

FEDERATED DEBLURRING AS A JIGSAW PUZZLE: A PRIVACY-PRESERVING CONSENSUS FRAMEWORK

Anonymous authors

Paper under double-blind review

ABSTRACT

Multi-observation deblurring seeks to recover a clear image from multiple blurred observations, yet most methods assume centralized access to full-scene data, which is unrealistic in practical scenarios where images are captured by distributed, independent devices. We introduce a more realistic federated setting where each client holds a private, partially overlapping view of a larger scene. This creates an “information jigsaw puzzle” that must be solved under strict privacy and regulatory constraints, without sharing raw images, kernels, or intermediate image estimates. We propose `FedDeblur`, a principled federated optimization framework based on consensus that decouples client-side local data fidelity updates and a server-side global image prior update. Clients transmit only carefully designed, desensitized variables, while the server coordinates global consensus. The modularity of our framework enables the server to flexibly incorporate diverse image priors, bridging classic regularizers like total variation with modern deep plug-and-play (PnP) denoisers, all transparently to clients. Crucially, all client-side updates admit efficient closed-form solutions, eliminating the need for inner iterations and making our framework practical for resource-constrained edge devices. Experiments demonstrate that `FedDeblur` seamlessly integrates fragmented information from partial views, effectively solving the jigsaw puzzle and achieving performance close to an idealized, non-private centralized oracle.

1 INTRODUCTION

Image deblurring is a classic inverse problem in digital image processing, which aims to recover a clear image from a blurred observation. Blur arises from lens imperfections, sensor characteristics, and camera or scene motion, and always coexists with random noise. These image degradations not only degrade visual quality but also harms downstream tasks such as classification, detection, and segmentation (Zhang et al., 2022; Aakanksha & Rajagopalan, 2023). Image deblurring therefore has broad practical value in photography, astronomy, medical imaging, and remote sensing (Donath et al., 2024; Feijoo et al., 2025).

Image blur is commonly modeled as a convolution of a clear image with a blur kernel, or point spread function (PSF). Since recovering the image from the observation is ill-posed, the problem is typically addressed by incorporating a regularizer that reflects prior knowledge about the clear image. Classic regularizers include total variation (TV) (Rudin et al., 1992) and its variants (Bredies et al., 2010), as well as priors based on the ℓ_q ($q < 1$) norm (Xu et al., 2011; Zuo et al., 2013), and other non-convex functions (You et al., 2019; Ru et al., 2022). More recently, implicit priors learned through deep learning have also been applied to improve image restoration performance (Venkatakrishnan et al., 2013; Zhang et al., 2021a; Wei et al., 2024).

Beyond designing priors for a single image, an alternative strategy exploits multiple observations of the same scene to jointly recover the latent image. When kernels are weakly correlated, each observation carries complementary information; aggregating them strengthens data-driven constraints and reduces reliance on strong priors. Existing multi-observation methods fall into two groups: model-based iterative methods that design priors and solve an objective iteratively (e.g., Sroubek & Milanfar (2011); Zhang et al. (2014)); and deep learning methods that learn an end-to-end mapping from multiple blurry inputs to one clear image (e.g., Aittala & Durand (2018); Oh & Kim (2022)). Despite good results in specific settings, these approaches struggle with generalizable priors and often incur heavy computation and storage, limiting deployment on edge devices.

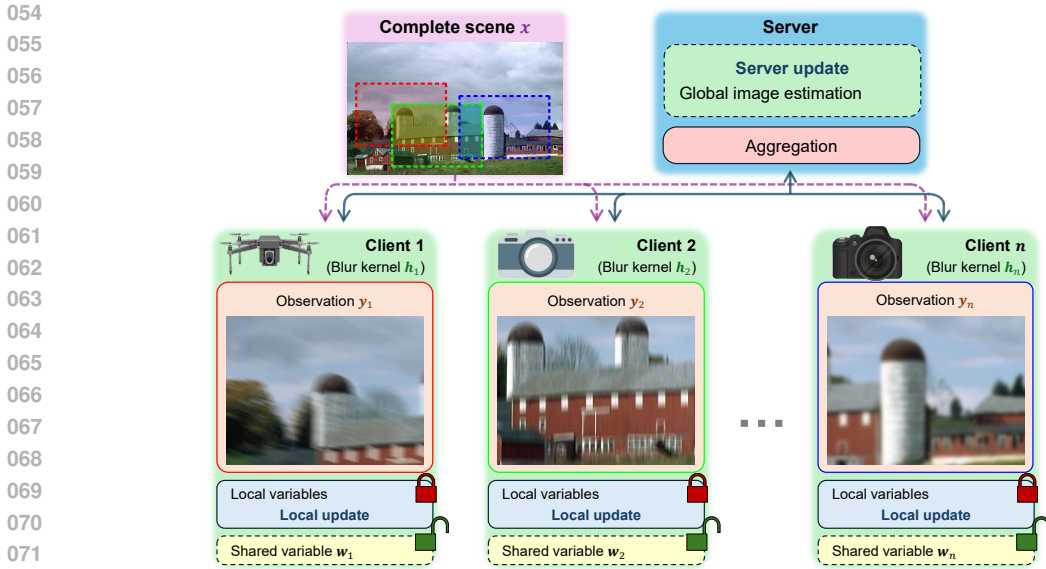


Figure 1: Overview of the proposed federated multi-observation deblurring framework. A central server coordinates multiple clients, each holding a private observation y_k of the scene x and a private kernel h_k . Per round, clients update local variables using private data and send only shared variables w_k ; the server aggregates them to update the global estimate and broadcasts it to guide the next round.

More importantly, most existing works assume centralized processing—all raw data are collected at one node, exposing pixel-level images and kernels. In practice, observations often come from independent devices with distinct hardware and viewpoints, naturally producing different blur kernels. Moreover, a more realistic, challenging case arises when each device captures only a region of the scene, and these regions overlap only partially (Figure 1). Such partially overlapping views form an “information jigsaw puzzle” that breaks the standard assumption of identical full-scene observations.

Furthermore, the degraded observations and their blur kernels can also leak sensitive device or user information. For example, the photo-response non-uniformity (PRNU) is unique to each camera sensor and is widely used as a sensor “fingerprint” (Dirik & Karaküçük, 2014; Banerjee & Ross, 2019), and motion blur can be tied to motion data and hardware (Park & Levoy, 2014; Rong et al., 2024). Therefore, transmitting raw images and kernels may raise significant privacy risks. Moreover, increasingly strict privacy regulations, such as the European Union’s General Data Protection Regulation (GDPR), China’s Personal Information Protection Law (PIPL), and the EU–US Data Privacy Framework (DPF), legally restrict the arbitrary collection and centralized processing of data.

Federated learning (FL) and federated optimization (McMahan et al., 2017) enable collaborative optimization without sharing raw data, protecting privacy while leveraging diverse sources. This paradigm has already been applied to deep learning (Li et al., 2020a; Banabilah et al., 2022; Huang et al., 2024), low-rank matrix approximation (Nayer & Vaswani, 2022; Abbasi & Vaswani, 2025), and singular value decomposition (SVD) (Chai et al., 2024; Guo et al., 2024). However, to the best of our knowledge, research on multi-observation deblurring has not yet adopted FL principles to achieve high-quality restoration without sharing raw images or kernels. Furthermore, a truly practical solution must also be able to effectively integrate the incomplete information from various participants’ partial observations—the “information jigsaw puzzle”—while preserving privacy.

Under the paradigm of federated optimization, we propose an iterative framework for privacy-preserving multi-observation deblurring (Figure 1). Multiple independent *clients* acquire data and communicate only with a central *server*. Clients use private observations to update *local variables* that remain on device; only carefully designed *shared variables* are uploaded for secure aggregation and a global update. With a modular and decoupled design, our framework provides closed-form updates that require no inner iterations, enabling efficient use on edge devices. Our framework strictly adheres to federated principles: (1) no raw observations or kernels are uploaded; (2) clients do not

transmit local clear-image estimates; (3) client updates are lightweight; and (4) in partial observation settings, clients do not reveal their observation regions. Our main contributions are as follows:

- We introduce and formalize the novel problem of federated multi-observation deblurring. In this setting, clients possess private and partially overlapping views of a scene, forming an “information jigsaw puzzle” that must be solved under strict privacy constraints.
- We propose FedDeblur, a privacy-preserving consensus framework based on federated optimization. It aggregates fragmented information to reconstruct a high-quality global image without sharing raw images, blur kernels, or intermediate image estimates.
- Our framework is highly modular, allowing seamless adaptation to different observation models (e.g., complete and partial observations) without modifying federated communication protocol. It further enables the server to flexibly integrate diverse image priors—from classic total variation to modern deep plug-and-play models—in a manner that remains transparent to the clients.
- We conduct extensive experiments demonstrating that FedDeblur achieves restoration quality close to an idealized centralized oracle, and significantly outperforms both non-collaborative and naive federated baselines, particularly under partial observation settings.

2 PROBLEM SETUP

We follow the general setting of federated optimization, adopting a distributed architecture with a central *server* and n *clients* (Ye et al., 2023; Huang et al., 2024). A *client* (indexed by k) is an independent device (e.g., camera, mobile device, or other edge systems) that holds its private blur kernel \mathbf{h}_k and blurry observation \mathbf{y}_k . The *server* coordinates computation without accessing client’s raw data: it aggregates privacy-preserving summaries from clients, updates a global model, and broadcasts it for the next round.

To facilitate the discussion, we distinguish two types of variables. *Local variables* are computed and stored on device; *shared variables* are intermediate quantities exchanged with the server and designed not to leak sensitive data. Based on privacy considerations and the computational constraints of edge devices, we establish the following federated collaboration principles:

- (1) **Data privacy:** Clients never upload raw observations \mathbf{y}_k or kernels \mathbf{h}_k , which may contain sensitive information such as device fingerprints or user behavioral patterns.
- (2) **Intermediate privacy:** Clients do not upload local clear-image estimates during iterations, avoiding inference of raw data from image content.
- (3) **Computational efficiency:** Local updates are lightweight and simple to implement on resource-constrained edge devices.
- (4) **Field-of-view privacy:** In partial observation, clients do not reveal their observed regions as this could reveal sensitive user geolocation data.

We model the multi-observation deblurring problem in two key scenarios.

Partial observation. In a realistic federated scenario, the observation regions of different clients often do not fully overlap and may only intersect slightly (as shown in Figure 1). In this case, we model $\mathbf{x} \in \mathbb{R}^d$ as a “large” latent image covering all observation areas, and each client’s observation \mathbf{y}_k corresponds to a local patch of \mathbf{x} . To formalize this, we introduce a *positional mask* $\mathbf{m}_k \in \{0, 1\}^d$, known only to client k , with ones at its observed pixel locations. This mask embeds the local observation \mathbf{y}_k into the full space \mathbb{R}^d . The observation model for each client is

$$\mathbf{y}_k = \mathbf{m}_k \odot (\mathbf{h}_k * \mathbf{x}) + \varepsilon_k, \quad k = 1, 2, \dots, n,$$

where \mathbf{h}_k and ε_k are the blur kernel and additive noise for client k , respectively, and \odot denotes the element-wise (Hadamard) product. For notational simplicity, we vectorize images, and by an abuse of notation, $*$ denotes 2D convolution with a vectorized input and output. Following the standard modeling approach with data fidelity and regularization terms (Stuart, 2010), we can get

$$\min_{\mathbf{x}} \frac{1}{n} \sum_{k=1}^n \|\mathbf{m}_k \odot (\mathbf{h}_k * \mathbf{x} - \mathbf{y}_k)\|_2^2 + \eta R(\mathbf{x}), \quad (1)$$

where $\eta > 0$ balances the fidelity term $F(\mathbf{x})$ with the regularization term $R(\mathbf{x})$. Note that \mathbf{m}_k ensures the loss is computed only on the pixels observed by each client. As private local information, \mathbf{m}_k should never be uploaded to the server to satisfy our “field-of-view privacy” principle.

Complete observation. This setting follows the basic model used in most existing multi-image deblurring literature. We point out that this scenario, which assumes all clients observe the entire scene, can be treated as a special case of our partial observation model. By setting each positional mask \mathbf{m}_k to be a vector of all ones, model (1) reduces to the complete observation model

$$\min_{\mathbf{x}} \frac{1}{n} \sum_{k=1}^n \|\mathbf{h}_k * \mathbf{x} - \mathbf{y}_k\|_2^2 + \eta R(\mathbf{x}). \quad (2)$$

3 PROPOSED METHOD

To address challenges posed by federated privacy and partial observation scenarios, we present FedDeblur, a federated deblurring framework built on the alternating direction method of multipliers (ADMM) (Boyd et al., 2011). Variable splitting and decoupling align with the “local computation—global aggregation” paradigm and provide a modular basis for more complex scenarios while respecting federated privacy constraints. We first develop the core framework under the complete observation setting, then show that minor modifications extend it to more complex scenarios, including partial observation and transform-domain regularization.

3.1 FEDDEBLUR ALGORITHM FRAMEWORK

We base our approach on consensus and federated ADMM (Shi et al., 2014; Tran Dinh et al., 2021) and adapt it to our federated deblurring setting. The objective in (2) is a sum of fidelity terms and a regularizer on a shared variable. We introduce a *local estimate* \mathbf{x}_k for each client k —a *local variable* updated on device from private data $\{\mathbf{y}_k, \mathbf{h}_k\}$. A server-side *global estimate* \mathbf{x} enforces the consensus constraint $\mathbf{x}_k = \mathbf{x}$ and aggregates information across clients.

This approach yields the original problem (2) as the following constrained optimization problem:

$$\min_{\mathbf{x}, \{\mathbf{x}_k\}} \frac{1}{n} \sum_{k=1}^n \|\mathbf{h}_k * \mathbf{x}_k - \mathbf{y}_k\|_2^2 + \eta R(\mathbf{x}) \quad \text{s.t. } \mathbf{x}_k = \mathbf{x}, k = 1, 2, \dots, n. \quad (3)$$

This decomposition separates the data-fidelity terms (depending only on each local variable \mathbf{x}_k) from regularization (depending only on global variable \mathbf{x}). Its augmented Lagrangian is

$$\mathcal{L}(\mathbf{x}, \{\mathbf{x}_k\}, \{\boldsymbol{\lambda}_k\}) = \sum_{k=1}^n L_k(\mathbf{x}, \mathbf{x}_k, \boldsymbol{\lambda}_k) + \eta R(\mathbf{x}),$$

where

$$L_k(\mathbf{x}, \mathbf{x}_k, \boldsymbol{\lambda}_k) = \frac{1}{n} \|\mathbf{h}_k * \mathbf{x}_k - \mathbf{y}_k\|_2^2 + \langle \boldsymbol{\lambda}_k, \mathbf{x}_k - \mathbf{x} \rangle + \frac{\rho}{2} \|\mathbf{x}_k - \mathbf{x}\|_2^2. \quad (4)$$

Here, $\boldsymbol{\lambda}_k$ is the *local dual variable* associated with the consensus constraint, and ρ is the penalty parameter. The term L_k is the *local objective* for client k : it balances data fidelity error based on the client’s private data with global consensus via the dual and quadratic terms.

Within the ADMM framework, server and clients alternate minimizing \mathcal{L} : the server updates the global variable \mathbf{x} and clients in parallel update \mathbf{x}_k followed by dual ascent on $\boldsymbol{\lambda}_k$. The overall procedure appears in Algorithm 1 (base option 1 and A), alongside the system schematic in Figure 1.

Server global update. The server updates the global estimate by minimizing the global objective \mathcal{L} over \mathbf{x} . By completing the square, this update is a proximal operator

$$\mathbf{x}^{t+1} = \text{Prox}_{\frac{\eta}{n\rho}R} \left(\frac{1}{n} \sum_{k=1}^n \left(\mathbf{x}_k^t + \frac{\boldsymbol{\lambda}_k^t}{\rho} \right) \right). \quad (5)$$

This is central to our federated design: the server updates \mathbf{x} without accessing clients’ private data, aggregating only a carefully designed *shared variable*

$$\mathbf{w}_k^t = \mathbf{x}_k^t + \frac{\boldsymbol{\lambda}_k^t}{\rho}.$$

This shared variable \mathbf{w}_k^t combines the local estimate and dual term, obscuring raw information and adhering to the “intermediate privacy” principle.

Algorithm 1 FedDeblur: base, transform-domain regularization and partial observation

Require: Communication rounds T , regularization parameter η , ADMM parameter ρ

- 1: **Server:** Initialize $\mathbf{x}^0, \mathbf{z}^0$ and $\boldsymbol{\mu}^0$
- 2: **Clients** $k = 1, \dots, n$: Initialize $\mathbf{x}_k^0, \boldsymbol{\lambda}_k^0, \mathbf{r}_k^0$ and $\boldsymbol{\beta}_k^0$; compute \mathbf{w}_k^0 and upload to server
- 3: **for** $t = 0, 1, \dots, T - 1$ **do**
- 4: **server do**
- 5: $\mathbf{w}^t \leftarrow \frac{1}{n} \sum_{k=1}^n \mathbf{w}_k^t$ ▷ Server aggregate
- 6: // Option 1: Base: Image-domain regularization
- 7: $\mathbf{x}^{t+1} \leftarrow \text{PROX}_{\eta R/n\rho}(\mathbf{w}^t)$
- 8: // Option 2: Transform-domain regularization
- 9: $\mathbf{z}^{t+1} \leftarrow \text{PROX}_{\eta R/\rho}(\mathbf{W}\mathbf{x}^t - \boldsymbol{\mu}^t/\rho)$
- 10: $\mathbf{x}^{t+1} \leftarrow (n\mathbf{I} + \mathbf{W}^T\mathbf{W})^{-1}(n\mathbf{w}^t + \mathbf{W}^T(\mathbf{z}^{t+1} + \boldsymbol{\mu}^t/\rho))$
- 11: $\boldsymbol{\mu}^{t+1} \leftarrow \boldsymbol{\mu}^t + \rho(\mathbf{z}^{t+1} - \mathbf{W}\mathbf{x}^{t+1})$
- 12: Broadcast \mathbf{x}^{t+1} to clients
- 13: **for each client** $k = 1, \dots, n$ **in parallel do**
- 14: // Option A: Base: Complete observation
- 15: $\mathbf{x}_k^{t+1} \leftarrow \mathcal{F}^{-1}\left(\left(\frac{2}{n}\overline{\mathcal{F}(\mathbf{h}_k)} \odot \mathcal{F}(\mathbf{y}_k) + \mathcal{F}(\rho\mathbf{x}^{t+1} - \boldsymbol{\lambda}_k^t)\right) / \left(\frac{2}{n}|\mathcal{F}(\mathbf{h}_k)|^2 + \rho\right)\right)$
- 16: // Option B: Partial observation
- 17: $\mathbf{r}_k^{t+1} \leftarrow (\rho(\mathbf{h}_k * \mathbf{x}_k^t - \mathbf{y}_k) - \boldsymbol{\beta}_k^t) / \left(\frac{2}{n}\mathbf{m}_k + \rho\right)$
- 18: $\mathbf{x}_k^{t+1} \leftarrow \mathcal{F}^{-1}\left(\left(\overline{\mathcal{F}(\mathbf{h}_k)} \odot \mathcal{F}(\mathbf{r}_k^{t+1} + \mathbf{y}_k + \boldsymbol{\beta}_k^t/\rho) + \mathcal{F}(\mathbf{x}^{t+1} - \boldsymbol{\lambda}_k^t/\rho)\right) / (|\mathcal{F}(\mathbf{h}_k)|^2 + 1)\right)$
- 19: $\boldsymbol{\beta}_k^{t+1} \leftarrow \boldsymbol{\beta}_k^t + \rho(\mathbf{r}_k^{t+1} - (\mathbf{h}_k * \mathbf{x}_k^{t+1} - \mathbf{y}_k))$
- 20: $\boldsymbol{\lambda}_k^{t+1} \leftarrow \boldsymbol{\lambda}_k^t + \rho(\mathbf{x}_k^{t+1} - \mathbf{x}^{t+1})$
- 21: $\mathbf{w}_k^{t+1} \leftarrow \mathbf{x}_k^{t+1} + \boldsymbol{\lambda}_k^{t+1}/\rho$ and upload to server ▷ Shared variable
- 22: **end for**
- 23: **end for**
- 24: **return** global estimate \mathbf{x}^T

Note that the server-side update (5) is a proximal step on regularization term $R(\mathbf{x})$, the framework naturally accommodates plug-and-play (PnP) priors (Laroche et al., 2023a;b; Wei et al., 2024) by replacing the proximal operator with a pre-trained denoiser $\mathcal{D}_{\eta/n\rho}(\cdot)$, which opens up promising directions for combining our federated framework with state-of-the-art learned priors.

Client parallel local updates. Given \mathbf{x}^{t+1} , each client k minimizes its local objective L_k with respect to \mathbf{x}_k in parallel. And this quadratic subproblem admits an efficient closed-form solution in the Fourier domain, which meets our “computational efficiency” requirement for edge devices:

$$\begin{aligned} \mathbf{x}_k^{t+1} &= \arg \min_{\mathbf{x}_k} L_k(\mathbf{x}^{t+1}, \mathbf{x}_k, \boldsymbol{\lambda}_k^t) \\ &= \mathcal{F}^{-1} \left(\frac{\frac{2}{n}\overline{\mathcal{F}(\mathbf{h}_k)} \odot \mathcal{F}(\mathbf{y}_k) + \mathcal{F}(\rho\mathbf{x}^{t+1} - \boldsymbol{\lambda}_k^t)}{\frac{2}{n}|\mathcal{F}(\mathbf{h}_k)|^2 + \rho} \right). \end{aligned} \quad (6)$$

Each client k then performs dual ascent on $\boldsymbol{\lambda}_k$, forms the shared variable $\mathbf{w}_k^{t+1} = \mathbf{x}_k^{t+1} + \boldsymbol{\lambda}_k^{t+1}/\rho$, and uploads it.

In each round (see the base options in Algorithm 1), every client solves its private local objective L_k (4) to enforce both data fidelity and consensus, then uploads only the desensitized shared variable \mathbf{w}_k^t . Thus raw data never leave devices, preserving privacy. The server aggregates \mathbf{w}_k^t and performs the proximal prior update (5). Centralizing the prior is appropriate because the scene prior is common across clients and the server’s greater computational resources accommodate demanding regularizers (e.g., deep learning-based PnP) without burdening edge devices. This separation of roles—local data fidelity, global prior—produces an efficient design with closed-form local steps and no inner iterations, strictly satisfying all four principles in section 2.

Theorem 1. *If R is proper, lower semi-continuous, convex, and bounded below, then for a sufficiently large ρ , the sequence of iterates generated by FedDeblur algorithm converges to a Karush-Kuhn-Tucker (KKT) point of (3). The detailed analysis is provided in appendix E.*

3.2 HANDLING PARTIAL OBSERVATIONS

We next address the more challenging problem of partial observations as formulated in (1). The private positional masks \mathbf{m}_k in this scenario complicate the data fidelity term and break the Fourier-domain diagonalization of convolution, making the \mathbf{x}_k -subproblem in (6) nontrivial.

To address this challenge, we exploit the *client-side* modularity of FedDeblur. Its client-side modifications appear as option B in Algorithm 1, and detailed derivation is provided in appendix C.1. The core federated consensus mechanism remains unchanged; instead, we introduce a *local* auxiliary variable $\mathbf{r}_k = \mathbf{h}_k * \mathbf{x}_k - \mathbf{y}_k$ within each client’s local problem. This variable splitting reformulates (1) as

$$\min_{\mathbf{x}, \{\mathbf{x}_k\}, \{\mathbf{r}_k\}} \frac{1}{n} \sum_{k=1}^n \|\mathbf{m}_k \odot \mathbf{r}_k\|_2^2 + \eta R(\mathbf{x}) \quad \text{s.t. } \mathbf{x}_k = \mathbf{x}, \mathbf{r}_k = \mathbf{h}_k * \mathbf{x}_k - \mathbf{y}_k, k = 1, \dots, n.$$

These additions are purely local, so the complexity stays on the client side; the server update and communication protocol are unchanged and continue to satisfy our federated privacy constraints.

Concretely, the client’s local updates in (6) decompose into two simple closed-form steps by introducing an additional local dual variable β_k :

$$\begin{cases} \mathbf{r}_k^{t+1} = \frac{\rho(\mathbf{h}_k * \mathbf{x}_k^t - \mathbf{y}_k) - \beta_k^t}{\frac{2}{n} \mathbf{m}_k + \rho}, \\ \mathbf{x}_k^{t+1} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(\mathbf{h}_k) \odot \mathcal{F}(\mathbf{r}_k^{t+1} + \mathbf{y}_k + \beta_k^t / \rho) + \mathcal{F}(\mathbf{x}^{t+1} - \lambda_k^t / \rho)}{|\mathcal{F}(\mathbf{h}_k)|^2 + 1} \right). \end{cases}$$

This extension underscores the modularity of our framework. To handle more complex local observation models (i.e., different data fidelity terms), only the client-side solver needs to be adjusted; the federated architecture and protocol remain unchanged, enabling adaptation to heterogeneous client observation models. Moreover, this variant never uploads the positional mask \mathbf{m}_k , thus directly enforcing the declared “field-of-view privacy” principle.

3.3 INCORPORATING TRANSFORM-DOMAIN REGULARIZATION

Natural images are not sparse in the image domain but often admit compact representations under linear transforms (e.g., gradient, wavelet, Fourier) (Simoncelli & Olshausen, 2001). Consequently, many restoration methods regularize in a transform domain (Zuo et al., 2013; You et al., 2019; Chen & Li, 2025). Let \mathbf{W} denote the transform matrix and consider

$$\min_{\mathbf{x}} F(\mathbf{x}) + \eta R(\mathbf{W}\mathbf{x}),$$

where $F(\mathbf{x})$ is the data fidelity term and $R(\cdot)$ is the transform-domain regularizer. For example, when \mathbf{W} is the discrete gradient operator \mathbf{D} and $R(\cdot)$ is the ℓ_1 -norm, the regularization is total variation (TV). Although the proximal operator for $R \circ \mathbf{W}(\mathbf{x})$ (needed in the server update step (5)) is well defined but often lacks a closed form, requiring inner iterations even for simple $R(\cdot)$.

To support such regularizers efficiently, we exploit the *server-side* modularity of our framework. Its server-side modifications are as option 2 in Algorithm 1. We integrate transform regularizers transparently to clients by applying variable splitting at the server. We introduce a *server* auxiliary variable $\mathbf{z} = \mathbf{W}\mathbf{x}$, the clear image in the transform domain. The full derivation of this extension is detailed in appendix C.2.

This change is purely server-side and converts the proximal step (5), which would otherwise require inner iterations, into two simple updates with introducing a new global dual variable $\boldsymbol{\mu}$:

$$\begin{cases} \mathbf{z}^{t+1} = \text{Prox}_{\frac{\eta}{\rho} R}(\mathbf{W}\mathbf{x}^t - \boldsymbol{\mu}^t / \rho), \\ \mathbf{x}^{t+1} = (n\mathbf{I} + \mathbf{W}^T \mathbf{W})^{-1} (n\mathbf{w}^t + \mathbf{W}^T (\mathbf{z}^{t+1} + \boldsymbol{\mu}^t / \rho)). \end{cases} \quad (7)$$

There, (7) decouples the proximal operator of $R \circ \mathbf{W}$ into a simpler one involving only R , such as soft shrinkage for TV regularization (Daubechies et al., 2004). And when \mathbf{W} has a convolutional structure, (8) is efficient in the Fourier domain. For example, for operator \mathbf{D} in TV, it becomes

$$\mathbf{x}^{t+1} = \mathcal{F}^{-1} \left(\frac{\mathcal{F}(n\mathbf{w}^t + \mathbf{D}^T (\mathbf{z}^{t+1} + \boldsymbol{\mu}^t / \rho))}{n + |\mathcal{F}(\mathbf{D})|^2} \right). \quad (9)$$

Crucially, client computation and the communication protocol are unchanged: clients perform the same local updates and upload the same shared variable w_k as before.

Thus, FedDeblur offers a unified, modular framework. In Algorithm 1, blue blocks mark the base framework introduced in section 3.1, green blocks mark the client-side partial observation variant in section 3.2, and red blocks mark the server-side transform regularization variant in section 3.3. Switching among these only replaces the relevant client or server module; the communication pattern and shared variable stay the same. This preserves strict federated principles and keeps computation light (no inner iterations) while still allowing restoration under heterogeneous, partially observed data. Detailed derivations of both extensions appear in appendix C.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Scenarios. We construct experimental scenarios for both two observation settings:

- **Complete observation.** In this setting, we simulate a scenario where all n clients observe the complete 256×256 image. For each client k , the blur kernel is one of motion, Gaussian, disk, or a random synthesis kernel that simulates camera shake.
- **Partial observation.** In this setting, we partition the complete image into a 3×3 grid of nine partially overlapping 100×100 regions, each overlapping adjacent ones by 22 pixels. Each region is the field of view of one of nine clients (see Figure 2a). To model heterogeneous degradations, each client uses an independent blur kernel from one of two families: motion blur with varying angles and lengths (motion) and randomly synthesized camera-shake kernels (synthesis, as shown in Figure 2b, and an example clean scene and its nine observed patches appear in Figure 2c and 2d).

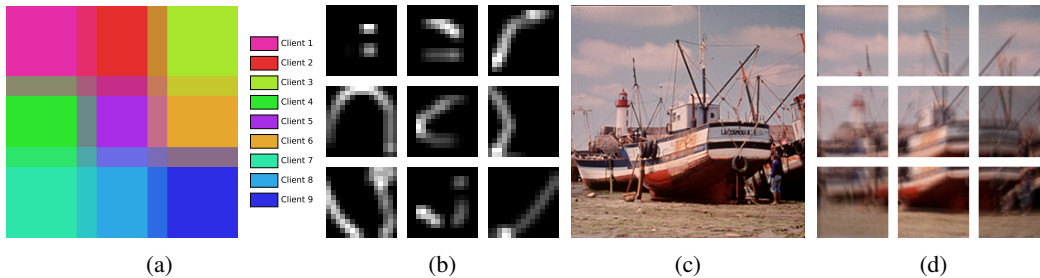


Figure 2: Visualization of the partial observation setup. We simulate 9 clients, each capturing only a partial, overlapping view of the complete scene. (a) Positional masks: each colored region is one client’s field of view. (b) An example set of 9 client-specific blur kernels (synthesis). (c) Example clean scene. (d) The corresponding nine observed patches.

Baselines. Our primary goal is to study the effect of the federated consensus architecture itself. Thus, to ensure a fair comparison, we fix the image prior across compared algorithms. We evaluate two priors: classic total variation (TV) regularization and a plug-and-play (PnP) prior instantiated with the DRUNet denoiser (Zhang et al., 2021a). We establish the following baseline methods:

- **Centralized Deblurring (CenDeblur):** A centralized deblurring method that has direct access to all observation data and masks. It solves the global optimization problem (2) or (1) using a classic ADMM algorithm, analogous to traditional methods like Jung et al. (2013); Yang et al. (2013). This method represents an idealized upper bound on performance, assuming privacy and communication constraints are disregarded.
- **Local Deblurring (LocDeblur):** A fully local method where each client independently solves its own deblurring problem using a classic ADMM algorithm (Goldstein & Osher, 2009; Boyd et al., 2011). This method involves no federated collaboration and thus serves as a performance lower bound. The resulting images are averaged only for evaluation purposes.
- **Federated Averaging Deblurring (FedAvgDeblur):** This method is inspired by the federated averaging (FedAvg) algorithm (McMahan et al., 2017). Its local iteration framework is similar to LocDeblur, but after each local ADMM update, all clients upload their local estimates x_k to the server, which averages them and broadcasts the result as the next round’s initialization. Its communication overhead is comparable to that of FedDeblur, but it violates the “intermediate

Table 1: Comparison under TV regularization. Results are presented for complete observation with a varying number of clients ($n = 10, 5, 3$) and partial observation with motion and synthetic blur kernels. For each metric, the best result is highlighted in magenta and the second-best in cyan.

Setup	Type	FedDeblur		CenDeblur		FedAvgDeblur		LocDeblur	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$n = 10$	Gray	40.0935	0.9780	40.1004	0.9780	39.4586	0.9745	33.8958	0.9363
	Color	39.8238	0.9791	39.8283	0.9791	39.1116	0.9755	32.7318	0.9350
$n = 5$	Gray	37.7249	0.9661	37.7295	0.9662	37.3823	0.9646	34.0861	0.9406
	Color	37.4406	0.9675	37.4436	0.9676	37.1558	0.9655	33.2227	0.9370
$n = 3$	Gray	36.3405	0.9571	36.3441	0.9572	36.2079	0.9565	33.3256	0.9365
	Color	35.9711	0.9590	35.9714	0.9591	35.8202	0.9574	32.3429	0.9332
Motion	Gray	33.7048	0.9317	33.7041	0.9317	23.2071	0.7298	23.2771	0.7359
	Color	32.8611	0.9276	32.8699	0.9277	22.5476	0.7129	22.6383	0.7084
Synthesis	Gray	31.4806	0.9121	31.6595	0.9108	21.1035	0.6761	21.5789	0.6656
	Color	30.5747	0.9044	30.7922	0.8995	21.1159	0.6606	21.2517	0.6508

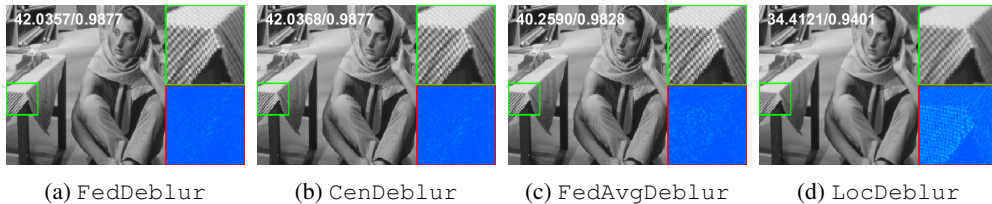


Figure 3: Visual comparison for the complete observation setting with $n = 10$ clients, using the PnP prior. Each image is accompanied by an enlarged patch (top right) and its residual map (bottom right). PSNR and SSIM values are displayed on the top left of each image.

privacy” principle by directly transmitting local image estimates. Furthermore, since each client must independently perform the proximal update, it can be computationally expensive when using deep priors, potentially violating the “computational efficiency” principle.

- **Ours (FedDeblur)**: Our proposed federated deblurring method.

Note that for LocDeblur and FedAvgDeblur in the partial observation setting, the averaging is performed only over the overlapping pixels. As federated algorithms, FedDeblur and FedAvgDeblur initialize the global estimate x^0 randomly, since the server cannot access client data. In contrast, CenDeblur initializes x^0 with the average of all observed blurry images, i.e., $x^0 = \frac{1}{n} \sum_{k=1}^n y_k$. For LocDeblur, each client initializes its estimate with its own blurry image, $x_k^0 = y_k$. Detailed information about these baseline algorithms is provided in appendix C.3.

4.2 EXPERIMENTAL RESULTS

We report quantitative results for TV and PnP regularization in Table 1 and 2. Each value is the mean metric over multiple grayscale and color test images (see appendix D.1), computed from blurred observations corrupted by additive Gaussian white noise (standard deviation 1). Across all settings, FedDeblur achieves performance very close to the idealized centralized method CenDeblur, which serves as an upper bound, with only a small gap. This indicates that our framework aggregates information from multiple clients to produce a high-quality global restoration without accessing raw data. Both FedDeblur and CenDeblur consistently outperform FedAvgDeblur and the non-collaborative baseline LocDeblur. LocDeblur, as the lower bound, yields the worst results, confirming the need to fuse information from multiple observations. The intermediate performance of FedAvgDeblur shows that while simple collaboration helps, naive averaging is suboptimal.

In the complete observation setting, the visuals in Figure 3 align with the quantitative results. The collaborative methods—FedDeblur, CenDeblur, and FedAvgDeblur—all recover sharp images with clear edges and fine details, such as the intricate patterns and textures on the tablecloth. In contrast, LocDeblur leaves the image blurry, underscoring its inability to exploit complementary information from different clients.

Table 2: Comparison on grayscale images under PnP prior. The experimental settings mirror those in Table 1. The best result is highlighted in magenta and the second-best in cyan.

Setup	FedDeblur		CenDeblur		FedAvgDeblur		LocDeblur	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$n = 10$	42.1380	0.9850	42.1415	0.9850	40.9416	0.9813	35.0000	0.9503
$n = 5$	39.7744	0.9737	39.7207	0.9759	37.9846	0.9669	35.6389	0.9543
$n = 3$	38.6532	0.9714	38.7273	0.9711	37.3461	0.9618	34.5894	0.9493
motion	34.7655	0.9416	35.3347	0.9458	22.7676	0.7369	22.5329	0.6996
synthesis	31.5206	0.9124	32.3711	0.9164	21.2241	0.6491	20.8160	0.5974

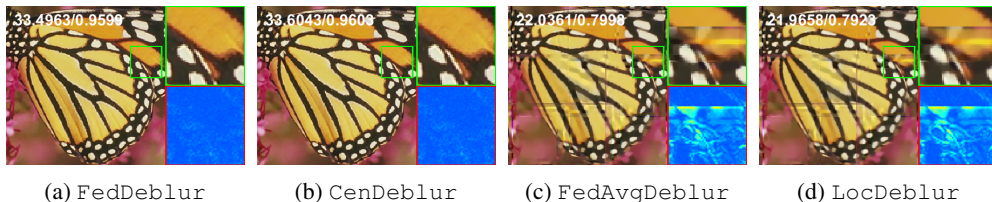


Figure 4: Visual comparison for the partial observation setting with motion blur, using TV regularization. Each image is accompanied by an enlarged patch (top right) and its residual map (bottom right). PSNR and SSIM values are displayed on the top left of each image.

The advantages of our framework are most pronounced in the challenging partial observation setting. In this case, as shown in Table 1 and 2, the performance of FedAvgDeblur and LocDeblur degrades sharply. Figure 4 further explains this failure: both methods create strong seam artifacts in overlapping regions. Moreover, the corresponding residual maps show structured, block-wise errors, indicating that naive averaging or independent processing cannot reconcile inconsistent fields of view and may even introduce conflicting updates. By contrast, FedDeblur and CenDeblur produce seamless, natural-looking images. This demonstrates that our method effectively integrates information from different spatial regions into a unified global estimate, solving the “information jigsaw puzzle” under strict federated privacy constraints.

Finally, our results highlight the framework’s robustness and generality. From Table 1 and 2, increasing the number of clients n from 3 to 10 leads to a consistent improvement for all collaborative methods (FedDeblur, CenDeblur, FedAvgDeblur), since more observations provide complementary constraints that suppress artifacts. Furthermore, although the PnP prior yields better absolute metrics than TV, the performance ranking and relative gaps among the four methods remain stable across priors. Therefore, the performance differences arise from the federated optimization designs rather than the choice of regularizer. Consequently, FedDeblur provides a flexible framework that combines cleanly with diverse image priors, from classic models to learned denoisers.

For a more comprehensive evaluation, we report additional experiments in appendix D. We analyze scalability with respect to the number of clients (appendix D.2), performance under partial client participation (appendix D.3), and robustness to varying levels of observation noise (appendix D.4). We also provide a detailed numerical analysis of convergence behavior (appendix D.5) and sensitivity to hyperparameters (appendix D.8).

5 CONCLUSION

We addressed multi-observation deblurring in a privacy-constrained federated setting where clients each observe only partial, overlapping regions of a scene. We introduced FedDeblur, a federated consensus framework that solves this “information jigsaw puzzle” by aggregating fragmented data without compromising privacy while strictly following federated principles. With a modular, decoupled design separating local data fidelity and the global image prior, FedDeblur handles diverse partial observation regimes and supports a wide range of priors, from classical regularizers to deep learning PnP models. Experiments show that the method achieves high-quality restoration close to an idealized non-private centralized solution. Future work will extend the framework to asynchronous client participation and richer forms of data heterogeneity.

486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539

REPRODUCIBILITY STATEMENT

We are committed to ensuring the reproducibility of our research. The complete source code for our proposed FedDeblur, all baseline methods, and the scripts to replicate our experiments are provided in <https://anonymous.4open.science/r/FedDeblur-53D7>. All experiments use fixed random seeds to ensure deterministic results and were conducted on a workstation with NVIDIA RTX 3090 GPU. The complete mathematical derivations and all proofs are stated in the appendix with explanations and underlying assumptions.

REFERENCES

- Aakanksha and A. N. Rajagopalan. Improving robustness of semantic segmentation to motion-blur using class-centric augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10470–10479, 2023.
- Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjrholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2): 153–168, 2016.
- Ahmed Ali Abbasi and Namrata Vaswani. Efficient federated low rank matrix completion. *IEEE Transactions on Information Theory*, 71(7):5493–5511, 2025.
- Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 126–135, 2017.
- Miika Aittala and Frédo Durand. Burst image deblurring using permutation invariant convolutional neural networks. In *Proceedings of the European Conference on Computer Vision*, pp. 731–747, 2018.
- Syreen Banabilah, Moayad Aloqaily, Eitaa Alsayed, Nida Malik, and Yaser Jararweh. Federated learning review: Fundamentals, enabling technologies, and future applications. *Information Processing & Management*, 59(6):103061, 2022.
- Sudipta Banerjee and Arun Ross. Smartphone camera de-identification while preserving biometric utility. In *Proceedings of the IEEE International Conference on Biometrics: Theory, Applications and Systems*, pp. 1–10, 2019.
- Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.
- Kristian Bredies, Karl Kunisch, and Thomas Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.
- 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 ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 46–55, 2022.
- Di Chai, Junxue Zhang, Liu Yang, Yilun Jin, Leye Wang, Kai Chen, and Qiang Yang. Efficient decentralized federated singular vector decomposition. In *Proceedings of the USENIX Annual Technical Conference*, pp. 1029–1047, 2024.
- Mingzhe Chen, Nir Shlezinger, H Vincent Poor, Yonina C Eldar, and Shuguang Cui. Communication-efficient federated learning. *Proceedings of the National Academy of Sciences*, 118(17):e2024789118, 2021.
- Xuesong Chen and Fang Li. Single-shot phase retrieval via gradient-sparse non-convex regularization integrating physical constraints. *Journal of Scientific Computing*, 102(3):63, 2025.
- Sebastian Dalleiger and Aristides Gionis. Creating coherence in federated non-negative matrix factorization. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(15):16135–16143, 2025.

- 540 Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm
541 for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied*
542 *Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–
543 1457, 2004.
- 544 Mauricio Delbracio and Guillermo Sapiro. Burst deblurring: Removing camera shake through
545 Fourier burst accumulation. In *Proceedings of the IEEE Conference on Computer Vision and*
546 *Pattern Recognition*, pp. 2385–2393, 2015.
- 547 Ahmet Emir Dirik and Ahmet Karaküçük. Forensic use of photo response non-uniformity of imaging
548 sensors and a counter method. *Optics Express*, 22(1):470–482, 2014.
- 549 Axel Donath, Aneta Siemiginowska, Vinay L Kashyap, David A van Dyk, and Douglas Burke.
550 Joint deconvolution of astronomical images in the presence of Poisson noise. *The Astronomical*
551 *Journal*, 168(4):182, 2024.
- 552 Alp Emre Durmus, Zhao Yue, Matas Ramon, Mattina Matthew, Whatmough Paul, and Saligrama
553 Venkatesh. Federated learning based on dynamic regularization. In *International Conference on*
554 *Learning Representations*, 2021.
- 555 Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced
556 correlation coefficient maximization. *IEEE Transactions on Pattern Analysis and Machine Intel-*
557 *ligence*, 30(10):1858–1865, 2008.
- 558 Esmail Faramarzi, Dinesh Rajan, and Marc P Christensen. Unified blind method for multi-image
559 super-resolution and single/multi-image blur deconvolution. *IEEE Transactions on Image Pro-*
560 *cessing*, 22(6):2101–2114, 2013.
- 561 Daniel Feijoo, Juan C Benito, Alvaro Garcia, and Marcos V Conde. DarkIR: Robust low-light
562 image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*
563 *Recognition*, pp. 10879–10889, 2025.
- 564 Tom Goldstein and Stanley Osher. The split Bregman method for L1-regularized problems. *SIAM*
565 *Journal on Imaging Sciences*, 2(2):323–343, 2009.
- 566 Yonghai Gong, Yichuan Li, and Nikolaos M Freris. FedADMM: A robust federated deep learning
567 framework with adaptivity to system heterogeneity. In *Proceedings of the IEEE International*
568 *Conference on Data Engineering*, pp. 2575–2587, 2022.
- 569 Andreas Grammenos, Rodrigo Mendoza Smith, Jon Crowcroft, and Cecilia Mascolo. Federated
570 principal component analysis. In *Advances in Neural Information Processing Systems*, volume 33,
571 pp. 6453–6464, 2020.
- 572 Rémi Gribonval and Mila Nikolova. A characterization of proximity operators. *Journal of Mathe-*
573 *matical Imaging and Vision*, 62(6):773–789, 2020.
- 574 Xiao Guo, Xiang Li, Xiangyu Chang, Shusen Wang, and Zhihua Zhang. FedPower: Privacy-
575 preserving distributed eigenspace estimation. *Machine Learning*, 113(11):8427–8458, 2024.
- 576 Anne Hartebrodt, Richard Röttger, and David B Blumenthal. Federated singular value decomposi-
577 tion for high-dimensional data. *Data Mining and Knowledge Discovery*, 38(3):938–975, 2024.
- 578 Wenke Huang, Mang Ye, Zekun Shi, Guancheng Wan, He Li, Bo Du, and Qiang Yang. Federated
579 learning for generalization, robustness, fairness: A survey and benchmark. *IEEE Transactions on*
580 *Pattern Analysis and Machine Intelligence*, 46(12):9387–9406, 2024.
- 581 Miyoun Jung, Antonio Marquina, and Luminita A Vese. Variational multiframe restoration of im-
582 ages degraded by noisy (stochastic) blur kernels. *Journal of Computational and Applied Mathe-*
583 *matics*, 240:123–134, 2013.
- 584 Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and
585 Ananda Theertha Suresh. SCAFFOLD: Stochastic controlled averaging for federated learning.
586 In *Proceedings of the International Conference on Machine Learning*, pp. 5132–5143, 2020.

- 594 Charles Laroche, Andrés Almansa, Eva Coupeté, and Matias Tassano. Provably convergent plug-
595 and-play linearized ADMM, applied to deblurring spatially varying kernels. In *Proceedings of*
596 *the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5, 2023a.
597
- 598 Charles Laroche, Andrés Almansa, and Matias Tassano. Deep model-based super-resolution with
599 non-uniform blur. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Com-*
600 *puter Vision*, pp. 1797–1808, 2023b.
- 601 Li Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. A review of applications in federated learning. *Com-*
602 *puters & Industrial Engineering*, 149:106854, 2020a.
603
- 604 Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith.
605 Federated optimization in heterogeneous networks. In *Proceedings of Machine Learning and*
606 *Systems*, volume 2, pp. 429–450, 2020b.
- 607 Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas.
608 Communication-efficient learning of deep networks from decentralized data. In *Proceedings of*
609 *the International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, 2017.
610
- 611 Seyedehsara Nayer and Namrata Vaswani. Fast and sample-efficient federated low rank matrix
612 recovery from column-wise linear and quadratic projections. *IEEE Transactions on Information*
613 *Theory*, 69(2):1177–1202, 2022.
- 614 Jihyong Oh and Munchurl Kim. DeMFI: Deep joint deblurring and multi-frame interpolation with
615 flow-guided attentive correlation and recursive boosting. In *Proceedings of the European Confer-*
616 *ence on Computer Vision*, pp. 198–215, 2022.
617
- 618 Sung Hee Park and Marc Levoy. Gyro-based multi-image deconvolution for removing handshake
619 blur. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.
620 3366–3373, 2014.
- 621 Fidel Alejandro Guerrero Peña, Pedro Diamel Marrero Fernández, Tsang Ing Ren, Jorge de Je-
622 sus Gomes Leandro, and Ricardo Massahiro Nishihara. Burst ranking for blind multi-image
623 deblurring. *IEEE Transactions on Image Processing*, 29:947–958, 2019.
624
- 625 Jianxiang Rong, Hua Huang, and Jia Li. IMU-assisted accurate blur kernel re-estimation in non-
626 uniform camera shake deblurring. *IEEE Transactions on Image Processing*, 33:3823–3838, 2024.
- 627 Yamin Ru, Fang Li, Faming Fang, and Guixu Zhang. Patch-based weighted SCAD prior for com-
628 pressive sensing. *Information Sciences*, 592:137–155, 2022.
629
- 630 Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal
631 algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.
- 632 Wei Shi, Qing Ling, Kun Yuan, Gang Wu, and Wotao Yin. On the linear convergence of the ADMM
633 in decentralized consensus optimization. *IEEE Transactions on Signal Processing*, 62(7):1750–
634 1761, 2014.
635
- 636 Eero P Simoncelli and Bruno A Olshausen. Natural image statistics and neural representation.
637 *Annual Review of Neuroscience*, 24(1):1193–1216, 2001.
- 638 Filip Sroubek and Peyman Milanfar. Robust multichannel blind deconvolution via fast alternating
639 minimization. *IEEE Transactions on Image Processing*, 21(4):1687–1700, 2011.
640
- 641 Andrew M Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.
642
- 643 Quoc Tran Dinh, Nhan H Pham, Dzung Phan, and Lam Nguyen. FedDR—randomized douglas-
644 rachford splitting algorithms for nonconvex federated composite optimization. In *Advances in*
645 *Neural Information Processing Systems*, volume 34, pp. 30326–30338, 2021.
- 646 Singanallur V Venkatakrishnan, Charles A Bouman, and Brendt Wohlberg. Plug-and-play priors
647 for model based reconstruction. In *Proceedings of the IEEE Global Conference on Signal and*
Information Processing, pp. 945–948, 2013.

- 648 Han Wang, Siddhartha Marella, and James Anderson. FedADMM: A federated primal-dual algorithm
649 allowing partial participation. In *Proceedings of the IEEE Conference on Decision and Control*,
650 pp. 287–294, 2022.
- 651 Deliang Wei, Peng Chen, and Fang Li. Learning pseudo-contractive denoisers for inverse problems.
652 In *Proceedings of the International Conference on Machine Learning*, pp. 52500–52524, 2024.
- 653 Deliang Wei, Peng Chen, Haobo Xu, Jiale Yao, Fang Li, and Tiejong Zeng. Learning cocoercive
654 conservative denoisers via Helmholtz decomposition for Poisson inverse problems. *arXiv preprint*
655 *arXiv:2505.08909*, 2025.
- 656 Patrick Wieschollek, Michael Hirsch, Bernhard Schölkopf, and Hendrik Lensch. Learning blind
657 motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision*,
658 pp. 231–240, 2017.
- 659 Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via L0 gradient minimization. In *Proceed-*
660 *ings of the ACM SIGGRAPH Asia Conference*, pp. 1–12, 2011.
- 661 Sen Yang, Jie Wang, Wei Fan, Xiatian Zhang, Peter Wonka, and Jieping Ye. An efficient ADMM
662 algorithm for multidimensional anisotropic total variation regularization problems. In *Proceed-*
663 *ings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*,
664 pp. 641–649, 2013.
- 665 Mang Ye, Xiuwen Fang, Bo Du, Pong C Yuen, and Dacheng Tao. Heterogeneous federated learning:
666 State-of-the-art and research challenges. *ACM Computing Surveys*, 56(3):1–44, 2023.
- 667 Juntao You, Yuling Jiao, Xiliang Lu, and Tiejong Zeng. A nonconvex model with minimax concave
668 penalty for image restoration. *Journal of Scientific Computing*, 78(2):1063–1086, 2019.
- 669 Haichao Zhang, David Wipf, and Yanning Zhang. Multi-image blind deblurring using a coupled
670 adaptive sparse prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern*
671 *Recognition*, pp. 1051–1058, 2013.
- 672 Haichao Zhang, David Wipf, and Yanning Zhang. Multi-observation blind deconvolution with an
673 adaptive sparse prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):
674 1628–1643, 2014.
- 675 Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play
676 image restoration with deep denoiser prior. *IEEE Transactions on Pattern Analysis and Machine*
677 *Intelligence*, 44(10):6360–6376, 2021a.
- 678 Kaihao Zhang, Wenqi Ren, Wenhan Luo, Wei-Sheng Lai, Björn Stenger, Ming-Hsuan Yang, and
679 Hongdong Li. Deep image deblurring: A survey. *International Journal of Computer Vision*, 130
680 (9):2103–2130, 2022.
- 681 Xinwei Zhang, Mingyi Hong, Sairaj Dhople, Wotao Yin, and Yang Liu. FedPD: A federated learning
682 framework with adaptivity to non-iid data. *IEEE Transactions on Signal Processing*, 69:6055–
683 6070, 2021b.
- 684 Hao Zhou, Yueting Chen, Huajun Feng, Guomian Lv, Zhihai Xu, and Qi Li. Rotated rectangular
685 aperture imaging through multi-frame blind deconvolution with hyper-Laplacian priors. *Optics*
686 *Express*, 29(8):12145–12159, 2021.
- 687 Shangchen Zhou, Jiawei Zhang, Wangmeng Zuo, Haozhe Xie, Jinshan Pan, and Jimmy S Ren.
688 DavaNet: Stereo deblurring with view aggregation. In *Proceedings of the IEEE/CVF Conference*
689 *on Computer Vision and Pattern Recognition*, pp. 10996–11005, 2019.
- 690 Shenglong Zhou and Geoffrey Ye Li. Federated learning via inexact ADMM. *IEEE Transactions*
691 *on Pattern Analysis and Machine Intelligence*, 45(8):9699–9708, 2023.
- 692 Wangmeng Zuo, Deyu Meng, Lei Zhang, Xiangchu Feng, and David Zhang. A generalized iterated
693 shrinkage algorithm for non-convex sparse coding. In *Proceedings of the IEEE International*
694 *Conference on Computer Vision*, pp. 217–224, 2013.
- 695
696
697
698
699
700
701

A RELATED WORK

A.1 MULTI-IMAGE DEBLURRING

Existing research in multi-image deblurring underpins our problem formulation but has largely assumed centralized access to data. These methods can be broadly divided into model-based iterative optimization and end-to-end deep learning approaches.

Within the framework of model-based iterative optimization, Sroubek & Milanfar (2011) first proposed a robust multi-image deblurring algorithm based on fast alternating minimization. Faramarzi et al. (2013) integrated multi-image super-resolution and deblurring into a unified framework. Zhang et al. (2013; 2014) introduced coupled adaptive sparse priors, which automatically assign weights to each observation based on its intrinsic quality. Zhou et al. (2021) applied multi-frame deconvolution with a hyper-Laplacian prior to rotated rectangular aperture imaging. In contrast to these optimization-based methods, Delbracio & Sapiro (2015) proposed a method that directly performs weighted information fusion in the Fourier domain. However, these iterative methods are computationally expensive and can be limited by handcrafted priors, which may adapt poorly to complex scenes. Our work retains the modeling strengths of these methods but targets deployment on edge devices: all client updates are closed-form, require no inner iterations, and the design remains compatible with diverse priors.

Deep learning methods learn an end-to-end mapping from multiple blurry inputs to a single clear image. Wieschollek et al. (2017) used a recurrent neural network (RNN) to process image bursts by exploiting temporal information. Aittala & Durand (2018) proposed a permutation-invariant convolutional neural network (CNN) architecture capable of processing an arbitrary number of input images in a sequence-agnostic manner. Peña et al. (2019) employed a ranking network to pre-select the sharpest frames from a burst before fusion to improve deblurring performance. Furthermore, Zhou et al. (2019) designed DAVANet for information aggregation in stereo cameras, while Oh & Kim (2022) introduced DeMFI for video data, which jointly performs multi-frame video deblurring and interpolation within a unified framework. These approaches, however, rely on large paired datasets and often have heavy models, making deployment on heterogeneous edge devices difficult. By contrast, our iterative framework does not require large-scale training data and, when available, can seamlessly integrate learned PnP priors.

A.2 FEDERATED OPTIMIZATION

Our work builds on principles of federated optimization to address privacy. The federated averaging (FedAvg) algorithm of McMahan et al. (2017) is the foundational work in this field. It significantly improves communication efficiency by allowing clients to perform multiple steps of local stochastic gradient descent (SGD) in each communication round. However, in typical federated learning scenarios, data on client devices is typically non-independent and identically distributed (non-IID), leading to what is known as “data heterogeneity” and the “client drift” problem (Ye et al., 2023).

To address this core challenge, researchers have proposed a range of federated optimization algorithms. One line of research modifies the local objective function to constrain local updates and prevent them from deviating too far from the global model. For example, Li et al. (2020b) proposed FedProx, which adds a proximal term to the local loss function to limit the magnitude of local updates. The FedDyn algorithm by Durmus et al. (2021) employs a dynamic regularizer, ensuring that the minimizers of the local objectives asymptotically align with the stationary points of the global loss. In contrast to these methods, SCAFFOLD, proposed by Karimireddy et al. (2020), focuses on directly correcting the direction of the local update gradient.

Another major class of methods leverages operator splitting techniques to provide a more structured mathematical framework for federated optimization problems. Our proposed FedDeblur framework is built upon this line of work, leveraging the consensus form of ADMM, which is a natural fit for the “local computation-global aggregation” paradigm of federated optimization. Zhang et al. (2021b) first proposed the FedPD algorithm under this framework, followed by several FedADMM algorithms from Gong et al. (2022) and Zhou & Li (2023). Additionally, Tran Dinh et al. (2021) proposed the FedDR algorithm based on Douglas-Rachford splitting, which is equivalent to ADMM under certain conditions (Wang et al., 2022). Operator-splitting frameworks offer strong decoupling

and handle non-smooth terms well, making them well-suited to our modular design that separates local data fitting from global regularization.

While initially designed for training deep learning models in federated learning, recent research has successfully extended the federated paradigm to a range of classic iterative optimization algorithms, offering distributed solutions that balance privacy and computational efficiency. Grammenos et al. (2020) designed a differentially private federated PCA algorithm for asynchronous and memory-constrained environments. In the field of low-rank matrix approximation, a strategy of decomposing the target matrix into local and shared factors to decouple local and global updates has been successfully applied to problems like low-rank matrix recovery (Nayer & Vaswani, 2022) and matrix completion (Abbasi & Vaswani, 2025). For the singular value decomposition (SVD) problem, Guo et al. (2024) extended the classic power method to the federated setting. Differing from this iterative approach, Chai et al. (2022; 2024) focused on achieving an exact federated SVD. Furthermore, for high-dimensional data scenarios such as genomics, Hartebrodt et al. (2024) designed a federated SVD algorithm with communication costs independent of the sample size. For non-negative matrix factorization (NMF), Dalleiger & Gionis (2025) proposed a solution based on “creating coherence”. Our work develops a federated framework for an image inverse problem—multi-observation deblurring—highlighting the breadth of applications for federated optimization.

B NOTATIONS

We summarize the notation used in this paper. Plain lowercase letters denote scalars; bold lowercase letters denote vectors; and capital letters denote matrices and linear operators. Images and related variables are treated as vectors in the d -dimensional real Euclidean space \mathbb{R}^d with inner product $\langle \cdot, \cdot \rangle$. The ℓ_2 -norm is denoted by $\|\cdot\|_2$. Two-dimensional convolution is denoted by $*$, and the 2D discrete Fourier transform and its inverse are $\mathcal{F}(\cdot)$ and $\mathcal{F}^{-1}(\cdot)$, respectively. For these 2D operators, although inputs and outputs are written as vectors, we apply the operations to their two-dimensional reshaped forms and then vectorize the results. Element-wise operations include the Hadamard product (\odot), division, exponentiation (for any vector \mathbf{v} and scalar p , \mathbf{v}^p denotes element-wise power), complex conjugation (denoted by an overline, e.g., \bar{z}), and modulus for complex vectors (for z , $|z|$ and $|z|^2$ denote the element-wise modulus and squared modulus). The transpose of a linear operator \mathbf{A} is \mathbf{A}^T , and I denotes the identity matrix. A superscript t (e.g., \mathbf{x}^t) indicates the iteration index in our algorithms.

C MORE ALGORITHMIC DETAILS

This section provides further details on the extensions of the FedDeblur framework, including partial observations and transform-domain regularization. We begin by deriving these extensions and presenting an integrated, unified algorithm. Subsequently, we will elaborate on the implementation details of the baseline algorithms used for comparison in the experiments.

C.1 FEDDEBLUR FOR PARTIAL OBSERVATION

For the partial observation scenario, the objective function is given by

$$\min_{\mathbf{x}} \frac{1}{n} \sum_{k=1}^n \|\mathbf{m}_k \odot (\mathbf{h}_k * \mathbf{x} - \mathbf{y}_k)\|_2^2 + \eta R(\mathbf{x}).$$

As in the base framework, we introduce local variables \mathbf{x}_k and enforce the consensus constraint $\mathbf{x}_k = \mathbf{x}$. The presence of the positional mask \mathbf{m}_k complicates the client subproblem, as it prevents a direct solution in the Fourier domain. To address this, we introduce an additional local auxiliary variable for each client, $\mathbf{r}_k = \mathbf{h}_k * \mathbf{x}_k - \mathbf{y}_k$. The problem can then be equivalently reformulated as

$$\begin{aligned} \min_{\mathbf{x}, \{\mathbf{x}_k\}, \{\mathbf{r}_k\}} \quad & \frac{1}{n} \sum_{k=1}^n \|\mathbf{m}_k \odot \mathbf{r}_k\|_2^2 + \eta R(\mathbf{x}) \\ \text{s.t.} \quad & \mathbf{x}_k = \mathbf{x}, \quad \mathbf{r}_k = \mathbf{h}_k * \mathbf{x}_k - \mathbf{y}_k, \quad k = 1, \dots, n. \end{aligned}$$

The augmented Lagrangian now includes terms for two sets of constraints:

$$\mathcal{L}(\mathbf{x}, \{\mathbf{x}_k\}, \{\mathbf{r}_k\}, \{\boldsymbol{\lambda}_k\}, \{\boldsymbol{\beta}_k\}) = \sum_{k=1}^n L_k(\mathbf{x}, \mathbf{x}_k, \mathbf{r}_k, \boldsymbol{\lambda}_k, \boldsymbol{\beta}_k) + \eta R(\mathbf{x}),$$

$$\begin{aligned} \text{where } L_k(\mathbf{x}, \mathbf{x}_k, \mathbf{r}_k, \boldsymbol{\lambda}_k, \boldsymbol{\beta}_k) &= \frac{1}{n} \|\mathbf{m}_k \odot \mathbf{r}_k\|_2^2 \\ &+ \langle \boldsymbol{\lambda}_k, \mathbf{x}_k - \mathbf{x} \rangle + \frac{\rho}{2} \|\mathbf{x}_k - \mathbf{x}\|_2^2 \\ &+ \langle \boldsymbol{\beta}_k, \mathbf{r}_k - (\mathbf{h}_k * \mathbf{x}_k - \mathbf{y}_k) \rangle + \frac{\rho}{2} \|\mathbf{r}_k - (\mathbf{h}_k * \mathbf{x}_k - \mathbf{y}_k)\|_2^2. \end{aligned} \quad (10)$$

Here, $\boldsymbol{\beta}_k$ is the local dual variable associated with the new constraint. The server update for \mathbf{x} remains unchanged. The client's local update now involves sequentially solving for \mathbf{r}_k and \mathbf{x}_k by minimizing the local objective L_k .

Client's local update. Each client k solves for its local variables by minimizing its local objective L_k :

$$\begin{aligned} \mathbf{r}_k^{t+1} &= \arg \min_{\mathbf{r}_k} L_k(\mathbf{x}^{t+1}, \mathbf{x}_k^t, \mathbf{r}_k, \boldsymbol{\lambda}_k^t, \boldsymbol{\beta}_k^t), \\ \mathbf{x}_k^{t+1} &= \arg \min_{\mathbf{x}_k} L_k(\mathbf{x}^{t+1}, \mathbf{x}_k, \mathbf{r}_k^{t+1}, \boldsymbol{\lambda}_k^t, \boldsymbol{\beta}_k^t). \end{aligned}$$

Both subproblems are quadratic optimization problems with closed-form solutions. Specifically,

$$\mathbf{r}_k^{t+1} = \frac{\rho(\mathbf{h}_k * \mathbf{x}_k^t - \mathbf{y}_k) - \boldsymbol{\beta}_k^t}{\frac{2}{n} \mathbf{m}_k^2 + \rho}.$$

Note that since $\mathbf{m}_k \in \{0, 1\}^d$, we have $\mathbf{m}_k^2 = \mathbf{m}_k$. The squaring and division operations here are performed element-wise. Notably, the \mathbf{x}_k -subproblem is a standard deconvolution problem that can be solved efficiently in the Fourier domain:

$$\mathbf{x}_k^{t+1} = \mathcal{F}^{-1} \left(\frac{\overline{\mathcal{F}(\mathbf{h}_k)} \odot \mathcal{F}(\mathbf{r}_k^{t+1} + \mathbf{y}_k + \boldsymbol{\beta}_k^t / \rho) + \mathcal{F}(\mathbf{x}^{t+1} - \boldsymbol{\lambda}_k^t / \rho)}{|\mathcal{F}(\mathbf{h}_k)|^2 + 1} \right).$$

Finally, the dual ascent updates for the local dual variables are standard:

$$\begin{aligned} \boldsymbol{\lambda}_k^{t+1} &= \boldsymbol{\lambda}_k^t + \rho(\mathbf{x}_k^{t+1} - \mathbf{x}^{t+1}), \\ \boldsymbol{\beta}_k^{t+1} &= \boldsymbol{\beta}_k^t + \rho(\mathbf{r}_k^{t+1} - (\mathbf{h}_k * \mathbf{x}_k^{t+1} - \mathbf{y}_k)). \end{aligned}$$

C.2 FEDDEBLUR WITH TRANSFORM-DOMAIN REGULARIZATION

To handle transform-domain regularization, the objective function takes the form of

$$\min_{\mathbf{x}} F(\mathbf{x}) + \eta R(\mathbf{W}\mathbf{x}),$$

where $F(\mathbf{x})$ is the task-specific data fidelity term, which can be either $\frac{1}{n} \sum_{k=1}^n \|\mathbf{h}_k * \mathbf{x} - \mathbf{y}_k\|_2^2$ or $\frac{1}{n} \sum_{k=1}^n \|\mathbf{m}_k \odot (\mathbf{h}_k * \mathbf{x} - \mathbf{y}_k)\|_2^2$, and \mathbf{W} is a linear transform operator. Directly applying the proximal operator of $R(\mathbf{W}\mathbf{x})$ on the server can be computationally expensive. Therefore, we introduce a global auxiliary variable $\mathbf{z} = \mathbf{W}\mathbf{x}$ at the server. The federated consensus problem becomes

$$\begin{aligned} \min_{\mathbf{x}, \mathbf{z}, \{\mathbf{x}_k\}} & F(\mathbf{x}_k) + \eta R(\mathbf{z}) \\ \text{s.t.} & \mathbf{x}_k = \mathbf{x}, \quad \mathbf{z} = \mathbf{W}\mathbf{x}, \quad k = 1, \dots, n. \end{aligned}$$

This modification only affects the server-side components and has no impact on the local variables or the local objective. The augmented Lagrangian is

$$\mathcal{L}(\mathbf{x}, \mathbf{z}, \boldsymbol{\mu}, \{\mathbf{x}_k\}, \dots) = \sum_{k=1}^n L_k(\mathbf{x}, \mathbf{x}_k, \dots) + \eta R(\mathbf{z}) + \langle \boldsymbol{\mu}, \mathbf{z} - \mathbf{W}\mathbf{x} \rangle + \frac{\rho}{2} \|\mathbf{z} - \mathbf{W}\mathbf{x}\|_2^2.$$

Algorithm 2 CenDeblur-TV for complete observation and partial observation

Require: Max iterations T , regularization parameter η , ADMM parameter ρ

- 1: Initialize $\mathbf{x}^0, \mathbf{z}^0, \boldsymbol{\mu}^0$ and β_k^0 for partial observation
- 2: **for** $t = 0, 1, \dots, T - 1$ **do**
- 3: $\mathbf{z}^{t+1} \leftarrow \text{Shrink}_{\eta/\rho}(\mathbf{D}\mathbf{x}^t - \boldsymbol{\mu}^t/\rho)$ ▷ Soft shrinkage
- 4: // Complete observation
- 5: $\mathbf{x}^{t+1} \leftarrow \mathcal{F}^{-1} \left(\frac{\frac{2}{n} \sum_{k=1}^n \overline{\mathcal{F}(\mathbf{h}_k)} \mathcal{F}(\mathbf{y}_k) + \mathcal{F}(\rho \mathbf{D}^T (\mathbf{z}^{t+1} + \boldsymbol{\mu}^t/\rho))}{\frac{2}{n} \sum_{k=1}^n |\mathcal{F}(\mathbf{h}_k)|^2 + \rho |\mathcal{F}(\mathbf{D})|^2} \right)$
- 6: // Partial observation
- 7: $\mathbf{x}^{t+1} \leftarrow \mathcal{F}^{-1} \left(\frac{\sum_{k=1}^n \overline{\mathcal{F}(\mathbf{h}_k)} \mathcal{F}(\mathbf{r}_k^t + \mathbf{y}_k + \beta_k^t/\rho) + \mathcal{F}(\mathbf{D}^T (\mathbf{z}^{t+1} + \boldsymbol{\mu}^t/\rho))}{\sum_{k=1}^n |\mathcal{F}(\mathbf{h}_k)|^2 + |\mathcal{F}(\mathbf{D})|^2} \right)$
- 8: **for** $k = 1, \dots, n$ **do**
- 9: $\mathbf{r}_k^{t+1} \leftarrow \frac{\rho(\mathbf{h}_k * \mathbf{x}^{t+1} - \mathbf{y}_k) - \beta_k^t}{\frac{2}{n} \mathbf{m}_k + \rho}$
- 10: $\beta_k^{t+1} \leftarrow \beta_k^t + \rho (\mathbf{r}_k^{t+1} - (\mathbf{h}_k * \mathbf{x}^{t+1} - \mathbf{y}_k))$
- 11: **end for**
- 12: $\boldsymbol{\mu}^{t+1} \leftarrow \boldsymbol{\mu}^t + \rho (\mathbf{z}^{t+1} - \mathbf{D}\mathbf{x}^{t+1})$
- 13: **end for**
- 14: **return** \mathbf{x}^T

Here, L_k is the task-specific local objective (related to the fidelity term) as defined in (4) or (10), and $\boldsymbol{\mu}$ is a global dual variable. Client updates for local variables remain consistent with the image-domain regularization cases described in sections 3.1 and 3.2. The server update for the global primal variables is now split into separate updates for \mathbf{z} and \mathbf{x} .

Server’s global update. The server performs the same global aggregation as before:

$$\mathbf{w}^t = \frac{1}{n} \sum_{k=1}^n \mathbf{w}_k^t.$$

The global variables are updated as follows:

$$\begin{aligned} \mathbf{z}^{t+1} &= \arg \min_{\mathbf{z}} \mathcal{L}(\mathbf{x}^t, \mathbf{z}, \boldsymbol{\mu}^t, \{\mathbf{x}_k^t\}, \dots), \\ \mathbf{x}^{t+1} &= \arg \min_{\mathbf{x}} \mathcal{L}(\mathbf{x}, \mathbf{z}^{t+1}, \boldsymbol{\mu}^t, \{\mathbf{x}_k^t\}, \dots), \\ \boldsymbol{\mu}^{t+1} &= \boldsymbol{\mu}^t + \rho(\mathbf{z}^{t+1} - \mathbf{W}\mathbf{x}^{t+1}). \end{aligned}$$

After some algebraic manipulation, the \mathbf{z} -subproblem decouples the proximal update, replacing the operator for $R(\mathbf{W}\mathbf{x})$ with a simpler one for $R(\mathbf{z})$:

$$\mathbf{z}^{t+1} = \text{Prox}_{\frac{\eta}{\rho} R}(\mathbf{W}\mathbf{x}^t - \boldsymbol{\mu}^t/\rho).$$

The \mathbf{x} -subproblem is a quadratic optimization problem with a closed-form solution

$$\mathbf{x}^{t+1} = \left(n\mathbf{I} + \mathbf{W}^T \mathbf{W} \right)^{-1} \left(n\mathbf{w}^t + \mathbf{W}^T (\mathbf{z}^{t+1} + \boldsymbol{\mu}^t/\rho) \right).$$

If \mathbf{W} has a convolutional structure (e.g., the gradient operator), this inverse can be computed efficiently in the Fourier domain, as shown in (9).

C.3 BASELINE ALGORITHMS

This section provides detailed information on the baseline algorithms used in the experiments (section 4). For each baseline method, we implement two variants: one using total variation (TV) regularization, and another using a plug-and-play (PnP) prior with a pre-trained DRUNet denoiser (Zhang et al., 2021a).

CenDeblur. This method represents an idealized centralized scenario and serves as an upper-bound benchmark for performance. In this setting, the central server has direct access to all clients’

Algorithm 3 CenDeblur-PnP for complete observation and partial observation**Require:** Max iterations T , regularization parameter η , ADMM parameter ρ

```

918 1: Initialize  $\mathbf{x}^0, \mathbf{z}^0, \boldsymbol{\mu}^0$  and  $\beta_k^0$  for partial observation
919
920 Require: Max iterations  $T$ , regularization parameter  $\eta$ , ADMM parameter  $\rho$ 
921 1: Initialize  $\mathbf{x}^0, \mathbf{z}^0, \boldsymbol{\mu}^0$  and  $\beta_k^0$  for partial observation
922 2: for  $t = 0, 1, \dots, T - 1$  do
923 3:    $\mathbf{z}^{t+1} \leftarrow \mathcal{D}_{\eta/\rho}(\mathbf{x}^t - \boldsymbol{\mu}^t/\rho)$  ▷ Denoising step
924 4:   // Complete observation
925 5:    $\mathbf{x}^{t+1} \leftarrow \mathcal{F}^{-1} \left( \frac{\frac{2}{n} \sum_{k=1}^n \overline{\mathcal{F}(\mathbf{h}_k)} \mathcal{F}(\mathbf{y}_k) + \mathcal{F}(\rho \mathbf{z}^{t+1} + \boldsymbol{\mu}^t)}{\frac{2}{n} \sum_{k=1}^n |\mathcal{F}(\mathbf{h}_k)|^2 + \rho} \right)$ 
926 6:   // Partial observation
927 7:    $\mathbf{x}^{t+1} \leftarrow \mathcal{F}^{-1} \left( \frac{\sum_{k=1}^n \overline{\mathcal{F}(\mathbf{h}_k)} \mathcal{F}(\mathbf{r}_k^t + \mathbf{y}_k + \beta_k^t/\rho) + \mathcal{F}(\mathbf{z}^{t+1} + \boldsymbol{\mu}^t/\rho)}{\sum_{k=1}^n |\mathcal{F}(\mathbf{h}_k)|^2 + 1} \right)$ 
928 8:   for  $k = 1, \dots, n$  do
929 9:      $\mathbf{r}_k^{t+1} \leftarrow \frac{\rho(\mathbf{h}_k * \mathbf{x}^{t+1} - \mathbf{y}_k) - \beta_k^t}{\frac{2}{n} \mathbf{m}_k + \rho}$ 
930 10:     $\beta_k^{t+1} \leftarrow \beta_k^t + \rho(\mathbf{r}_k^{t+1} - (\mathbf{h}_k * \mathbf{x}^{t+1} - \mathbf{y}_k))$ 
931 11:  end for
932 12:   $\boldsymbol{\mu}^{t+1} \leftarrow \boldsymbol{\mu}^t + \rho(\mathbf{z}^{t+1} - \mathbf{x}^{t+1})$ 
933 13: end for
934 14: return  $\mathbf{x}^T$ 

```

Algorithm 4 LocDeblur-TV and FedAvgDeblur-TV**Require:** Max iterations T , regularization parameter η , ADMM parameter ρ

```

938 Algorithm 4 LocDeblur-TV and FedAvgDeblur-TV
939
940 Require: Max iterations  $T$ , regularization parameter  $\eta$ , ADMM parameter  $\rho$ 
941 1: Initialize  $\mathbf{x}_k^0, \mathbf{z}_k^0, \boldsymbol{\mu}_k^0$  for  $k = 1, \dots, n$ .
942 2: for  $t = 0, 1, \dots, T - 1$  do
943 3:   for  $k = 1, \dots, n$  in parallel do
944 4:      $\mathbf{x}_k^t \leftarrow \bar{\mathbf{x}}^t$  // Only for FedAvgDeblur
945 5:      $\mathbf{z}_k^{t+1} \leftarrow \text{Shrink}_{\eta/\rho}(\mathbf{D}\mathbf{x}_k^t - \boldsymbol{\mu}_k^t/\rho)$  ▷ Soft shrinkage
946 6:      $\mathbf{x}_k^{t+1} \leftarrow \mathcal{F}^{-1} \left( \frac{\overline{\mathcal{F}(\mathbf{h}_k)} \mathcal{F}(\mathbf{y}_k) + \mathcal{F}(\rho \mathbf{D}^T(\mathbf{z}_k^{t+1} + \boldsymbol{\mu}_k^t/\rho))}{|\mathcal{F}(\mathbf{h}_k)|^2 + \rho |\mathcal{F}(\mathbf{D})|^2} \right)$ 
947 7:      $\boldsymbol{\mu}_k^{t+1} \leftarrow \boldsymbol{\mu}_k^t + \rho(\mathbf{z}_k^{t+1} - \mathbf{D}\mathbf{x}_k^{t+1})$ 
948 8:   end for
949 9:    $\bar{\mathbf{x}}^{t+1} \leftarrow \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k^{t+1}$  // Only for FedAvgDeblur ▷ Averaging step
950 ▷ Average each pixel only over observed clients under partial observation
951 10: end for
952 11: return  $\bar{\mathbf{x}}^T$  for FedAvgDeblur or  $\{\mathbf{x}_k^T\}$  for LocDeblur

```

observation data (\mathbf{y}_k), blur kernels (\mathbf{h}_k), and positional masks (\mathbf{m}_k). The algorithm directly solves the global optimization problem (2) or (1) using the classic ADMM framework.

The implementations for the TV and PnP variants are detailed in Algorithm 2 and Algorithm 3, respectively. For the TV variant, the algorithm uses variable splitting by introducing an auxiliary variable $\mathbf{z} = \mathbf{D}\mathbf{x}$ to handle the TV regularizer, and the solution to the \mathbf{z} -subproblem corresponds to a soft shrinkage operation. For the PnP variant, the variable splitting is simplified to $\mathbf{z} = \mathbf{x}$, where the update for \mathbf{z} becomes a denoising step performed by the DRUNet model. In both variants, for the partial observation setting, the presence of the positional masks \mathbf{m}_k requires introducing additional auxiliary variables $\mathbf{r}_k = \mathbf{h}_k * \mathbf{x} - \mathbf{y}_k$ to decouple the problem, similar to the approach used in the partial observation extension of FedDeblur.

LocDeblur and FedAvgDeblur. The core of both LocDeblur and FedAvgDeblur is a standard ADMM-based deblurring method that runs locally on each client. The implementation for the TV versions is unified in Algorithm 4, and for the PnP versions in Algorithm 5. The primary difference between the two versions lies in the local proximal update step: the TV variant uses a soft shrinkage operation for the auxiliary variable \mathbf{z}_k , while the PnP variant employs a denoising network.

Algorithm 5 LocDeblur-PnP and FedAvgDeblur-PnP**Require:** Max iterations T , regularization parameter η , ADMM parameter ρ

```

1: Initialize  $\mathbf{x}_k^0, \mathbf{z}_k^0, \boldsymbol{\mu}_k^0$  for  $k = 1, \dots, n$ .
2: for  $t = 0, 1, \dots, T - 1$  do
3:   for  $k = 1, \dots, n$  in parallel do
4:      $\mathbf{x}_k^t \leftarrow \bar{\mathbf{x}}^t$  // Only for FedAvgDeblur
5:      $\mathbf{z}_k^{t+1} \leftarrow \mathcal{D}_{\eta/\rho}(\mathbf{x}_k^t - \boldsymbol{\mu}_k^t/\rho)$  ▷ Denoising step
6:      $\mathbf{x}_k^{t+1} \leftarrow \mathcal{F}^{-1}\left(\frac{\mathcal{F}(\mathbf{h}_k)\mathcal{F}(\mathbf{y}_k) + \mathcal{F}(\rho\mathbf{z}_k^{t+1} + \boldsymbol{\mu}_k^t)}{|\mathcal{F}(\mathbf{h}_k)|^2 + \rho}\right)$ 
7:      $\boldsymbol{\mu}_k^{t+1} \leftarrow \boldsymbol{\mu}_k^t + \rho(\mathbf{z}_k^{t+1} - \mathbf{x}_k^{t+1})$ 
8:   end for
9:    $\bar{\mathbf{x}}^{t+1} \leftarrow \frac{1}{n} \sum_{k=1}^n \mathbf{x}_k^{t+1}$  // Only for FedAvgDeblur ▷ Averaging step
   ▷ Average each pixel only over observed clients under partial observation
10: end for
11: return  $\bar{\mathbf{x}}^T$  for FedAvgDeblur or  $\{\mathbf{x}_k^T\}$  for LocDeblur

```

LocDeblur is a fully non-collaborative baseline that represents a performance lower bound. Each client independently runs its local ADMM algorithm until convergence without any communication with the server or other clients. For evaluation purposes, the final restored images from all clients $\{\mathbf{x}_k^T\}$ are averaged.

FedAvgDeblur is inspired by the federated averaging (FedAvg) algorithm. Unlike LocDeblur, it involves communication in every round. In each round, every client begins with the global average model from the previous round, $\bar{\mathbf{x}}^t$, and performs a single-step local ADMM update to obtain its new local estimate, \mathbf{x}_k^{t+1} . The clients then upload these updated estimates to the server. The server computes their average to produce the new global model, $\bar{\mathbf{x}}^{t+1}$, and broadcasts it to all clients for the next round. Although FedAvgDeblur enables collaboration, it requires the direct transmission and averaging of the image estimates \mathbf{x}_k , which violates the “intermediate privacy” principle. Furthermore, since each client must independently perform the proximal update, it can be computationally expensive when using deep priors, potentially violating the “computational efficiency” principle.

Modularity Comparison. A key advantage of our FedDeblur framework (Algorithm 1) is its modularity, particularly when switching between different image priors. When changing from a TV regularizer to a PnP prior, the necessary modifications are confined entirely to the server-side update steps (line 7 versus lines 9 to 11). Crucially, the client-side computations and the communication protocol remain unchanged, making the switch completely transparent to the clients.

In contrast, the baseline methods require more substantial alterations. For instance, comparing CenDeblur-TV (Algorithm 2) and CenDeblur-PnP (Algorithm 3) reveals that nearly every step of the iterative process is modified due to the different variable splitting strategies. Similarly, for LocDeblur and FedAvgDeblur, switching from the TV version (Algorithm 4) to the PnP version (Algorithm 5) requires changing the entire local update logic for each client. For FedAvgDeblur, only the averaging step remains the same, while for LocDeblur, every step of the iteration is different. This inherent flexibility demonstrates the robustness and practical advantages of our proposed federated design.

D ADDITIONAL EXPERIMENTAL DETAILS

D.1 IMPLEMENTATION DETAILS

Our experiments are conducted on a set of standard grayscale and color images, as shown in Figure 5. All images are resized to 256×256 pixels. We quantitatively evaluate the quality of the restored images using the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM). Except for appendix D.4, the blurred observations in all experiments are corrupted by additive white

Gaussian noise with a standard deviation of 1. All iterative algorithms are run for a maximum of 500 iterations or until the relative change in the restored image falls below a threshold of 10^{-5} , i.e., $\|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2 / \|\mathbf{x}^t\|_2 < 10^{-5}$. For each method under each setting, hyperparameters are individually optimized via grid search. All experiments were implemented in Python using the PyTorch framework¹ and performed on a workstation equipped with an NVIDIA RTX 3090 GPU.

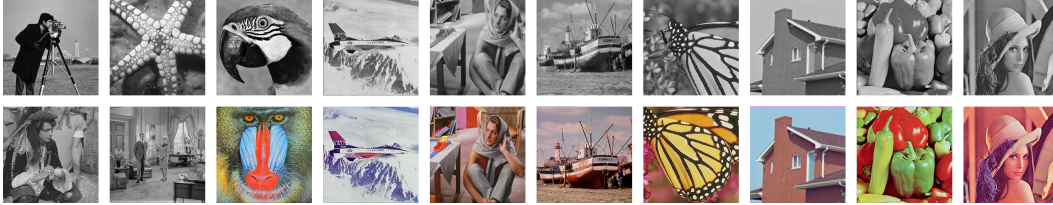


Figure 5: Test images used in our experiments. The grayscale set includes Cameraman, Fishstar, Parrot, Jetplane, Barbara, Boat, Butterfly, House, Pepper, Lena, Male, and Couple. The color set includes Baboon, Jetplane, Barbara, Boat, Butterfly, House, Pepper, and Lena.

For the complete observation setting, we simulate scenarios with n clients, where n ranges from 3 to 10. The blur kernel for each client is selected from a predefined set of 10 diverse kernels, including motion, Gaussian, disk, and synthesized types. The specific parameters for these 10 kernels are detailed in Table 3. An experiment with n clients utilizes the first n kernels from this table.

Table 3: Blur kernel parameters for each of the 10 clients in the complete observation setting.

Client k	1	2	3	4	5	6	7	8	9	10
Kernel Type	Motion	Gaussian	Synthetic	Motion	Disk	Motion	Gaussian	Disk	Synthetic	Motion
Parameters	len=15 $\theta=45^\circ$	size=9 $\sigma=2$	15×15	len=20 $\theta=0^\circ$	$r=5$	len=15 $\theta=135^\circ$	size=10 $\sigma=3$	$r=10$	12×12	len=20 $\theta=90^\circ$

For the partial observation setting, the division of the scene into client-specific observation regions is described in section 4.1. We consider two distinct blur degradation scenarios: “motion” and “synthesis”. In the “motion” scenario, each of the nine clients is assigned a unique motion blur kernel with varying lengths and angles, as specified in Table 4. In the “synthesis” scenario, the nine blur kernels are randomly generated camera-shake kernels. These kernels are generated using the public code from KAIR² with a fixed random seed to ensure reproducibility. The resulting kernels are visualized in Figure 2b.

Table 4: Motion blur kernel parameters for each of the 9 clients in the partial observation setting.

Client k	1	2	3	4	5	6	7	8	9
Length	10	11	13	15	17	19	21	23	25
Angle ($^\circ$)	0	22	45	67	90	112	135	157	180

D.2 EFFECT OF THE NUMBER OF CLIENTS

To evaluate the scalability and effectiveness of our federated framework, we vary the number of clients n from 1 to 10 in the complete observation setting. As shown in Figure 6, PSNR and SSIM increase steadily as more clients participate across different image priors and datasets. The largest gains occur when moving from a small number of clients, after which the curves gradually plateau, indicating that while adding more clients is always beneficial, the marginal utility diminishes. This consistent trend across different test data (grayscale and color) and priors (TV and PnP) confirms that the observed improvement is a fundamental property of our algorithm and the complementary information contributed by each client’s unique degradation.

¹Code is available at <https://anonymous.4open.science/r/FedDeblur-53D7>.

²https://github.com/cszn/KAIR/blob/master/utils/utils_deblur.py.

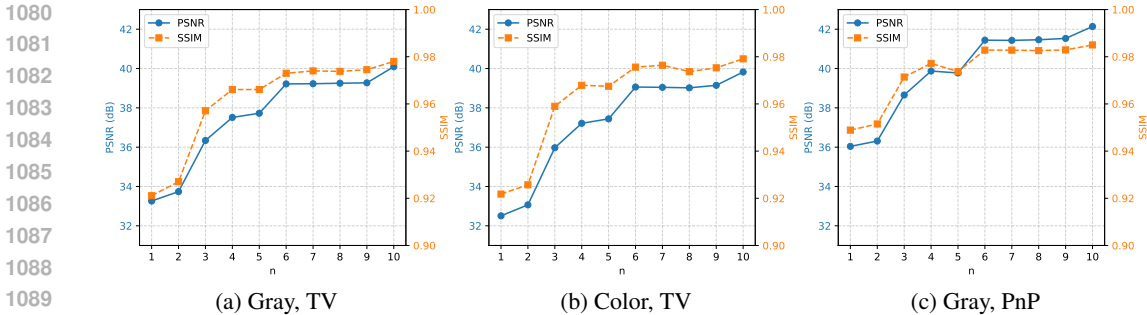


Figure 6: Performance of FedDeblur as the number of clients n in the complete observation setting. Results are averaged over all test images for the TV prior on (a) grayscale and (b) color datasets, and for the PnP prior on (c) the grayscale dataset.

This trend arises from aggregating complementary information across clients. Each client provides an observation degraded by a unique blur kernel, which corrupts the underlying image in different ways. Figure 7 illustrates this effect: as n increases from 3 to 7, the restored image recovers finer textures. The convergence plots in Figure 7d and 7e further show that more clients yield higher final quality and faster convergence.

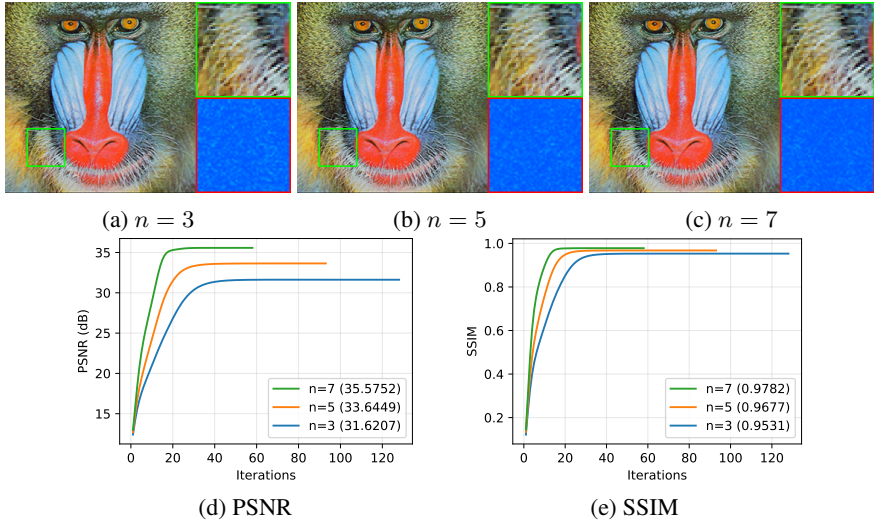


Figure 7: Visual examples and convergence plots for different numbers of clients n on the “Baboon” image with TV prior. (a-c) Reconstruction crops for $n = 3, 5, 7$. Each image is accompanied by an enlarged patch (top right) and its residual map (bottom right). PSNR and SSIM values are displayed on the top left of each image. (d-e) PSNR and SSIM versus iteration for varying n ; final metric values are listed in the legends.

D.3 PARTIAL CLIENT PARTICIPATION

In practical federated learning deployments, participating clients often consist of a wide range of devices with heterogeneous hardware, leading to varying local computation speeds (Li et al., 2020b; Ye et al., 2023). Furthermore, these devices are typically connected to diverse network environments with significant differences in bandwidth, latency, and stability. This can result in clients completing their communication at vastly different times, with some even failing to respond within a given time window (McMahan et al., 2017; Chen et al., 2021). These factors contribute to the “straggler” problem, where the server would be forced to wait for the slowest clients before performing a global aggregation, thereby reducing overall efficiency (Li et al., 2020b). A common strategy to mitigate

Algorithm 6 FedDeblur for partial client participation

Require: Communication rounds T , regularization parameter η , ADMM parameter ρ

- 1: **Server:** Initialize $\mathbf{x}^0, \mathcal{S}_0 = \{1, \dots, n\}$
- 2: **Clients** $k = 1, \dots, n$: Initialize $\mathbf{x}_k^0, \boldsymbol{\lambda}_k^0, \mathbf{w}_k^0 = \mathbf{x}_k^0 + \boldsymbol{\lambda}_k^0/\rho$; and upload \mathbf{w}_k^0 to the server
- 3: **for** $t = 0, 1, \dots, T - 1$ **do**
- 4: **server do**
- 5: Aggregate $\mathbf{w}^t \leftarrow \frac{1}{|\mathcal{S}_t|} \sum_{k \in \mathcal{S}_t} \mathbf{w}_k^t$
- 6: Server update same as Algorithm 1
- 7: Select the active clients set \mathcal{S}_{t+1} and broadcast \mathbf{x}^{t+1} to the active clients
- 8: **for each active client** $k \in \mathcal{S}_{t+1}$ **in parallel do**
- 9: Receive global model \mathbf{x}^{t+1} from the server
- 10: Update local variables same as Algorithm 1
- 11: Compute \mathbf{w}_k^{t+1} and upload to the server
- 12: **end for**
- 13: **end for**
- 14: **return** global estimate \mathbf{x}^T

this is to proceed with aggregation after a sufficient number of clients have responded, which requires the federated optimization algorithm to support partial client participation.

We model this scenario by having the server select a random subset of active clients $\mathcal{S}^t \subseteq \{1, 2, \dots, n\}$ in each communication round t . The modified workflow is detailed in Algorithm 6. To study the impact of partial participation, we conduct experiments on a system with a total of $n = 10$ clients, varying the number of randomly selected active clients, $p = |\mathcal{S}^t|$, from 2 to 10 in each round.

The results, shown in Figure 8, demonstrate that while a higher participation rate consistently improves performance, our framework remains robust and effective even at low participation rates. As shown in the convergence plots in Figure 9, lower participation rates lead to slower convergence and more pronounced oscillations in the PSNR curve. This increased variability is expected, as each server update is based on a smaller, more random subset of client information, making the global update trajectory more stochastic. Nonetheless, the visual results confirm that high-quality restoration is achievable even when only a small fraction of clients participates in each round.

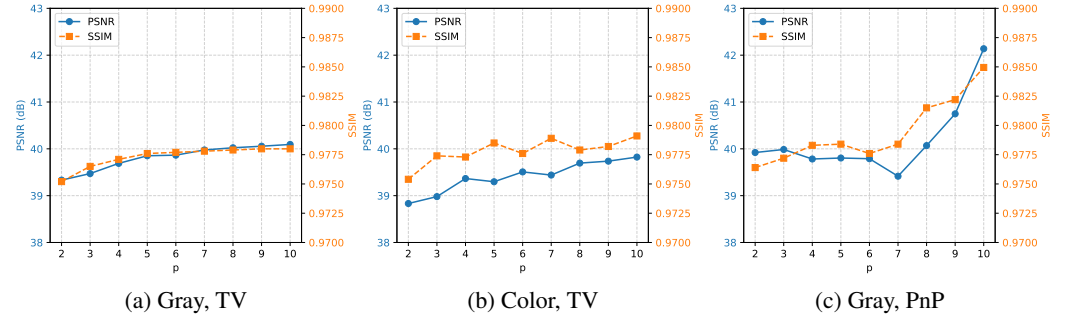
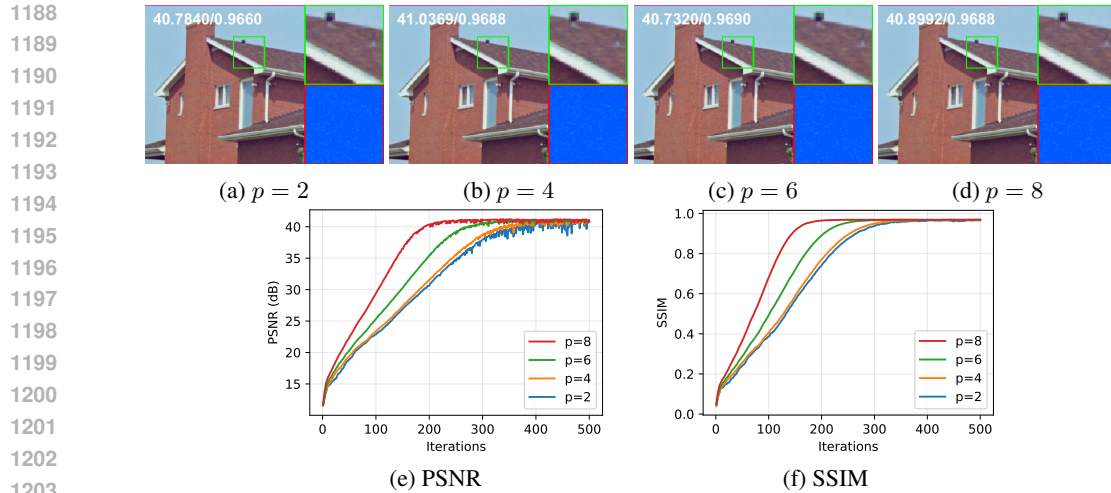


Figure 8: Performance of FedDeblur with partial client participation on a system with $n = 10$ total clients, as the number of active clients per round p varied from 2 to 10. Results are shown for (a) grayscale images with TV prior, (b) color images with TV prior, and (c) grayscale images with PnP prior.

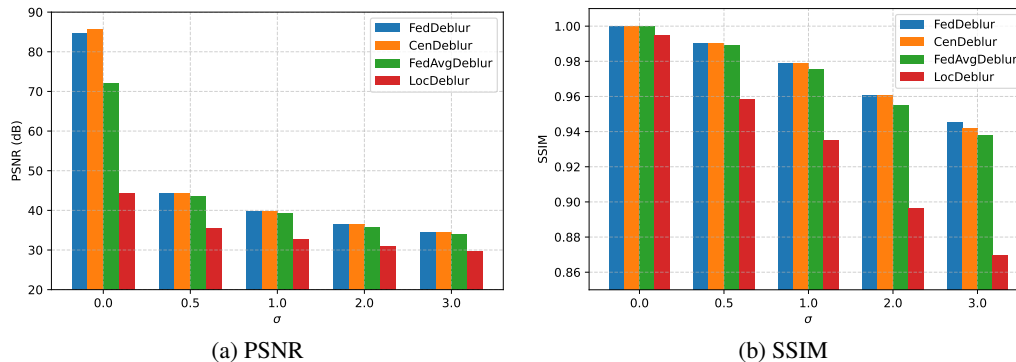
D.4 ROBUSTNESS TO OBSERVATION NOISE

To assess the robustness of our framework to observation noise, we evaluate performance under additive white Gaussian noise with standard deviation σ ranging from 0 (noiseless) to 3. Complete quantitative results are reported in Table 5 (results for $\sigma = 1$ appear in Table 1 in section 4.2), and Figure 10 summarizes the case of $n = 10$ color images.



1204
1205
1206
1207
1208
1209
1210
1211

Figure 9: Visual results and convergence analysis for partial client participation on the House image with $n = 10$ total clients and the TV prior. (a-d) Final deblurring results for different numbers of active clients p . Each image is accompanied by an enlarged patch (top right) and its residual map (bottom right). PSNR and SSIM values are displayed on the top left of each image. (e-f) Corresponding convergence curves for PSNR and SSIM.



1224
1225
1226
1227
1228

Figure 10: Performance under varying noise levels (σ) in the complete observation setting with $n = 10$ on color images.

1229
1230
1231
1232
1233
1234
1235

Across all noise levels, the relative ranking of methods remains stable, confirming that the advantages of our federated design are fundamental and not tied to a specific noise setting. As noise increases, all methods degrade as expected, yet *FedDeblur* closely matches the idealized centralized oracle *CenDeblur* with only a small gap. This shows strong robustness of our framework to noise. Our consensus mechanism effectively aggregates out the effects of random noise from multiple observations, yielding a cleaner global reconstruction than non-collaborative or naive averaging approaches.

1236
1237
1238
1239
1240
1241

In particular, as shown in Figure 10b, the SSIM gap between the collaborative methods (*FedDeblur*, *CenDeblur*, *FedAvgDeblur*) and the non-collaborative baseline (*LocDeblur*) grows with noise, which highlights the importance of information fusion in noisy scenarios. The consistent margin between *FedDeblur* and *FedAvgDeblur* further supports the advantage of our consensus optimization over naive parameter averaging, especially when observations are heavily corrupted. Even in the noiseless case ($\sigma = 0$), where PSNR values are very high, the architectural advantages of our method remain evident.

Table 5: Comparison with TV regularization across different noise levels (σ). For each metric, the best result is highlighted in magenta and the second-best in cyan.

Setup	Type	FedDeblur		CenDeblur		FedAvgDeblur		LocDeblur	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$\sigma = 0$									
$n = 10$	Gray	81.3782	1.0000	81.2093	1.0000	68.0343	0.9999	44.9688	0.9944
	Color	84.6834	1.0000	85.5895	1.0000	71.8875	1.0000	44.2351	0.9947
$n = 5$	Gray	74.3669	1.0000	74.5099	1.0000	65.0938	0.9999	46.2258	0.9960
	Color	76.6074	1.0000	76.6508	1.0000	66.8386	0.9999	45.2914	0.9961
$n = 3$	Gray	64.2039	0.9999	64.1226	0.9999	58.0734	0.9994	42.7921	0.9922
	Color	66.4204	0.9999	66.2443	0.9999	60.7934	0.9997	41.6485	0.9919
Motion	Gray	41.4498	0.9866	41.7750	0.9865	23.2112	0.7302	23.2817	0.7364
	Color	40.7575	0.9871	41.0725	0.9872	22.5916	0.7077	22.6414	0.7086
Synthesis	Gray	35.6857	0.9634	35.5461	0.9686	21.6630	0.6642	21.5664	0.6661
	Color	34.5416	0.9618	34.0416	0.9645	21.3235	0.6587	21.2558	0.6509
$\sigma = 0.5$									
$n = 10$	Gray	44.3863	0.9898	44.3857	0.9899	43.7826	0.9881	36.1205	0.9634
	Color	44.1880	0.9901	44.1937	0.9902	43.5805	0.9889	35.3630	0.9583
$n = 5$	Gray	41.3922	0.9818	41.3976	0.9818	41.1850	0.9811	36.4778	0.9609
	Color	41.1870	0.9838	41.1890	0.9838	40.9501	0.9822	35.6259	0.9599
$n = 3$	Gray	39.5199	0.9763	39.5249	0.9764	39.4486	0.9754	35.2012	0.9549
	Color	39.2602	0.9771	39.2200	0.9777	39.1419	0.9757	34.3165	0.9544
Motion	Gray	36.6886	0.9594	36.6801	0.9594	23.2099	0.7301	23.2805	0.7362
	Color	35.9757	0.9585	35.8457	0.9589	22.4477	0.7169	22.6406	0.7085
Synthesis	Gray	33.4607	0.9372	33.6281	0.9370	21.6621	0.6641	21.5366	0.6658
	Color	32.5743	0.9353	32.7310	0.9357	21.3575	0.6530	21.2060	0.6539
$\sigma = 2$									
$n = 10$	Gray	36.7494	0.9595	36.7347	0.9597	36.3046	0.9546	31.5904	0.9132
	Color	36.3651	0.9605	36.3690	0.9604	35.7339	0.9549	30.7809	0.8961
$n = 5$	Gray	34.6002	0.9420	34.6011	0.9420	34.1911	0.9413	31.7884	0.9142
	Color	34.0418	0.9448	34.0465	0.9447	33.9568	0.9404	30.9929	0.9043
$n = 3$	Gray	33.4663	0.9336	33.4664	0.9336	33.3500	0.9322	31.5310	0.9113
	Color	33.0918	0.9303	33.0994	0.9303	32.9590	0.9290	30.7611	0.8956
Motion	Gray	31.0227	0.8958	30.9980	0.8962	23.1970	0.7290	23.2635	0.7348
	Color	30.2842	0.8881	30.2700	0.8870	22.4907	0.7160	22.6069	0.7097
Synthesis	Gray	29.7694	0.8773	29.7735	0.8769	21.6534	0.6632	21.4859	0.6636
	Color	28.9029	0.8659	28.9224	0.8660	21.3164	0.6580	21.2428	0.6502
$\sigma = 3$									
$n = 10$	Gray	34.9053	0.9399	34.9918	0.9455	34.3851	0.9388	30.3957	0.8923
	Color	34.4920	0.9453	34.4931	0.9418	33.8898	0.9379	29.5742	0.8697
$n = 5$	Gray	33.0265	0.9255	33.0331	0.9254	32.3920	0.9230	30.6704	0.8776
	Color	32.5696	0.9258	32.5790	0.9256	32.2623	0.9239	29.7825	0.8814
$n = 3$	Gray	31.7953	0.9143	32.1334	0.9137	31.4859	0.9099	30.4770	0.8924
	Color	31.5071	0.9125	31.5205	0.9125	31.2762	0.9102	29.5814	0.8819
Motion	Gray	29.6523	0.8708	29.7025	0.8694	23.1817	0.7278	23.2491	0.7330
	Color	28.9283	0.8605	28.9197	0.8565	22.4484	0.7164	22.5858	0.7075
Synthesis	Gray	28.6313	0.8529	28.6659	0.8534	21.6431	0.6621	21.4184	0.6654
	Color	27.9674	0.8371	27.9135	0.8386	21.0973	0.6616	21.2364	0.6493

D.5 NUMERICAL CONVERGENCE ANALYSIS

To better illustrate the optimization dynamics of our framework, we present a numerical analysis of convergence for several representative settings, shown in Figure 11 to 13.

The federated methods, FedDeblur and FedAvgDeblur begin with low metric values because the server uses random initialization to comply with privacy principles that prohibit access to any client data. By contrast, the centralized method, CenDeblur, is initialized with the average of all blurry observations, and the non-collaborative baseline, LocDeblur, starts from each client’s observation, giving both a significant head start. Despite the lower starting point, FedDeblur improves rapidly, and its trajectory is nearly parallel to that of the idealized CenDeblur oracle, indicating a comparable convergence rate. Moreover, FedDeblur consistently converges faster and to a better result than the naive federated baseline, FedAvgDeblur. This holds for both TV and PnP priors, highlighting the effectiveness of the proposed consensus mechanism.

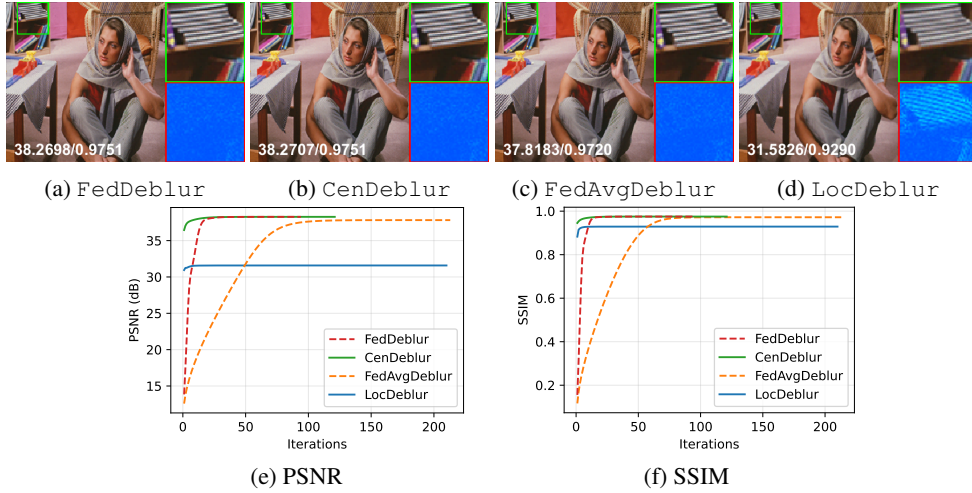


Figure 11: Visual results and convergence analysis for the complete observation setting with $n = 10$ clients on the color Barbara image using the TV prior. (a-d) Final results; Each image is accompanied by an enlarged patch (top right) and its residual map (bottom right); PSNR and SSIM are shown in the bottom left of each image. (e-f) PSNR and SSIM versus iteration number.

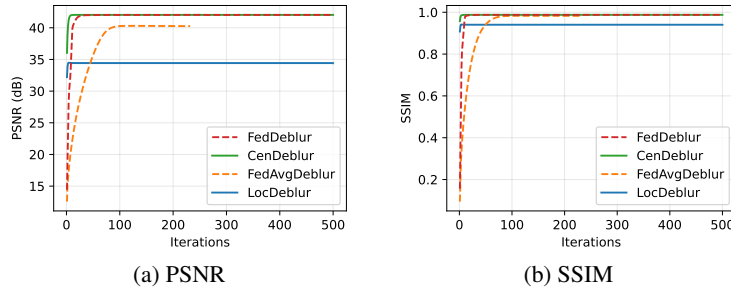


Figure 12: Convergence analysis for the complete observation setting with $n = 10$ clients on the grayscale Barbara image using the PnP prior. The corresponding visual results appear in Figure 3.

Furthermore, in the partial observation setting (Figure 13), the PSNR curve for FedDeblur exhibits periodic fluctuations. To study this behavior, we track several quantities: the consensus error $\varepsilon^t = \mathbf{x}_k^t - \mathbf{x}^t$, the change in the global primal variable $\Delta \mathbf{x}^t = \mathbf{x}^t - \mathbf{x}^{t-1}$, the change in the shared variable $\Delta \mathbf{w}^t = \mathbf{w}^t - \mathbf{w}^{t-1}$, and the change in PSNR per iteration. As shown in Figure 13g, these metrics are strongly correlated and oscillatory.

This behavior reflects a feedback loop caused by the lag between local optimization and global consensus. In each iteration, clients update \mathbf{x}_k to balance fidelity to private data with agreement to the global model from the *previous* round. In overlapping regions, different blur kernels yield conflicting estimates for the same pixels. The server aggregates these into a new global model that can be a compromise and thus suboptimal for any individual client. The induced consensus error is stored in the dual variables, which act as memory. In the next iteration, these dual variables drive a strong correction toward consensus; this can overshoot and produce oscillations across iterations.

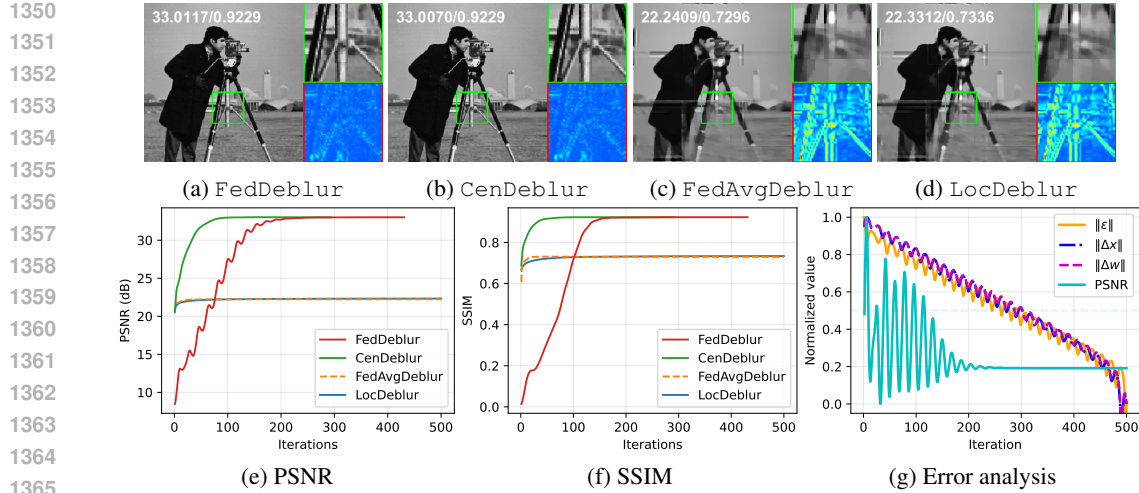


Figure 13: Visual results and convergence analysis for the partial observation setting with motion blur on the Cameraman image using the TV prior. (a-d) Final results; Each image is accompanied by an enlarged patch (top right) and its residual map (bottom right); PSNR and SSIM are shown in the top left of each image. (e-f) Performance metrics versus iteration. (g) The consensus error ($\|\epsilon^t\|$), changes in the local and shared variables ($\|\Delta x^t\|$, $\|\Delta w^t\|$), and the change in PSNR. Data in (g) were log-transformed and linearly scaled for clarity.

The synchronized peaks in the error metrics and the PSNR derivative in Figure 13g support this explanation: a build-up of consensus error triggers system-wide corrections, which manifest as periodic fluctuations.

To further demonstrate the superior convergence properties of our ADMM-based federated optimization framework, we compare it against a typical gradient descent (GD) based strategy, which serves as the backbone for many federated learning algorithms (e.g. McMahan et al. (2017); Karimireddy et al. (2020)). We formulate the FedGDDeblur baseline by directly applying gradient descent to the global objective

$$\min_{\mathbf{x}} \frac{1}{n} \sum_{k=1}^n \|\mathbf{h}_k * \mathbf{x} - \mathbf{y}_k\|_2^2 + \eta \|\mathbf{D}\mathbf{x}\|_1.$$

The update rule for gradient descent with a learning rate α is given by

$$\mathbf{x}^{t+1} = \mathbf{x}^t - \alpha \left[\frac{2}{n} \sum_{k=1}^n \bar{\mathbf{h}}_k * (\mathbf{h}_k * \mathbf{x}^t - \mathbf{y}_k) - \eta \operatorname{div} \left(\frac{\mathbf{D}\mathbf{x}^t}{|\mathbf{D}\mathbf{x}^t|} \right) \right],$$

where $\bar{\mathbf{h}}_k$ denotes the flipped kernel (adjoint of convolution), and the regularization gradient is derived from the smoothed total variation. We implement this in a federated manner as shown in Algorithm 7, where clients compute gradients of the data fidelity term locally, and the server aggregates them and applies the regularization gradient.

We evaluated both methods on the color “Jetplane” image under the complete observation setting ($n = 10$) with TV regularization. As shown in Figure 14, FedDeblur significantly outperforms FedGDDeblur in both convergence speed and restoration quality. The visual results in Figure 14a and 14b reveal that FedGDDeblur fails to effectively remove blur and noise, resulting in artifacts, whereas FedDeblur produces a sharp, clean image.

The quantitative curves in Figure 14c and 14d confirm that FedDeblur achieves high PSNR and SSIM values within very few iterations. In contrast, FedGDDeblur converges extremely slowly and plateaus at a much lower quality level. Furthermore, the objective function plot in Figure 14e shows that our ADMM-based approach minimizes the energy function rapidly, while the GD-based approach struggles with the ill-conditioned nature of the deblurring problem. This comparison underscores the advantage of using proximal splitting methods (which handle non-smooth regularizers and ill-posed operators robustly) over standard gradient descent in federated inverse problems.

Algorithm 7 FedGDDeblur

Require: Communication rounds T , learning rate α , regularization parameter η

- 1: **Server:** Initialize global estimate \mathbf{x}^0
- 2: **for** $t = 0, 1, \dots, T - 1$ **do**
- 3: **for each client** $k = 1, \dots, n$ **in parallel do**
- 4: Receive global model \mathbf{x}^t from the server
- 5: $\mathbf{g}_k^t \leftarrow 2\mathbf{h}_k * (\mathbf{h}_k * \mathbf{x}^t - \mathbf{y}_k)$ ▷ Compute local gradient
- 6: Upload \mathbf{g}_k^t to the server
- 7: **end for**
- 8: **Server do**
- 9: $\bar{\mathbf{g}}_{\text{data}}^t \leftarrow \frac{1}{n} \sum_{k=1}^n \mathbf{g}_k^t$ ▷ Aggregate data fidelity gradients
- 10: $\mathbf{g}_{\text{reg}}^t \leftarrow -\eta \operatorname{div} \left(\frac{D\mathbf{x}^t}{|D\mathbf{x}^t|} \right)$ ▷ Compute regularization gradient
- 11: $\mathbf{x}^{t+1} \leftarrow \mathbf{x}^t - \alpha (\bar{\mathbf{g}}_{\text{data}}^t + \mathbf{g}_{\text{reg}}^t)$ ▷ Update global estimate
- 12: **end for**
- 13: **return** final estimate \mathbf{x}^T

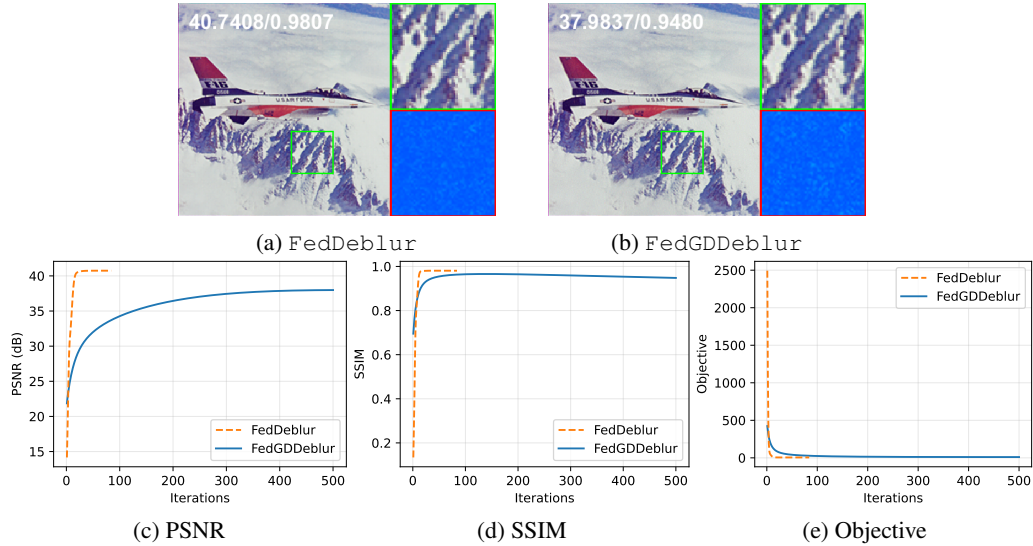


Figure 14: Comparison between FedDeblur and the gradient descent baseline FedGDDeblur on the color “Jetplane” image ($n = 10$, complete observation, TV prior). (a-b) Visual results with PSNR and SSIM shown in the top left of each image. (c-e) Convergence curves for PSNR, SSIM, and the objective function value.

D.6 COMPUTATIONAL EFFICIENCY AND RUNTIME ANALYSIS

To further evaluate the computational resource consumption of the proposed framework, we recorded the average wall-clock time per communication round and the total time required to reach specific image quality thresholds. We compared the two federated methods, FedDeblur and FedAvgDeblur, in the complete observation setting with $n = 10$ clients using TV regularization. The experiments were performed on the same hardware described in appendix D.1. Table 6 summarizes the results. The first two columns report the average time per iteration averaged over the entire grayscale and color datasets, respectively. The remaining columns detail the number of communication rounds and the total cumulative time required to reach specific PSNR (35, 37, 38 dB) and SSIM (0.95, 0.97, 0.975) targets on the color “Barbara” image.

The results demonstrate the significant efficiency advantage of our method. In terms of per-iteration cost, FedDeblur is approximately $2\times$ to $3\times$ faster than FedAvgDeblur. Furthermore, FedDeblur exhibits a much faster convergence rate. To achieve a PSNR of 37 dB, our method requires only 14 rounds (0.37 seconds), whereas FedAvgDeblur requires 82 rounds (4.35 sec-

1458 Table 6: Runtime and convergence speed comparison between FedDeblur and FedAvgDeblur.
 1459 The average iteration time is computed over the entire dataset. The convergence targets (rounds /
 1460 total time) are reported for the color “Barbara” image ($n = 10$, complete observation, TV prior).
 1461 Entries marked with “–” indicate that the method failed to reach the target within 500 rounds.
 1462

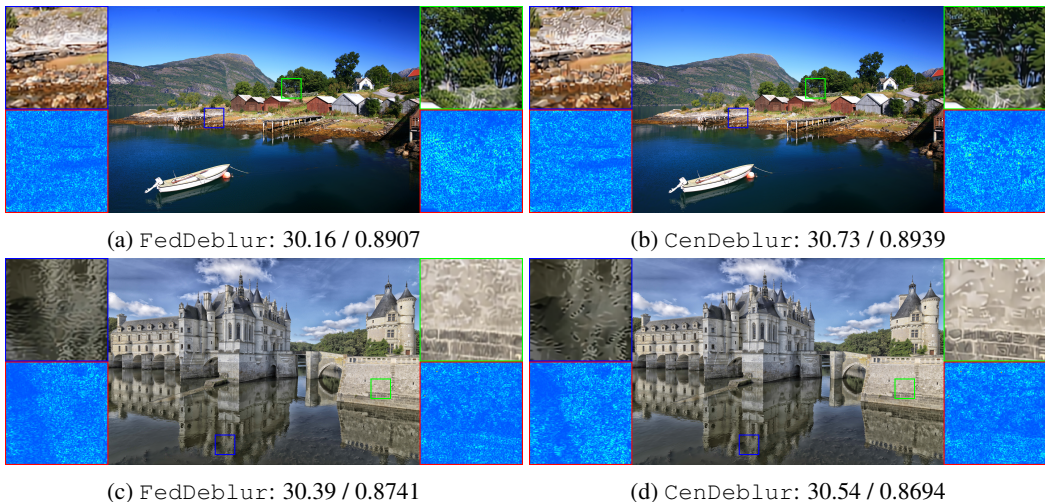
Method	Avg. Time per Round (s)		Rounds / Total Time (s) to reach target					
	Gray	Color	PSNR (dB)			SSIM		
			35	37	38	0.95	0.97	0.975
FedDeblur	0.0129	0.0275	12 / 0.32	14 / 0.37	20 / 0.53	12 / 0.32	15 / 0.40	42 / 1.11
FedAvgDeblur	0.0370	0.0560	64 / 3.41	82 / 4.35	–	65 / 3.46	84 / 4.45	–

1469
 1470 onds)—a speedup of over $10\times$ in total time. More importantly, FedDeblur is able to reach higher
 1471 quality thresholds (PSNR 38 dB and SSIM 0.975) that FedAvgDeblur fails to attain even after
 1472 extended iterations. This confirms that the consensus-based optimization of FedDeblur not only
 1473 converges faster but also converges to a superior solution compared to the naive averaging of local
 1474 estimates.

1475 **D.7 SCALABILITY TO HIGH-RESOLUTION AND REAL-WORLD SCENARIOS**
 1476

1477 To further validate the scalability and robustness of our framework in more realistic and challeng-
 1478 ing environments, we extended our evaluation to high-resolution data and real-world multi-view
 1479 scenarios.

1480 First, we assessed the performance on high-definition images using the DIV2K dataset (Agustsson
 1481 & Timofte, 2017). We selected real-world images with an resolution of 2040×1356 . To simulate a
 1482 large-scale federated system, we constructed a scenario with $n = 16$ clients arranged in a 4×4 grid,
 1483 where each client captures a high-resolution partial view of 600×500 pixels. This setup represents
 1484 a significant increase in scale compared to previous experiments. Furthermore, to heighten the
 1485 difficulty of the restoration task, we employed larger 25×25 synthetic blur kernels. A larger kernel
 1486 size induces more severe blur and stronger inter-pixel dependencies, making the inverse problem
 1487 significantly more ill-posed. The restoration was performed using the PnP prior following the same
 1488 protocol described in appendix D.1.



1506 Figure 15: Visual comparison on high-resolution images from the DIV2K dataset. The scene is
 1507 captured by $n = 16$ clients arranged in a 4×4 grid with partial overlap, degraded by 25×25
 1508 synthetic blur kernels. PSNR (dB) and SSIM are reported below each image.
 1509

1510 As illustrated in Figure 15, FedDeblur maintains exceptional performance even in this challeng-
 1511 ing setting. Quantitatively, the PSNR and SSIM scores are comparable to those of the idealized
 centralized baseline (CenDeblur). Qualitatively, the restored images exhibit sharp details, and

notably, there are no visible seam artifacts at the boundaries of the overlapping client sub-regions. This confirms that our consensus mechanism effectively fuses local information into a coherent global image, scaling seamlessly to larger client arrays and higher resolutions.

Subsequently, we evaluated our method in a more realistic multi-view scenario using the DTU MVS dataset (Aanæs et al., 2016). Unlike the synthetic cropping used in previous experiments, we selected five adjacent views of a single scene to represent five independent clients. These images naturally exhibit distinct perspectives and varying lighting conditions, thereby introducing significant data heterogeneity that better reflects practical federated deblurring deployments.

To mitigate the domain gap caused by physical camera displacement and sensor variations, we performed standard pre-processing steps prior to optimization: the enhanced correlation coefficient (ECC) algorithm (Evangelidis & Psarakis, 2008) was used for spatial alignment, and histogram matching was applied for color correction. It is important to note that these pre-processing steps are external to our FedDeblur algorithm.

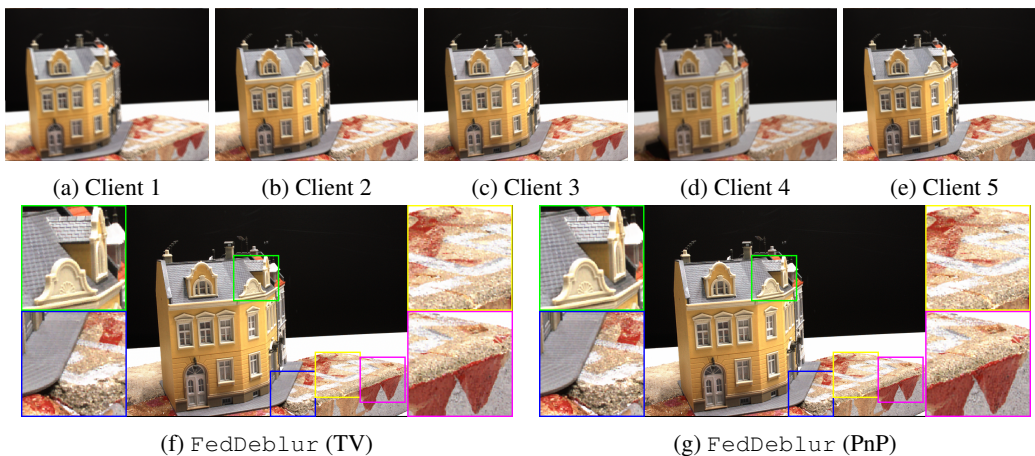


Figure 16: Visual evaluation on real-world multi-view data from the DTU MVS dataset. The system consists of $n = 5$ clients, each capturing the scene from a different viewing angle with varying lighting conditions. (a-e) The blurred observations held by each client after preprocessing. (f-g) The global clear image recovered by FedDeblur using TV and PnP priors, respectively.

Since ground truth is unavailable for this specific real-world configuration, we rely on visual inspection. As shown in Figure 16, the input observations from the five clients suffer from significant degradation. However, FedDeblur successfully integrates the information from these heterogeneous views. Both the TV and PnP variants recover sharp geometric structures and fine textures (e.g. details of roof tiles and rough surfaces) while effectively suppressing noise. This demonstrates that our consensus-based framework is robust not only to synthetic partitions but also to the geometric and photometric inconsistencies inherent in real-world multi-view imaging.

D.8 SENSITIVITY TO HYPERPARAMETERS

We examine the sensitivity of FedDeblur to its two hyperparameters: the regularization parameter η , which balances the data fidelity and prior terms, and the ADMM penalty parameter ρ , which sets the strength of the constraint penalty. We evaluate performance on an 11×11 grid centered at the grid-searched optimum, with adjacent values differing by 10%. Figure 17 shows performance surfaces for two representative cases: complete observation with $n = 5$ and partial observation with synthesized kernels, both using a TV prior.

The results show that the framework is robust to both hyperparameters. In particular, performance is very stable with respect to ρ . As shown across all subplots in Figure 17, the surfaces are nearly flat along the ρ -axis, indicating that the final restoration quality is largely insensitive to ρ over a wide range. The convergence analysis in Figure 18 provides a more dynamic perspective. As shown in Figure 18m and 18n, while different values of ρ influence the convergence path and speed, they

all lead to nearly identical final performance. This confirms that while ρ can be tuned to affect convergence speed, its impact on the final restoration quality is minimal.

As expected, the regularization parameter η has a more pronounced effect. Even so, the landscape is smooth with a broad peak around the optimum, without sharp drops for nearby values. The convergence curves in Figure 18k and 18l further clarify this relationship. Different values of η lead to different final performance levels, as expected, since this parameter controls the trade-off between data fidelity and regularization. However, the visual results and the convergence plots show that performance degrades gracefully away from the optimal value rather than collapsing. This reinforces the finding from the surface plots: while η matters, a near-optimal value suffices for strong results, and the method is not overly sensitive to its precise setting.

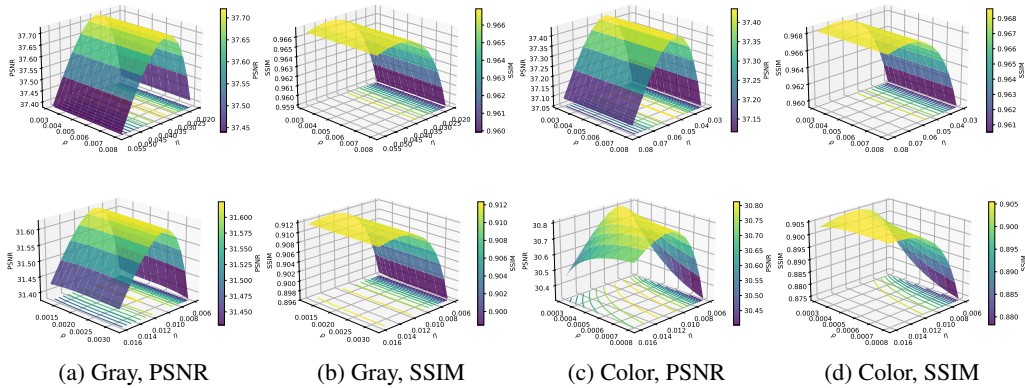


Figure 17: Sensitivity of FedDeblur to the regularization parameter η and the ADMM penalty parameter ρ . First row: complete observation with $n = 5$. Second row: partial observation with synthesized kernels.

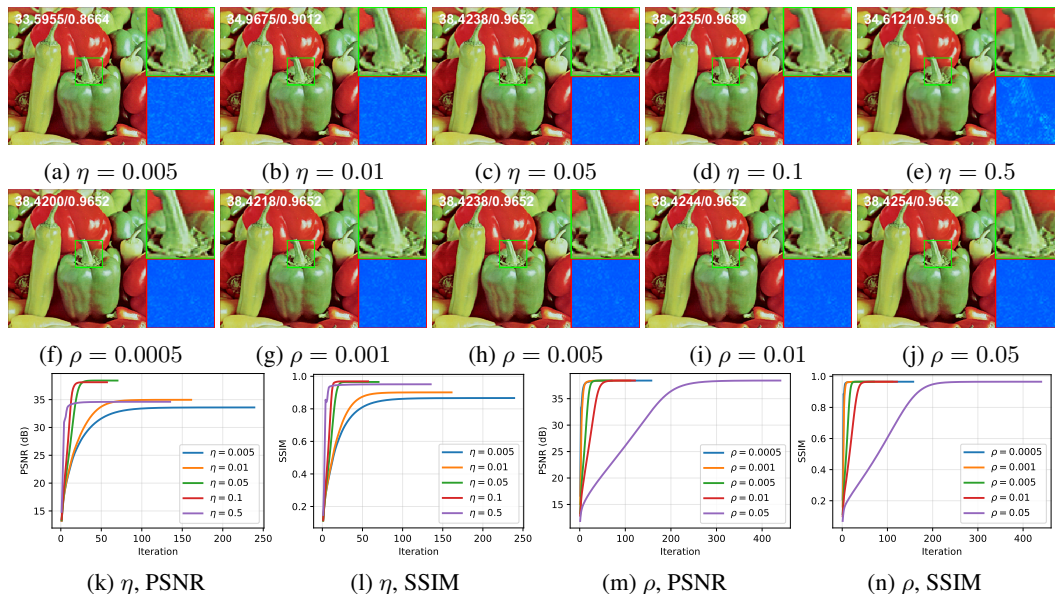


Figure 18: Visual and numerical analysis of hyperparameter sensitivity for the complete observation setting with $n = 5$ on the Pepper image using the TV prior. (a-e) Deblurring results for different values of η with a fixed $\rho = 0.005$. (f-j) Deblurring results for different values of ρ with a fixed $\eta = 0.05$. Each image is accompanied by an enlarged patch (top right) and its residual map (bottom right). PSNR and SSIM values are displayed on the top left of each image. (k-l) Convergence curves showing the effect of varying η . (m-n) Convergence curves showing the effect of varying ρ .

1620 E THEORETICAL CONVERGENCE ANALYSIS

1622 E.1 ASSUMPTIONS AND PRELIMINARIES

1623 To facilitate the analysis, we denote

$$1624 f_k(\mathbf{x}_k) = \frac{1}{n} \|\mathbf{H}_k \mathbf{x}_k - \mathbf{y}_k\|_2^2, \quad k = 1, \dots, n, \quad \text{and} \quad g(\mathbf{x}) = \eta R(\mathbf{x}), \quad (11)$$

1625 where \mathbf{H}_k is the linear operator corresponding to the convolution with kernel \mathbf{h}_k . We also define
1626 the original objective function from (2) as

$$1627 P(\mathbf{x}) = \sum_{k=1}^n f_k(\mathbf{x}) + g(\mathbf{x}).$$

1628 We begin by formally stating our assumptions and then present basic properties of the objective
1629 function.

1630 **Assumption 1.** *The regularization function $R : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ is proper, lower semi-continuous,
1631 convex, and bounded below.*

1632 Note that R here can be non-smooth, such as the ℓ_1 norm. For implicit regularization, such as a PnP
1633 prior, if the denoiser satisfies appropriate conditions, it corresponds to the proximal operator of a
1634 convex function (Gribonval & Nikolova, 2020; Wei et al., 2025). In that case, Assumption 1 holds.

1635 **Property 1.** *Each local fidelity function $f_k : \mathbb{R}^d \rightarrow \mathbb{R}$ is convex and has an L -Lipschitz continuous
1636 gradient (L -smooth); that is, there exists $L > 0$ such that for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$,*

$$1637 \|\nabla f_k(\mathbf{x}_1) - \nabla f_k(\mathbf{x}_2)\|_2 \leq L \|\mathbf{x}_1 - \mathbf{x}_2\|_2.$$

1638 *Proof.* The gradient and Hessian of $f_k(\mathbf{x})$ are

$$1639 \nabla f_k(\mathbf{x}) = \frac{2}{n} \mathbf{H}_k^T (\mathbf{H}_k \mathbf{x} - \mathbf{y}_k) \quad \text{and} \quad \nabla^2 f_k(\mathbf{x}) = \frac{2}{n} \mathbf{H}_k^T \mathbf{H}_k.$$

1640 Since the Hessian $\nabla^2 f_k(\mathbf{x})$ is independent of \mathbf{x} and is always positive semi-definite, f_k is convex
1641 and its gradient is Lipschitz continuous. A valid Lipschitz constant for each client k is

$$1642 L_k = \frac{2}{n} \|\nabla^2 f_k(\mathbf{x})\|_2 = \frac{2}{n} \|\mathbf{H}_k\|_2^2.$$

1643 We can then choose $L = \max_k \{L_k\}$. □

1644 **Property 2.** *If Assumption 1 holds, then the global objective $P : \mathbb{R}^d \rightarrow \mathbb{R} \cup \{+\infty\}$ is bounded
1645 below; that is, there exists P^* with $P(\mathbf{x}) \geq P^* > -\infty$ for all $\mathbf{x} \in \mathbb{R}^d$.*

1646 *Proof.* By Assumption 1, $g(\mathbf{x})$ is bounded below, and we denote the bound by g^* . Since $f_k(\mathbf{x}) \geq 0$
1647 for all k ,

$$1648 P(\mathbf{x}) = \sum_{k=1}^n f_k(\mathbf{x}) + g(\mathbf{x}) \geq g(\mathbf{x}) \geq g^* > -\infty.$$

1649 Hence $P(\mathbf{x})$ is bounded below. □

1650 E.2 KEY LEMMAS

1651 This section presents two standard lemmas that are used in the subsequent analysis. We include their
1652 proofs for completeness.

1653 **Lemma 1** (Three-point identity). *For any vectors $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^d$, the following identity holds:*

$$1654 \|\mathbf{a} - \mathbf{c}\|_2^2 - \|\mathbf{b} - \mathbf{c}\|_2^2 = 2\langle \mathbf{a} - \mathbf{b}, \mathbf{a} - \mathbf{c} \rangle - \|\mathbf{a} - \mathbf{b}\|_2^2.$$

1674 *Proof.* We prove the identity by starting from the right-hand side and rearranging the terms:

$$\begin{aligned}
1675 & 2\langle \mathbf{a} - \mathbf{b}, \mathbf{a} - \mathbf{c} \rangle - \|\mathbf{a} - \mathbf{b}\|_2^2 = 2\langle \mathbf{a} - \mathbf{b}, \mathbf{a} - \mathbf{c} \rangle - \langle \mathbf{a} - \mathbf{b}, \mathbf{a} - \mathbf{b} \rangle \\
1676 & = \langle \mathbf{a} - \mathbf{b}, 2(\mathbf{a} - \mathbf{c}) - (\mathbf{a} - \mathbf{b}) \rangle \\
1677 & = \langle \mathbf{a} - \mathbf{b}, \mathbf{a} + \mathbf{b} - 2\mathbf{c} \rangle \\
1678 & = \langle (\mathbf{a} - \mathbf{c}) - (\mathbf{b} - \mathbf{c}), (\mathbf{a} - \mathbf{c}) + (\mathbf{b} - \mathbf{c}) \rangle \\
1679 & = \|\mathbf{a} - \mathbf{c}\|_2^2 - \|\mathbf{b} - \mathbf{c}\|_2^2.
\end{aligned}$$

1682 The last equality follows from the difference of squares identity. \square

1683 **Lemma 2** (Descent lemma). *Let the function $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be L -smooth. Then for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^d$,*

$$1684 \text{ we have } f(\mathbf{x}_1) - f(\mathbf{x}_2) - \langle \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle \leq \frac{L}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2.$$

1687 *Proof.* By the fundamental theorem of calculus,

$$1688 f(\mathbf{x}_1) - f(\mathbf{x}_2) = \int_0^1 \langle \nabla f(\mathbf{x}_2 + t(\mathbf{x}_1 - \mathbf{x}_2)), \mathbf{x}_1 - \mathbf{x}_2 \rangle dt.$$

1692 Therefore,

$$\begin{aligned}
1693 & f(\mathbf{x}_1) - f(\mathbf{x}_2) - \langle \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle \\
1694 & = \int_0^1 \langle \nabla f(\mathbf{x}_2 + t(\mathbf{x}_1 - \mathbf{x}_2)) - \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle dt.
\end{aligned}$$

1698 Applying the Cauchy-Schwarz inequality, we have

$$\begin{aligned}
1699 & \langle \nabla f(\mathbf{x}_2 + t(\mathbf{x}_1 - \mathbf{x}_2)) - \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle \\
1700 & \leq \|\nabla f(\mathbf{x}_2 + t(\mathbf{x}_1 - \mathbf{x}_2)) - \nabla f(\mathbf{x}_2)\|_2 \cdot \|\mathbf{x}_1 - \mathbf{x}_2\|_2.
\end{aligned}$$

1702 Using the L -smoothness of f , we can bound the first term:

$$1703 \|\nabla f(\mathbf{x}_2 + t(\mathbf{x}_1 - \mathbf{x}_2)) - \nabla f(\mathbf{x}_2)\|_2 \leq L \|(\mathbf{x}_2 + t(\mathbf{x}_1 - \mathbf{x}_2)) - \mathbf{x}_2\|_2 = Lt \|\mathbf{x}_1 - \mathbf{x}_2\|_2.$$

1705 Substituting this back into the integral, we get

$$1706 f(\mathbf{x}_1) - f(\mathbf{x}_2) - \langle \nabla f(\mathbf{x}_2), \mathbf{x}_1 - \mathbf{x}_2 \rangle \leq \int_0^1 Lt \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 dt = \frac{L}{2} \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2.$$

1709 This completes the proof. \square

1711 E.3 MAIN PROOF

1712 Using the notation from (11), the optimization problem (3) can be equivalently written as

$$\begin{aligned}
1713 & \min_{\mathbf{x}, \{\mathbf{x}_k\}} \sum_{k=1}^n f_k(\mathbf{x}_k) + g(\mathbf{x}) \\
1714 & \text{s.t. } \mathbf{x}_k - \mathbf{x} = \mathbf{0}, \quad k = 1, \dots, n.
\end{aligned} \tag{12}$$

1718 Its augmented Lagrangian is

$$1719 \mathcal{L}_\rho(\mathbf{x}, \{\mathbf{x}_k\}, \{\boldsymbol{\lambda}_k\}) = \sum_{k=1}^n L_k(\mathbf{x}, \mathbf{x}_k, \boldsymbol{\lambda}_k) + g(\mathbf{x}), \tag{13}$$

$$1722 \text{ where } L_k(\mathbf{x}, \mathbf{x}_k, \boldsymbol{\lambda}_k) = f_k(\mathbf{x}_k) + \langle \boldsymbol{\lambda}_k, \mathbf{x}_k - \mathbf{x} \rangle + \frac{\rho}{2} \|\mathbf{x}_k - \mathbf{x}\|_2^2.$$

1725 The ADMM updates of FedDeblur algorithm can be expressed as

$$1726 \mathbf{x}^{t+1} = \arg \min_{\mathbf{x}} \left\{ g(\mathbf{x}) + \sum_{k=1}^n \left(\langle \boldsymbol{\lambda}_k^t, \mathbf{x}_k^t - \mathbf{x} \rangle + \frac{\rho}{2} \|\mathbf{x}_k^t - \mathbf{x}\|_2^2 \right) \right\}; \tag{14}$$

$$\mathbf{x}_k^{t+1} = \arg \min_{\mathbf{x}_k} L_k(\mathbf{x}_k^{t+1}, \mathbf{x}_k, \boldsymbol{\lambda}_k^t) \quad \text{for } k = 1, \dots, n \text{ in parallel;} \quad (15)$$

$$\boldsymbol{\lambda}_k^{t+1} = \boldsymbol{\lambda}_k^t + \rho(\mathbf{x}_k^{t+1} - \mathbf{x}_k^t) \quad \text{for } k = 1, \dots, n \text{ in parallel.} \quad (16)$$

The first-order optimality conditions for the subproblems (14) and (15) are, respectively,

$$\begin{aligned} \mathbf{0} &\in \partial g(\mathbf{x}^{t+1}) - \sum_{k=1}^n (\boldsymbol{\lambda}_k^t + \rho(\mathbf{x}^{t+1} - \mathbf{x}_k^t)), \\ \nabla f_k(\mathbf{x}_k^{t+1}) + \boldsymbol{\lambda}_k^t + \rho(\mathbf{x}_k^{t+1} - \mathbf{x}_k^t) &= \mathbf{0}, \quad k = 1, \dots, n. \end{aligned}$$

By substituting the expression for $\boldsymbol{\lambda}_k$ from (16), we get

$$\sum_{k=1}^n (\boldsymbol{\lambda}_k^{t+1} - \rho(\mathbf{x}_k^{t+1} - \mathbf{x}_k^t)) \in \partial g(\mathbf{x}^{t+1}), \quad (17)$$

$$\boldsymbol{\lambda}_k^{t+1} = -\nabla f_k(\mathbf{x}_k^{t+1}), \quad k = 1, \dots, n. \quad (18)$$

Lemma 3 (Sufficient descent). *If Assumption 1 holds, then for any $t \geq 0$, the iterates satisfy*

$$\mathcal{L}_\rho^{t+1} - \mathcal{L}_\rho^t \leq -\frac{n\rho}{2} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2 - \sum_{k=1}^n \left(\frac{\rho - L}{2} - \frac{L^2}{\rho} \right) \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2^2,$$

where $\mathcal{L}_\rho^t = \mathcal{L}_\rho(\mathbf{x}^t, \{\mathbf{x}_k^t\}, \{\boldsymbol{\lambda}_k^t\})$.

Proof. We decompose the difference $\mathcal{L}_\rho^{t+1} - \mathcal{L}_\rho^t$ into three parts:

$$\mathcal{L}_\rho^{t+1} - \mathcal{L}_\rho^t = p_1^t + p_2^t + p_3^t,$$

where

$$\begin{aligned} p_1^t &:= \mathcal{L}_\rho(\mathbf{x}^{t+1}, \{\mathbf{x}_k^t\}, \{\boldsymbol{\lambda}_k^t\}) - \mathcal{L}_\rho(\mathbf{x}^t, \{\mathbf{x}_k^t\}, \{\boldsymbol{\lambda}_k^t\}), \\ p_2^t &:= \mathcal{L}_\rho(\mathbf{x}^{t+1}, \{\mathbf{x}_k^{t+1}\}, \{\boldsymbol{\lambda}_k^t\}) - \mathcal{L}_\rho(\mathbf{x}^{t+1}, \{\mathbf{x}_k^t\}, \{\boldsymbol{\lambda}_k^t\}), \\ p_3^t &:= \mathcal{L}_\rho(\mathbf{x}^{t+1}, \{\mathbf{x}_k^{t+1}\}, \{\boldsymbol{\lambda}_k^{t+1}\}) - \mathcal{L}_\rho(\mathbf{x}^{t+1}, \{\mathbf{x}_k^{t+1}\}, \{\boldsymbol{\lambda}_k^t\}). \end{aligned}$$

Bounding p_1^t . Let $G(\mathbf{x})$ be the objective for the \mathbf{x} -update. We can write it as

$$G(\mathbf{x}) = g(\mathbf{x}) - \sum_{k=1}^n \langle \boldsymbol{\lambda}_k^t + \rho \mathbf{x}_k^t, \mathbf{x} \rangle + \frac{n\rho}{2} \|\mathbf{x}\|_2^2 + \text{const.}$$

Since g is convex, $G(\mathbf{x})$ is $n\rho$ -strongly convex. By the optimality of the update (14), we have

$$G(\mathbf{x}^{t+1}) \leq G(\mathbf{x}^t) - \frac{n\rho}{2} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2.$$

Therefore,

$$p_1^t = G(\mathbf{x}^{t+1}) - G(\mathbf{x}^t) \leq -\frac{n\rho}{2} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2.$$

Bounding p_2^t . We have $p_2^t = \sum_{k=1}^n Q_k^t$, where

$$\begin{aligned} Q_k^t &= f_k(\mathbf{x}_k^{t+1}) - f_k(\mathbf{x}_k^t) + \langle \boldsymbol{\lambda}_k^t, \mathbf{x}_k^{t+1} - \mathbf{x}_k^t \rangle + \frac{\rho}{2} \left(\|\mathbf{x}_k^{t+1} - \mathbf{x}^{t+1}\|_2^2 - \|\mathbf{x}_k^t - \mathbf{x}^{t+1}\|_2^2 \right) \\ &= f_k(\mathbf{x}_k^{t+1}) - f_k(\mathbf{x}_k^t) + \langle \boldsymbol{\lambda}_k^t, \mathbf{x}_k^{t+1} - \mathbf{x}_k^t \rangle \\ &\quad + \rho \langle \mathbf{x}_k^{t+1} - \mathbf{x}_k^t, \mathbf{x}_k^{t+1} - \mathbf{x}^{t+1} \rangle - \frac{\rho}{2} \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2^2 \quad (\text{by Lemma 1}) \\ &= f_k(\mathbf{x}_k^{t+1}) - f_k(\mathbf{x}_k^t) + \langle \boldsymbol{\lambda}_k^{t+1}, \mathbf{x}_k^{t+1} - \mathbf{x}_k^t \rangle - \frac{\rho}{2} \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2^2 \quad (\text{by (16)}) \\ &= f_k(\mathbf{x}_k^{t+1}) - f_k(\mathbf{x}_k^t) - \langle \nabla f_k(\mathbf{x}_k^{t+1}), \mathbf{x}_k^{t+1} - \mathbf{x}_k^t \rangle - \frac{\rho}{2} \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2^2 \quad (\text{by (18)}) \\ &\leq \frac{L - \rho}{2} \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2^2 \quad (\text{by Lemma 2}). \end{aligned}$$

Thus,

$$p_2^t \leq -\frac{\rho-L}{2} \sum_{k=1}^n \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2^2.$$

Bounding p_3^t . From (16), we have $\lambda_k^{t+1} - \lambda_k^t = \rho(\mathbf{x}_k^{t+1} - \mathbf{x}_k^t)$. Using this and Property 1:

$$\begin{aligned} p_3^t &= \sum_{k=1}^n \langle \lambda_k^{t+1} - \lambda_k^t, \mathbf{x}_k^{t+1} - \mathbf{x}_k^t \rangle = \frac{1}{\rho} \sum_{k=1}^n \|\lambda_k^{t+1} - \lambda_k^t\|_2^2 \\ &= \frac{1}{\rho} \sum_{k=1}^n \|\nabla f_k(\mathbf{x}_k^{t+1}) - \nabla f_k(\mathbf{x}_k^t)\|_2^2 \leq \frac{L^2}{\rho} \sum_{k=1}^n \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2^2. \end{aligned}$$

Combining these bounds yields the result:

$$\mathcal{L}_\rho^{t+1} - \mathcal{L}_\rho^t = p_1^t + p_2^t + p_3^t \leq -\frac{n\rho}{2} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2^2 - \sum_{k=1}^n \left(\frac{\rho-L}{2} - \frac{L^2}{\rho} \right) \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2^2.$$

This completes the proof. \square

Lemma 4 (Asymptotic regularity). *If Assumption 1 holds and $\rho > 2L$, the sequence of augmented Lagrangians $\{\mathcal{L}_\rho^t\}_{t=0}^\infty$ converges, and*

$$\lim_{t \rightarrow \infty} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2 = \lim_{t \rightarrow \infty} \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2 = \lim_{t \rightarrow \infty} \|\lambda_k^{t+1} - \lambda_k^t\|_2 = \lim_{t \rightarrow \infty} \|\mathbf{x}_k^t - \mathbf{x}^t\|_2 = 0$$

for each $k = 1, \dots, n$.

Proof. First, we show that $\{\mathcal{L}_\rho^t\}$ is bounded below. By Lemma 2,

$$f_k(\mathbf{x}^t) - f_k(\mathbf{x}_k^t) \leq \langle \nabla f_k(\mathbf{x}_k^t), \mathbf{x}^t - \mathbf{x}_k^t \rangle + \frac{L}{2} \|\mathbf{x}^t - \mathbf{x}_k^t\|_2^2.$$

Using (18), this becomes

$$f_k(\mathbf{x}^t) - f_k(\mathbf{x}_k^t) \leq \langle -\lambda_k^t, \mathbf{x}^t - \mathbf{x}_k^t \rangle + \frac{L}{2} \|\mathbf{x}^t - \mathbf{x}_k^t\|_2^2.$$

Thus, for $\rho > L$,

$$\begin{aligned} \mathcal{L}_\rho^t &= g(\mathbf{x}^t) + \sum_{k=1}^n \left(f_k(\mathbf{x}_k^t) + \langle \lambda_k^t, \mathbf{x}_k^t - \mathbf{x}^t \rangle + \frac{\rho}{2} \|\mathbf{x}_k^t - \mathbf{x}^t\|_2^2 \right) \\ &\geq g(\mathbf{x}^t) + \sum_{k=1}^n f_k(\mathbf{x}^t) \geq P^* > -\infty \quad (\text{by Property 2}). \end{aligned}$$

This shows that the sequence $\{\mathcal{L}_\rho^t\}$ is bounded below. From Lemma 3, if ρ is large enough such that $\frac{\rho-L}{2} - \frac{L^2}{\rho} > 0$ (which holds when $\rho > 2L$), then $\{\mathcal{L}_\rho^t\}$ is monotonically decreasing. So it converges.

The convergence of $\{\mathcal{L}_\rho^t\}$ implies $\lim_{t \rightarrow \infty} (\mathcal{L}_\rho^t - \mathcal{L}_\rho^{t+1}) = 0$. From the inequality in Lemma 3, this requires that both non-negative terms on the right-hand side go to zero:

$$\lim_{t \rightarrow \infty} \|\mathbf{x}^{t+1} - \mathbf{x}^t\|_2 = 0 \quad \text{and} \quad \lim_{t \rightarrow \infty} \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2 = 0, \quad \forall k.$$

The L -smoothness of f_k implies

$$\|\lambda_k^{t+1} - \lambda_k^t\|_2 = \|\nabla f_k(\mathbf{x}_k^{t+1}) - \nabla f_k(\mathbf{x}_k^t)\|_2 \leq L \|\mathbf{x}_k^{t+1} - \mathbf{x}_k^t\|_2,$$

so

$$\lim_{t \rightarrow \infty} \|\lambda_k^{t+1} - \lambda_k^t\|_2 = 0.$$

Finally, since

$$\mathbf{x}_k^{t+1} - \mathbf{x}^{t+1} = \frac{1}{\rho} (\lambda_k^{t+1} - \lambda_k^t),$$

1836 the consensus residual also vanishes:

$$1837 \lim_{t \rightarrow \infty} \|\mathbf{x}_k^t - \mathbf{x}^t\|_2 = 0.$$

1840 This completes the proof. \square

1841 **Theorem 2.** *Let $\{(\mathbf{x}^t, \{\mathbf{x}_k^t\}, \{\boldsymbol{\lambda}_k^t\})\}$ be the sequence generated by iterations (14-16). If Assump-*
 1842 *tion 1 holds and $\rho > 2L$, then any accumulation point $(\mathbf{x}^\infty, \{\mathbf{x}_k^\infty\}, \{\boldsymbol{\lambda}_k^\infty\})$ of this sequence is a*
 1843 *Karush-Kuhn-Tucker (KKT) point of the optimization problem (12), satisfying*

$$1845 \mathbf{x}_k^\infty = \mathbf{x}^\infty, \quad k = 1, \dots, n;$$

$$1846 \nabla f_k(\mathbf{x}_k^\infty) + \boldsymbol{\lambda}_k^\infty = \mathbf{0}, \quad k = 1, \dots, n;$$

$$1847 \mathbf{0} \in \partial g(\mathbf{x}^\infty) - \sum_{k=1}^n \boldsymbol{\lambda}_k^\infty.$$

1851 *Proof.* Let $(\mathbf{x}^\infty, \{\mathbf{x}_k^\infty\}, \{\boldsymbol{\lambda}_k^\infty\})$ be an accumulation point of the sequence. Then there exists a
 1852 subsequence, indexed by $\{t_s\}_{s=1}^\infty$, such that

$$1853 \lim_{s \rightarrow \infty} (\mathbf{x}^{t_s}, \{\mathbf{x}_k^{t_s}\}, \{\boldsymbol{\lambda}_k^{t_s}\}) = (\mathbf{x}^\infty, \{\mathbf{x}_k^\infty\}, \{\boldsymbol{\lambda}_k^\infty\}).$$

1856 From Lemma 4, the consensus residual converges to zero, $\lim_{t \rightarrow \infty} (\mathbf{x}_k^t - \mathbf{x}^t) = \mathbf{0}$. Taking the limit
 1857 along the subsequence gives

$$1858 \mathbf{x}_k^\infty - \mathbf{x}^\infty = \mathbf{0}, \quad \forall k.$$

1859 The optimality condition for the \mathbf{x}_k -update (18) gives $\nabla f_k(\mathbf{x}_k^{t_s+1}) + \boldsymbol{\lambda}_k^{t_s+1} = \mathbf{0}$. As f_k is L -
 1860 smooth, its gradient ∇f_k is continuous. Taking the limit as $s \rightarrow \infty$ and using the results of Lemma 4,
 1861 $\mathbf{x}_k^{t_s} \rightarrow \mathbf{x}_k^\infty, \boldsymbol{\lambda}_k^{t_s} \rightarrow \boldsymbol{\lambda}_k^\infty$, then

$$1862 \nabla f_k(\mathbf{x}_k^\infty) + \boldsymbol{\lambda}_k^\infty = \mathbf{0}, \quad \forall k.$$

1864 Finally, from the optimality condition for the \mathbf{x} -update (17), we have

$$1865 \sum_{k=1}^n (\boldsymbol{\lambda}_k^{t_s+1} - \rho(\mathbf{x}_k^{t_s+1} - \mathbf{x}_k^{t_s})) \in \partial g(\mathbf{x}^{t_s+1}).$$

1869 As $s \rightarrow \infty$, the left-hand side converges to $\sum_{k=1}^n \boldsymbol{\lambda}_k^\infty$ due to Lemma 4. Since g is proper, convex,
 1870 and lower semi-continuous, its subdifferential operator ∂g has a closed graph. Therefore, we can
 1871 take the limit inside the subdifferential, which yields

$$1872 \sum_{k=1}^n \boldsymbol{\lambda}_k^\infty \in \partial g(\mathbf{x}^\infty).$$

1876 This confirms that the accumulation point is a KKT point of the original problem. \square

1878 STATEMENT ON THE USE OF LARGE LANGUAGE MODELS

1879 In adherence to the conference guidelines, we clarify the role of Large Language Models (LLMs)
 1880 in the preparation of this manuscript. We utilized an LLM as a general-purpose assistive tool, pri-
 1881 marily for language refinement and grammatical correction. The core scientific contributions of this
 1882 work—including the initial research ideation, the formulation of the federated multi-observation
 1883 deblurring problem, the development of the FedDeblur framework, all theoretical analyses, the
 1884 design and execution of experiments, and the interpretation of results—were conceived and carried
 1885 out exclusively by the human authors. The LLM was employed to improve the clarity and readabil-
 1886 ity of the text. The authors take full responsibility for all content presented in this paper, including
 1887 its scientific accuracy and integrity. The LLM is not considered an author of this work.