides gives the upper bound. The second equality in the corollary statement follows from substituting $\kappa(\boldsymbol{D}_\Omega) = \rho^{-(d-1)}$ and using the identity $\sec\theta_k = 1/\cos\theta_k$. Combining the bounds completes the proof. □

D Communication Cost

This section provides a rigorous mathematical analysis of the communication cost per iteration of the Federated WNNM algorithm (Algorithm 2). The cost is quantified as the total number of floating-point values transmitted between the central server and the participating clients during a single Alternating Direction Method of Multipliers (ADMM) iteration.

Let the number of clients be denoted by p. The global data matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ is partitioned such that client i holds $\boldsymbol{A}_i \in \mathbb{R}^{m_i\times n}$, where $\sum_{i=1}^p m_i = m$. The objective is to find a rank-k approximation. We define the oversampling parameter as p_{over}, leading to a sketching dimension of $d = k + p_{\text{over}}$.

The communication flow is dissected into two primary channels: Client-to-Server (C2S) and Server-to-Client (S2C).

D.1 Derivation of Communication Cost per ADMM Iteration

One complete ADMM iteration in Algorithm 2 involves a federated $\boldsymbol{X}$-update (which invokes the PPF-SVD subroutine), local $\boldsymbol{Z}$-updates, and a dual ascent step with a convergence check. We analyze the communication incurred by each phase.

D.1.1 Phase 1: Federated $\boldsymbol{X}$-Update (via PPF-SVD)

This phase is the most communication-intensive part of the iteration.

Step 1 (S2C): Broadcast of Sketching Matrix The server initiates PPF-SVD by generating a sketching matrix $\boldsymbol{\Omega} \in \mathbb{R}^{n\times d}$ and broadcasting it to all p clients (Algorithm 1, line 5). The cost is:

$$C_1 = p \cdot n \cdot d$$

Step 2 (C2S): Transmission of Local QR Factors Each client i computes its local sketch $\boldsymbol{Y}_i = \boldsymbol{A}_i\boldsymbol{\Omega} \in \mathbb{R}^{m_i\times d}$. Following a local QR decomposition, each client transmits the upper triangular factor $\boldsymbol{R}_{Y,i} \in \mathbb{R}^{d\times d}$ to the server (Algorithm 1, line 9). The cost is:

$$C_2 = p \cdot d^2$$

Step 3 (C2S): Transmission of Local Projections Each client i transmits the projected matrix $\boldsymbol{B}_i' = \boldsymbol{Q}_{Y,i}^T \boldsymbol{A}_i \in \mathbb{R}^{d\times n}$ to the server (Algorithm 1, line 15). The cost is:

$$C_3 = p \cdot d \cdot n$$

Step 4 (S2C): Broadcast of Global Low-Rank Factors The server distributes components for reconstructing $(\boldsymbol{X}^{iter+1})_i$. This combines communications from Algorithm 1 (line 21) and Algorithm 2 (line 13). The server broadcasts to all p clients: $\tilde{\boldsymbol{U}} \in \mathbb{R}^{d\times k}$, $(\boldsymbol{Q}_R)_i \in \mathbb{R}^{d\times d}$ for each client, a vector of k singular values, and $\boldsymbol{V} \in \mathbb{R}^{n\times k}$. The total cost is:

$$C_4 = p \cdot (d \cdot k + d^2 + k + n \cdot k) = p(d^2 + dk + (n+1)k)$$

D.1.2 Phase 2: Federated $\boldsymbol{Z}$-Update

This update (Algorithm 2, line 16) is performed entirely on the client side.

$$C_5 = 0$$

D.1.3 Phase 3: Dual Update and Convergence Check

To check for global convergence, each client transmits two scalar values (r_i^2, s_i^2) to the server (Algorithm 2, line 20).

$$C_6 = 2p$$

D.2 Total Communication Cost

The total communication cost per ADMM iteration is the summation of the costs from all phases.

Total Client-to-Server (C2S) Cost:

$$\begin{aligned} C_{\text{C2S}} &= C_2 + C_3 + C_6 \\ &= pd^2 + pdn + 2p \\ &= p(dn + d^2 + 2) \end{aligned}$$

Total Server-to-Client (S2C) Cost:

$$\begin{aligned} C_{\text{S2C}} &= C_1 + C_4 \\ &= pnd + p(d^2 + dk + (n+1)k) \\ &= p(nd + d^2 + (n + d + 1)k) \end{aligned}$$

Total Aggregated Communication Cost: The total communication cost per iteration, C_{total}, is the sum of the C2S and S2C costs.

$$\begin{aligned} C_{\text{total}} &= C_{\text{C2S}} + C_{\text{S2C}} \\ &= p(dn + d^2 + 2) + p(nd + d^2 + (n + d + 1)k) \\ &= p[2dn + 2d^2 + (n + d + 1)k + 2] \end{aligned}$$

By substituting the definition $d = k + p_{\text{over}}$, we arrive at the final expression:

$$C_{\text{total}} = p[2n(k + p_{\text{over}}) + 2(k + p_{\text{over}})^2 + (n + k + p_{\text{over}} + 1)k + 2]$$

This result confirms that the communication complexity scales linearly with both the number of clients, p, and the feature dimension, n.

E Empirical Validation and Sensitivity Analysis

We now empirically validate the proposed PPF-SVD algorithm 1 and the FedWNNM algorithm 2, with a focus on the sensitivity to key hyperparameters.

E.1 Analysis of the Oversampling Parameter p_{over}

We first examine the influence of the oversampling parameter, p_{over}, on the accuracy of Algorithm 1. In this experiment, we fix the privacy parameter to a value ($\rho = 0.1$) to isolate the effect of oversampling. Using the default federated setting (8 clients, a 1000×500 global matrix with true rank 20), we vary p_{over} and measure the same set of relative error metrics.

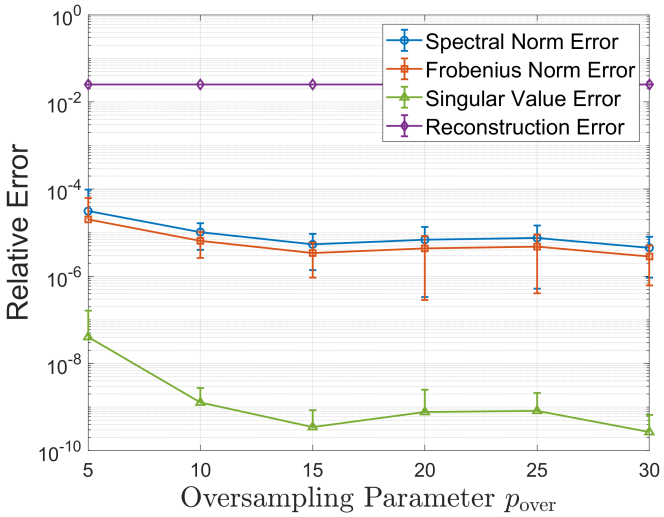
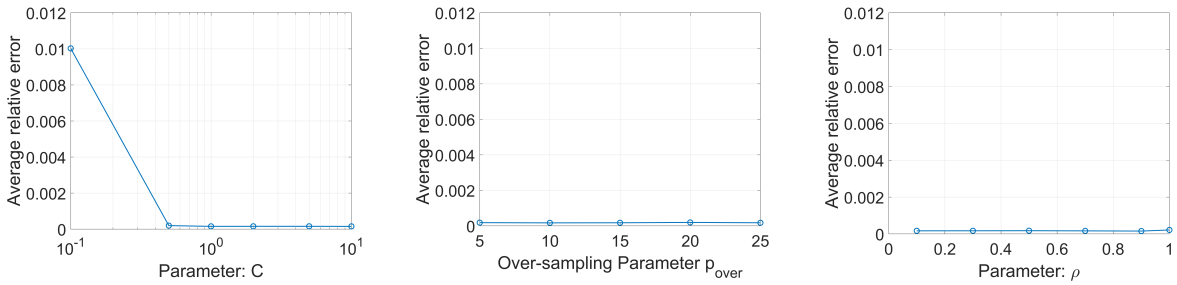


Figure 5: Effect of the oversampling parameter p_{over} on relative error metrics, plotted on a semi-logarithmic scale. Increasing p_{over} consistently improves the approximation accuracy across all metrics. The privacy parameter is held constant at $\rho = 0.1$.

The results are depicted in Figure 5. As anticipated, increasing the oversampling parameter leads to a monotonic decrease in all approximation-related errors (spectral, Frobenius, and singular value), as shown by the logarithmic scale on the y-axis. This is because a larger p_{over} expands the dimension of the random projection, increasing the probability of capturing the complete action of the matrix's dominant singular vectors. This experiment demonstrates that oversampling serves as an effective mechanism to enhance decomposition accuracy. However, this improvement comes at the cost of increased computational and communication overhead, as clients must compute and transmit larger intermediate matrices. Therefore, the choice of p_{over} represents a trade-off between accuracy and resource efficiency.

E.2 Robustness and Hyperparameter Sensitivity of FedWNNM

We now analyze the sensitivity of the complete FedWNNM framework with respect to its principal hyperparameters: the weight parameter C, the oversampling parameter p_{over}, and the projector parameter ρ. This analysis was conducted on a 500×500 matrix (rank 20, 4 clients) with a fixed and challenging observation ratio of $p_{\text{obs}} = 0.3$.



(a) Sensitivity to weight parameter C. (b) Impact of oversampling p_{over}. (c) Sensitivity to parameter ρ.

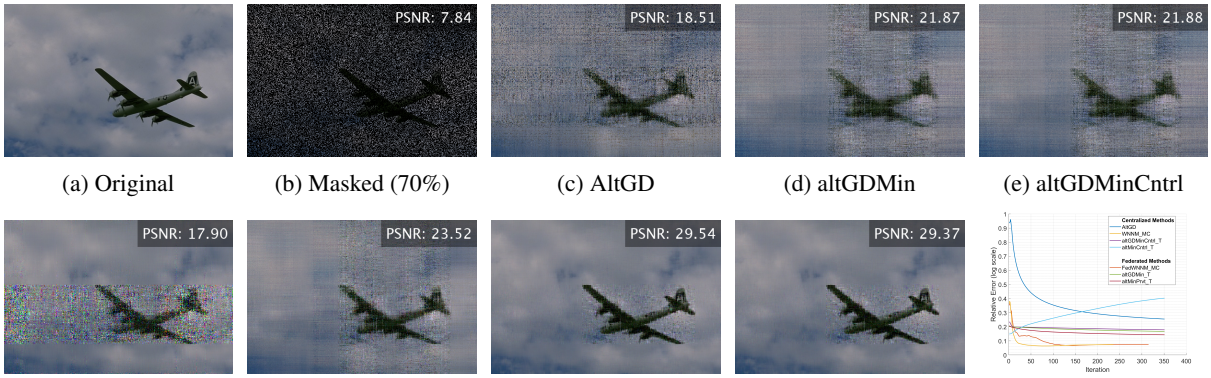
Figure 6: Sensitivity analysis of the FedWNNM algorithm with respect to its principal hyperparameters, evaluated in terms of Average Relative Error.

The sensitivity of the algorithm to its main hyperparameters is presented in Figure 6. The results indicate that the algorithm is stable with respect to these parameters. As shown in Figure 6a, the mean relative error varies

minimally for the weight parameter C over the interval $[0.1, 10]$. Similarly, Figures 6b and 6c demonstrate that the relative error is insensitive to variations in the oversampling parameter p_{over} and the projector parameter ρ. For all tested values of these hyperparameters, the relative error remains on the order of 10^{-4}. This insensitivity to hyperparameter selection is a significant practical advantage, as it suggests that the FedWNNM framework can maintain low recovery error without extensive parameter tuning.

F Qualitative Comparison of Image Inpainting Results

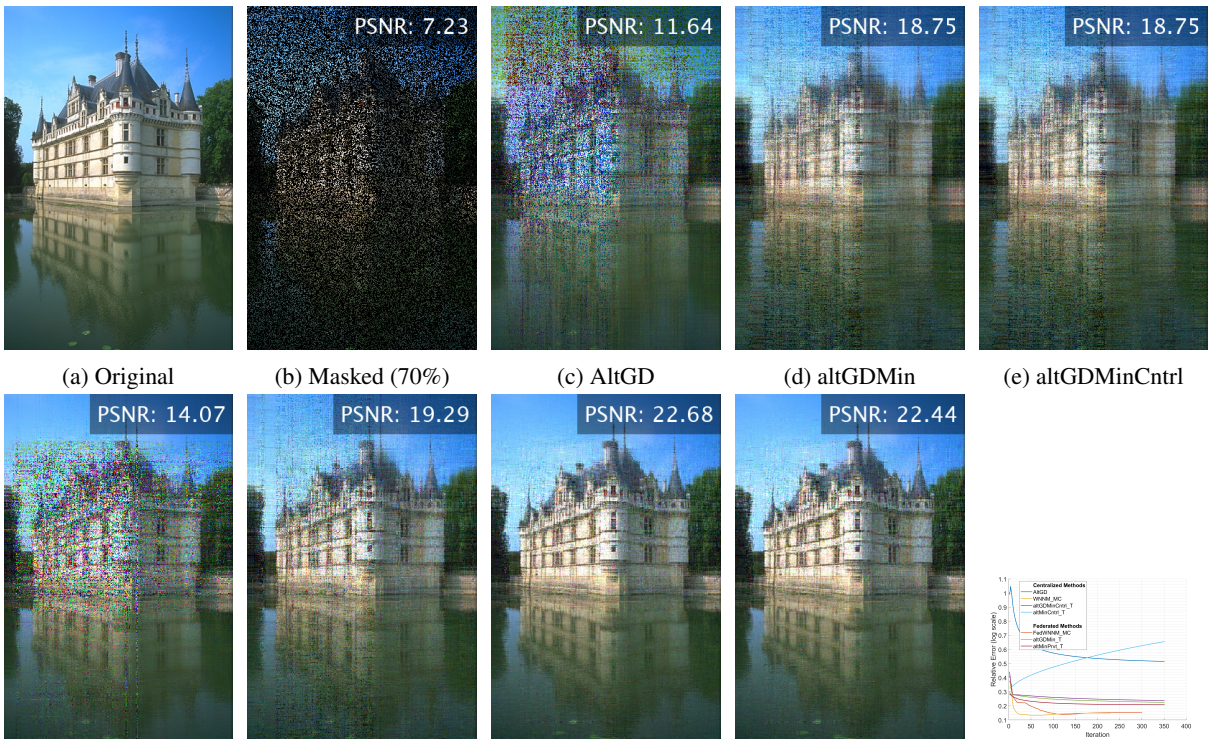
This appendix presents a comprehensive set of qualitative results for the image inpainting task on the CBSD68 dataset. We provide visual comparisons under 70% missing rates. Each figure showcases a side-by-side comparison, including the original image, the corrupted (masked) image, the reconstructions from various baseline methods, and the result from our proposed



(a) Original (b) Masked (70%) (c) AltGD (d) altGDMin (e) altGDMinCntrl

(f) altMinCntrl (g) altMinPrvt (h) WNNM (i) **FedWNNM** (Ours) (j) Convergence

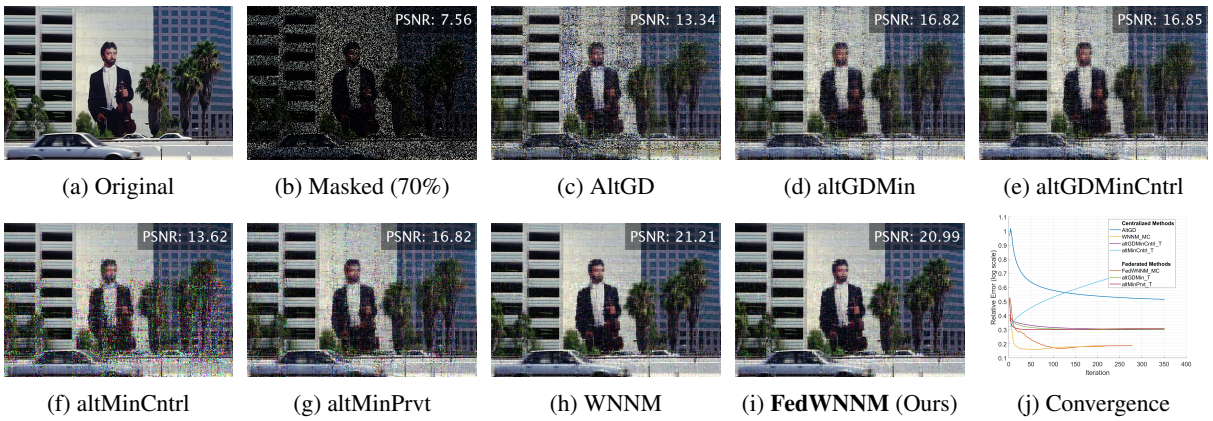
Figure 7: Qualitative comparison of image inpainting with a 70% pixel missing rate for image 0000. The final subfigure shows the convergence curves for relative error.



(a) Original (b) Masked (70%) (c) AltGD (d) altGDMin (e) altGDMinCntrl

(f) altMinCntrl (g) altMinPrvt (h) WNNM (i) **FedWNNM** (Ours) (j) Convergence

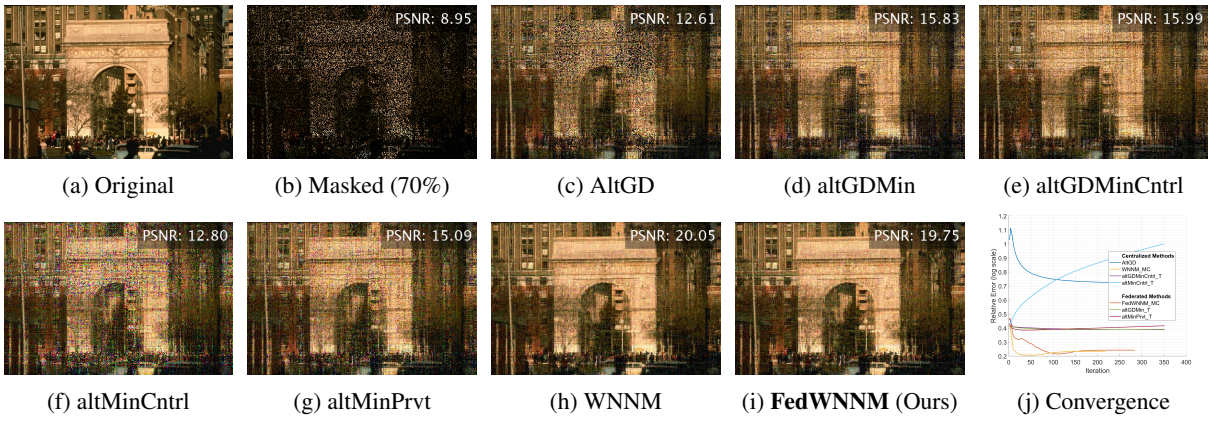
Figure 8: Qualitative comparison of image inpainting with a 70% pixel missing rate for image 0010. The final subfigure shows the convergence curves for relative error.



(a) Original (b) Masked (70%) (c) AltGD (d) altGDMin (e) altGDMinCntrl

(f) altMinCntrl (g) altMinPrvt (h) WNNM (i) **FedWNNM** (Ours) (j) Convergence

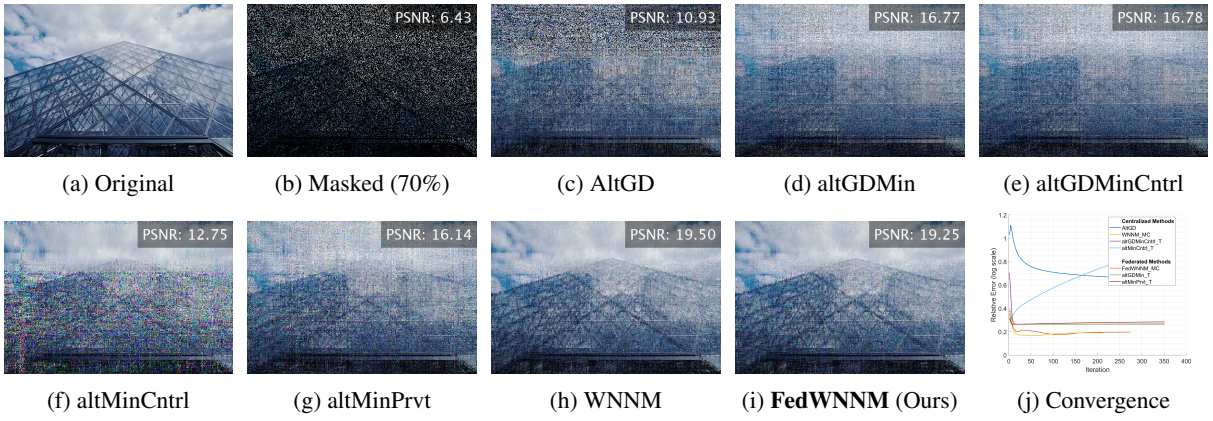
Figure 9: Qualitative comparison of image inpainting with a 70% pixel missing rate for image 0018. The final subfigure shows the convergence curves for relative error.



(a) Original (b) Masked (70%) (c) AltGD (d) altGDMin (e) altGDMinCntrl

(f) altMinCntrl (g) altMinPrvt (h) WNNM (i) **FedWNNM** (Ours) (j) Convergence

Figure 10: Qualitative comparison of image inpainting with a 70% pixel missing rate for image 0027. The final subfigure shows the convergence curves for relative error.



(a) Original (b) Masked (70%) (c) AltGD (d) altGDMin (e) altGDMinCntrl

(f) altMinCntrl (g) altMinPrvt (h) WNNM (i) **FedWNNM** (Ours) (j) Convergence

Figure 11: Qualitative comparison of image inpainting with a 70% pixel missing rate for image 0046. The final subfigure shows the convergence curves for relative error.