

Robust Matrix Completion Based on Factorization and Truncated-Quadratic Loss Function

Zhi-Yong Wang^{id}, Xiao Peng Li^{id}, and Hing Cheung So^{id}, *Fellow, IEEE*

Abstract—Robust matrix completion refers to recovering a low-rank matrix given a subset of the entries corrupted by gross errors, and has various applications since many real-world signals can be modeled as low-rank matrices. Most of the existing methods only perform well for noise-free data or those with zero-mean white Gaussian noise, and their performance will be degraded in the presence of outliers. In this paper, based on the factorization framework, we propose a novel robust matrix completion scheme via using the truncated-quadratic loss function, which is non-convex and non-smooth, and half-quadratic theory is adopted for its optimization. By introducing an auxiliary variable, half-quadratic optimization (HO) can transform the loss function into two tractable forms, that is, additive and multiplicative formulations. Block coordinate descent method is then exploited as their solver. Compared with the additive form, the multiplicative variant has lower computational cost since we attempt to take the observations contaminated by outliers as missing entries. Numerical simulations and experimental results based on image inpainting and hyperspectral image recovery demonstrate that our algorithms are superior to the state-of-the-art methods in terms of restoration accuracy and runtime. MATLAB code is available at <https://github.com/bestzywang>.

Index Terms—Robust low-rank matrix completion, outlier, non-convex and non-smooth loss function, block coordinate descent.

I. INTRODUCTION

LOW-RANK matrix completion [1], [2], [3] is an important approach to recover the missing entries given an incomplete matrix. It has been widely used in numerous areas, e.g., hyperspectral remote sensing [4], [5], collaborative filtering [6], [7], image inpainting [8], [9], multi-task learning [10], system identification [11] and deep learning [12], because many real-world data have a low-rank or approximately low-rank structure. For instance, matrix completion can be used to restore the missing pixels of images since their main information often lies in a lower-dimensional subspace [13].

Intuitively, constrained rank minimization [14] can be utilized to match an incomplete matrix, and recover its

missing entries. Nevertheless, the model is NP-hard since the rank function is discrete. Nuclear norm as the convex relaxation of the rank function is suggested to practically address this problem, and the corresponding theoretical analysis can be found in [14] and [15]. To solve nuclear norm minimization, schemes such as singular value thresholding (SVT) [16], fixed point continuation with approximating singular value decomposition (FPCA) [17], and accelerated proximal gradient with linesearch (APGL) [18], are proposed. However, full singular value decomposition (SVD) calculation is needed at each iteration, leading to a huge computational cost, particularly for large-size data.

To avoid performing full SVD, approaches based on reduced rank approximation or truncated SVD, including singular value projection (SVP) [19], normalized iterative hard thresholding (NIHT) [20], and alternating projection (AP) [21], are developed. Since truncated SVD is performed at each iteration, their computational cost can be greatly reduced especially when the assumed rank r is much smaller than the matrix column/row length. In addition, low-rank matrix factorization based methods such as low-rank matrix fitting (LMaFit) [22] and alternating minimization for matrix completion (AltMin-complete) [23] have been proposed, whose idea is to replace the target matrix with the product of two rank- r matrices. Moreover, Zhu et al. [24], [25] have analyzed that although the matrix completion problem based on factorization is non-convex, there are no spurious local minima and the global optimality can be attained under some mild conditions.

Conventionally, the derivation of matrix completion methods is based on the Euclidean space. The forementioned schemes can work well under the assumption of data without noise or corrupted by white Gaussian noise. However, non-Gaussian gross errors occur in many practical scenarios [26], [27]. For example, the salt-and-pepper noise is a common impulsive disturbance in images [28]. The recovery performance of traditional matrix completion algorithms will be degraded when the observations contain outliers. Hence, it is necessary to develop recovery schemes to resist outliers. In this study, outliers and gross errors refer to sparse corruptions whose magnitudes are much bigger than those of their vicinity. In [29], the authors replace the ℓ_2 -norm with the ℓ_p -norm ($0 < p < 2$) to achieve robustness, and have proposed two algorithms, namely, iterative ℓ_p -regression and ℓ_p -alternating direction method of multipliers (ℓ_p -ADMM). The ℓ_p -norm has also been applied in low-rank tensor completion and robust PCA in [8] and [30], respectively. Although they attain

Manuscript received 22 June 2022; revised 22 September 2022; accepted 11 October 2022. Date of publication 13 October 2022; date of current version 5 April 2023. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project CityU 11207922. This article was recommended by Associate Editor S. Du. (Corresponding author: Hing Cheung So.)

The authors are with the Department of Electrical Engineering, City University of Hong Kong, Hong Kong, China (e-mail: z.y.wang@my.cityu.edu.hk; x.p.li@my.cityu.edu.hk; hceso@ee.cityu.edu.hk).

This article has supplementary material provided by the authors and color versions of one or more figures available at <https://doi.org/10.1109/TCSVT.2022.3214583>.

Digital Object Identifier 10.1109/TCSVT.2022.3214583

1051-8215 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See <https://www.ieee.org/publications/rights/index.html> for more information.

outstanding performance in the presence of outliers, there are two open questions. The first issue is to decide p because its appropriate value varies with the intensity of impulsive noise. Second, when $0 < p < 1$, ℓ_p -norm is non-convex and non-smooth, which results in a challenging optimization problem. Furthermore, a new surrogate of ℓ_0 -‘norm’ is devised to achieve low-rank recovery and combat outliers, but it introduces two additional user-defined parameters [31] and it is vital to choose their proper values. Two new robust loss functions are put forth in [32] to reduce the impact of outliers on the recovery performance, and they use a distributed optimization framework to tackle matrix completion. Huber function [33] as a common M-estimator is also applied on matrix completion to resist against gross errors and majorization minimization (MM) algorithm is used to jointly estimate its parameter vector and scale. Moreover, robust matrix factorization based on MM (RMF-MM) [34] is developed. The authors replace the ℓ_2 -norm with the ℓ_1 -norm to attain robustness, find a convex surrogate to relax the non-smooth and non-convex objective function, and then employ the MM technique as the solver. Recently, He et al. [37] adopt the non-convex Welsch cost function to handle outliers, and propose two robust matrix completion algorithms, i.e., half-quadratic power factorization (HQ-PF) and half-quadratic alternating steepest descent (HQ-ASD). Furthermore, Bayesian methods, including variational Bayesian matrix factorization based on ℓ_1 -norm (VBMFL1) [35] and Bayesian estimator for noisy matrix completion (BENMC) [36], are suggested.

Truncated-quadratic loss function is non-convex and non-smooth, and has been utilized as the regularization term for edge-detection and denoising [38], [39], [43]. However, it is exploited as a non-convex loss function to resist outliers in this paper. Research results [32], [33], [37], [40] have shown that compared with the convex loss functions such as ℓ_1 -norm and Huber function, whose influence functions are not re-descending, the non-convex penalty function can suppress large outliers well. Intuitively, truncated-quadratic function as the cost function is more robust than the ℓ_p -norm and Huber function especially for large outliers, because the corresponding weights in the former will be assigned to zero. Besides, it is more preferable that loss functions only assign smaller weights for outliers, while keeping the weights of normal data fixed. Compared with other non-convex loss functions [41], i.e., Cauchy and Welsch, which down-weight all observations including uncorrupted data, the truncated-quadratic function is better to combat outliers since it only down-weights the entries contaminated by outliers. However, its non-smoothness and non-convexity create challenges in the optimization process.

In this study, to avoid full SVD calculation and resist gross errors, the factorization idea and truncated-quadratic cost function are jointly leveraged for robust matrix completion. Moreover, since it is difficult for MM technique to find the surrogate function [42], half-quadratic optimization (HO) theory [43], [44] is utilized. In order to seek the equivalent form of the original optimization problem, a new auxiliary term is

introduced in the HO formulation to transform the non-convex and non-smooth problem into a tractable quadratic optimization problem, which permits to derive an efficient algorithm based on alternating minimization between the matrix entries and auxiliary variable. It is worth pointing out that from the HO theory, the truncated-quadratic function can be converted into two different optimization forms, namely, the additive and multiplicative formulations. Accordingly, two algorithms are devised, called HO based on additive form for truncated-quadratic function (HOAT) and HO based on multiplicative form for truncated-quadratic function (HOMT). In summary, the main contributions of our paper are as follows:

- 1) We derive the *dual* function and minimizer function of the truncated-quadratic function based on the HO theory.
- 2) We devise two robust factorization based matrix completion algorithms, namely, HOAT and HOMT, to resist gross errors. In particular, the observations detected as outliers are considered as missing entries in HOMT.
- 3) Experiments demonstrate that compared to the competing methods, the proposed algorithms exhibit better recovery performance for different real-world data including gray-scale and hyperspectral images, in the presence of outliers and Gaussian noise.

The remainder of this paper is organized as follows. In Section II, notations are provided, and matrix completion methods as well as HO theory are reviewed. The two developed approaches and their theoretical analysis, are presented in Section III. Section IV discusses the numerical simulation results based on synthetic data and real images. Finally, conclusions are drawn in Section V.

II. PRELIMINARIES

In this section, notations, related works and HO theory are reviewed.

A. Notations

Italic, bold lower-case and bold upper-case letters denote scalars, vectors and matrices, respectively. The i th entry of a vector \mathbf{a} is represented by a_i , and the (i, j) element of a matrix \mathbf{A} is denoted by \mathbf{A}_{ij} . In particular, a matrix of zeros and ones are signified by $\mathbf{0}$ and $\mathbf{1}$, respectively. Besides, $\Omega \subset \{1, \dots, m\} \times \{1, \dots, n\}$ is used to denote the index set of the observed entries of \mathbf{A} , Ω^c is the complement of Ω , and $(\cdot)_{\Omega}$ is a projection operator, defined as:

$$[\mathbf{A}_{\Omega}]_{ij} = \begin{cases} \mathbf{A}_{ij}, & \text{if } (i, j) \in \Omega \\ 0, & \text{if } (i, j) \in \Omega^c \end{cases}$$

In addition, $\|\mathbf{a}\|_2 = \sqrt{\mathbf{a}^T \mathbf{a}}$ is the ℓ_2 -norm of $\mathbf{a} \in \mathbb{R}^m$, $\|\mathbf{A}\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n \mathbf{A}_{ij}^2}$ is the Frobenius norm of $\mathbf{A} \in \mathbb{R}^{m \times n}$, and $\text{vec}(\mathbf{A})$ is a vector generated via vectorizing \mathbf{A} . Moreover, $\mathbf{A}_{j,:}$ and $\mathbf{A}_{:,j}$ represent the j th row and j th column of \mathbf{A} , respectively. Furthermore, $|a|$ stands for the absolute value of a , while $|\mathcal{I}|$ is the cardinality of the set \mathcal{I} . Finally, the pseudo-inverse operator and transpose operator are denoted by $(\cdot)^{\dagger}$ and $(\cdot)^T$, respectively.

B. Related Work

1) *Matrix Completion*: To obtain a low-rank matrix \mathbf{M} to match the incomplete matrix \mathbf{X}_Ω , a direct formulation is [14]:

$$\min_{\mathbf{M}} \text{rank}(\mathbf{M}), \text{ s.t. } \mathbf{M}_\Omega = \mathbf{X}_\Omega \quad (1)$$

However, (1) is a NP-hard optimization problem. Thereby, nuclear norm [15] is suggested, leading to

$$\min_{\mathbf{M}} \|\mathbf{M}\|_*, \text{ s.t. } \mathbf{M}_\Omega = \mathbf{X}_\Omega \quad (2)$$

where the nuclear norm $\|\mathbf{M}\|_*$ is the sum of singular values of \mathbf{M} . Nevertheless, it requires performing full SVD at each iteration, which limits its application on large-size data. To avoid full SVD calculation, matrix factorization [22] is applied, resulting in:

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X}_\Omega - (\mathbf{UV})_\Omega\|_F^2 \quad (3)$$

where $\mathbf{U} \in \mathbb{R}^{m \times r}$ and $\mathbf{V} \in \mathbb{R}^{r \times n}$ are the two small size matrices with rank $r \ll \min(m, n)$. Apparently, $\mathbf{M} = \mathbf{UV}$ has low rank. However, its performance will be degraded in the presence of outliers.

On the other hand, in order to achieve robustness, a modified model [29] based on (3) via employing ℓ_p -norm with $0 < p < 2$, is proposed:

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X}_\Omega - (\mathbf{UV})_\Omega\|_p^p \quad (4)$$

For the entry $(i, j) \in \Omega$ corrupted by outliers, its residual error $e = \mathbf{X}_{i,j} - \mathbf{M}_{i,j}$ in the ℓ_p -norm space is less than the error in the Frobenius norm space, i.e., $|e|^p < |e|^2$ with $0 < p < 2$. Thus, (4) can reduce the impact of outliers, and the ℓ_p -norm is more capable to resist gross errors than the Frobenius norm.

In addition, robust M-estimation based matrix completion method [33] is developed, formulated as:

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X}_\Omega - (\mathbf{UV})_\Omega\|_{\sigma, c} \quad (5)$$

where $\|\mathbf{X}_\Omega - (\mathbf{UV})_\Omega\|_{\sigma, c} = \sum_{(i,j) \in \Omega} \rho\left(\frac{\mathbf{X}_{i,j} - (\mathbf{UV})_{i,j}}{\sigma}\right)$ with $(i, j) \in \Omega$, and $\rho(\cdot)$ is the Huber function:

$$\rho(x) = \begin{cases} \frac{1}{2}x^2, & |x| \leq c \\ c|x| - \frac{1}{2}c^2, & |x| > c \end{cases} \quad (6)$$

However, Huber function is a ‘monotone M-estimator’ [49], namely, large outliers still affect its robustness, but the effect remains bounded.

2) *Half-Quadratic Theory*: HO is an efficient tool for convex and non-convex optimization based on the conjugate function theory. Let $\phi(x)$ denote a loss function that satisfies the conditions of HO theory [43], [44], [45], and $\phi(x)$ can be reformulated as the following model:

$$\phi(x) = \inf_y Q(x, y) + \varphi(y) \quad (7)$$

where y is an introduced auxiliary variable, $\varphi(y)$ is the dual function of $\phi(x)$, and $Q(x, y)$ is a quadratic function of x , which has two forms:

$$Q(x, y) = \frac{(x - y)^2}{2}, \quad y \in \mathbb{R} \quad (8)$$

and

$$Q(x, y) = \frac{y \cdot x^2}{2}, \quad y \in \mathbb{R}_+ \quad (9)$$

Accordingly, $\phi(x)$ has two expressions, that is, the additive and multiplicative forms:

$$\phi(x) = \inf_y \frac{(x - y)^2}{2} + \varphi(y) \quad (10)$$

$$\phi(x) = \inf_y \frac{y \cdot x^2}{2} + \varphi(y) \quad (11)$$

In addition, for a fixed x in (10) and (11), there exists the corresponding minimizer function $\delta_A(x)$ and $\delta_M(x)$:

$$\delta_A(x) := \arg \inf_y \frac{(x - y)^2}{2} + \varphi(y) \quad (12)$$

and

$$\delta_M(x) := \arg \inf_y \frac{y \cdot x^2}{2} + \varphi(y). \quad (13)$$

Moreover, if $\delta_A(x)$ and $\delta_M(x)$ exist, the minimization of loss function $\phi(x)$ is equal to:

$$\min_{x, y} Q(x, y) + \varphi(y) \quad (14)$$

which can be solved by alternating minimization. In other words, for a given x , the optimal solution to y is obtained via (12) or (13). While given y , (14) can also be easily solved since it is a quadratic optimization problem.

III. PROPOSED ALGORITHMS

In this section, we first derive the minimizer functions of the additive and multiplicative forms according to the definition of HO theory, and their corresponding robust matrix completion methods are then devised. In particular, the convergence and computational complexity of the developed algorithms are analyzed.

A. Robust Matrix Completion via Additive Form

The truncated-quadratic function has the form of [38]:

$$\phi(x) = \begin{cases} \frac{x^2}{2}, & |x| < e \\ \frac{e^2}{2}, & |x| \geq e \end{cases} \quad (15)$$

Fig. 1 plots the truncated-quadratic function and typical loss functions. It is seen that the truncated-quadratic function expresses a wish to suppress big outliers, compared with other functions. However, it is non-convex and non-smooth, which results in an intractable optimization problem. In this study, HO theory is exploited to transform (15) into (10) via adding an auxiliary variable y . According to Section II-B.2, we obtain the minimizer function of the additive form of (15) first, i.e.,

$$\delta_A(x) = \begin{cases} 0, & |x| < e \\ x, & |x| \geq e \end{cases} \quad (16)$$

whose derivation is shown in Appendix A.

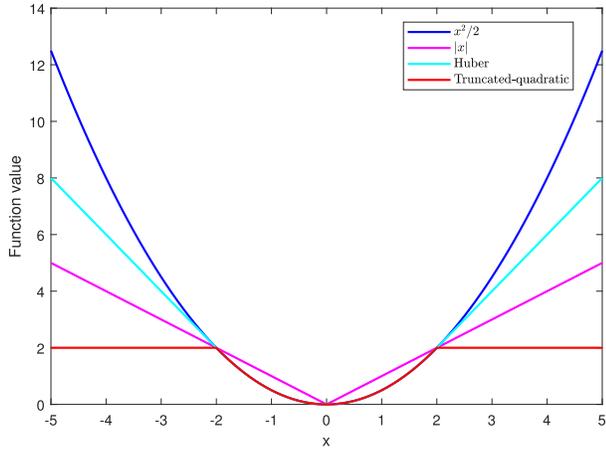


Fig. 1. Different loss functions with $c = 2$ in Huber function and $e = 2$ in truncated-quadratic function.

Motivated by (4) and (5), the truncated-quadratic function is adopted to achieve robustness in the presence of gross errors, formulated as:

$$\min_{\mathbf{U}, \mathbf{V}} \|\mathbf{X}_\Omega - (\mathbf{UV})_\Omega\|_\phi \quad (17)$$

Since $\|\cdot\|_\phi$ is an entry-wise operator, (17) can be reformulated as:

$$\min_{\mathbf{U}, \mathbf{V}} \sum_{(i,j) \in \Omega} \phi(\mathbf{X}_{i,j} - (\mathbf{UV})_{i,j}) \quad (18)$$

Based on the additive form of HO theory in (10), (18) becomes:

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{N}} \sum_{(i,j) \in \Omega} \frac{1}{2} (\mathbf{X}_{i,j} - (\mathbf{UV})_{i,j} - \mathbf{N}_{i,j})^2 + \varphi(\mathbf{N}_{i,j}) \quad (19)$$

where $\mathbf{N}_{i,j}$ is the auxiliary variable to model the outlier component of noise, and is determined by the minimizer function $\delta_A(\cdot)$ in (16). Besides, $\varphi(\mathbf{N}_{i,j})$ is the regularization term, whose expression is provided in Appendix A. We denote $\varphi(\mathbf{N}) = \sum_{(i,j)} \varphi(\mathbf{N}_{i,j})$, and then (19) can be rewritten as the matrix form:

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{N}} \frac{1}{2} \|\mathbf{X}_\Omega - (\mathbf{UV})_\Omega - \mathbf{N}_\Omega\|_F^2 + \varphi(\mathbf{N}_\Omega) \quad (20)$$

It is worth mentioning that (20) has three variables without constraints, and block coordinate descent (BCD) [46], [47] is adopted to find its solutions, whose idea is to optimize one factor while fixing the remaining two factors. Thus, (20) can be decomposed as the following three subproblems:

$$\mathbf{U}^{k+1} = \arg \min_{\mathbf{U}} \frac{1}{2} \|\mathbf{X}_\Omega - (\mathbf{UV}^k)_\Omega - \mathbf{N}_\Omega^k\|_F^2 \quad (21)$$

$$\mathbf{V}^{k+1} = \arg \min_{\mathbf{V}} \frac{1}{2} \|\mathbf{X}_\Omega - (\mathbf{U}^{k+1}\mathbf{V})_\Omega - \mathbf{N}_\Omega^k\|_F^2 \quad (22)$$

$$\mathbf{N}^{k+1} = \arg \min_{\mathbf{N}} \frac{1}{2} \|\mathbf{X}_\Omega - (\mathbf{U}^{k+1}\mathbf{V}^{k+1})_\Omega - \mathbf{N}_\Omega\|_F^2 + \varphi(\mathbf{N}_\Omega) \quad (23)$$

It is easy to know that given \mathbf{V}^k and \mathbf{N}^k , (21) can be decoupled as m subproblems with respect to (w.r.t.) \mathbf{u}_i for $i = 1, \dots, m$:

$$\mathbf{u}_i^{k+1} = \arg \min_{\mathbf{u}_i} \frac{1}{2} \|\mathbf{X}_{i, \mathcal{J}_i} - \mathbf{u}_i^T \mathbf{V}_{\mathcal{J}_i}^k - \mathbf{N}_{i, \mathcal{J}_i}^k\|_2^2 \quad (24)$$

where \mathbf{u}_i^T represents the i th row of \mathbf{U} , i.e.,

$$\mathbf{U} = \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \\ \vdots \\ \mathbf{u}_m^T \end{bmatrix} \quad (25)$$

and \mathcal{J}_i is denoted as the set of column indices of the observed entries in the i th row of \mathbf{X} with $\sum_{i=1}^m |\mathcal{J}_i| = |\Omega|$. Herein, we provide an example to determine \mathcal{J}_i . If $\mathbf{X}_{i,:} = [x_{i,1} \ 0 \ x_{i,3} \ 0 \ x_{i,5}]$, then $\mathcal{J}_i = \{1, 3, 5\}$, and $|\mathcal{J}_i| = 3$. Besides, $\mathbf{V}_{\mathcal{J}_i}^k \in \mathbb{R}^{r \times |\mathcal{J}_i|}$ contains $|\mathcal{J}_i|$ columns indexed by \mathcal{J}_i , namely,

$$\mathbf{V}_{\mathcal{J}_i}^k = [\mathbf{V}_{:, \mathcal{J}_{i1}} \ \mathbf{V}_{:, \mathcal{J}_{i2}} \ \cdots \ \mathbf{V}_{:, \mathcal{J}_{i|\mathcal{J}_i|}}] \quad (26)$$

where \mathcal{J}_{ij} stands for the j th entry of \mathcal{J}_i with $j = 1, \dots, |\mathcal{J}_i|$. Similarly,

$$\mathbf{X}_{i, \mathcal{J}_i} = [\mathbf{X}_{i, \mathcal{J}_{i1}} \ \mathbf{X}_{i, \mathcal{J}_{i2}} \ \cdots \ \mathbf{X}_{i, \mathcal{J}_{i|\mathcal{J}_i|}}] \quad (27)$$

and

$$\mathbf{N}_{i, \mathcal{J}_i}^k = [\mathbf{N}_{i, \mathcal{J}_{i1}}^k \ \mathbf{N}_{i, \mathcal{J}_{i2}}^k \ \cdots \ \mathbf{N}_{i, \mathcal{J}_{i|\mathcal{J}_i|}}^k] \quad (28)$$

Thus, (24) is rewritten as:

$$\mathbf{u}_i^{k+1} = \arg \min_{\mathbf{u}_i} \frac{1}{2} \|\mathbf{D}_{i, \mathcal{J}_i}^k - \mathbf{u}_i^T \mathbf{V}_{\mathcal{J}_i}^k\|_2^2 \quad (29)$$

where $\mathbf{D}^k = \mathbf{X} - \mathbf{N}^k$. It is clear that (29) is a least squares problem, whose closed-form solution is:

$$\mathbf{u}_i^{k+1} = \left((\mathbf{V}_{\mathcal{J}_i}^k)^T \right)^\dagger (\mathbf{D}_{i, \mathcal{J}_i}^k)^T \quad (30)$$

and its computational complexity is $\mathcal{O}(|\mathcal{J}_i|r^2)$.

Problem (22) has a similar structure to (21), and hence can be solved in an analogous manner. Given \mathbf{U}^{k+1} and \mathbf{N}^k , it is easy to observe that (22) can be decoupled w.r.t. \mathbf{v}_j for $j = 1, \dots, n$, leading to:

$$\mathbf{v}_j^{k+1} = \arg \min_{\mathbf{v}_j} \frac{1}{2} \|\mathbf{D}_{\mathcal{I}_j, j}^k - \mathbf{U}_{\mathcal{I}_j}^{k+1} \mathbf{v}_j\|_2^2 \quad (31)$$

where \mathbf{v}_j is the j th column of \mathbf{V} , \mathcal{I}_j is the set of row indices of the observed entries in the j th column of \mathbf{X} , and $\sum_{j=1}^n |\mathcal{I}_j| = |\Omega|$. Apparently, (31) is also a least squares problem, and its closed-form solution is:

$$\mathbf{v}_j^{k+1} = \left(\mathbf{U}_{\mathcal{I}_j}^{k+1} \right)^\dagger \mathbf{D}_{\mathcal{I}_j, j}^k \quad (32)$$

whose computational complexity is $\mathcal{O}(|\mathcal{I}_j|r^2)$.

Finally, given \mathbf{U}^{k+1} and \mathbf{V}^{k+1} , (23) is equal to:

$$\mathbf{N}^{k+1} = \arg \min_{\mathbf{N}} \frac{1}{2} \|\mathbf{R}_\Omega^{k+1} - \mathbf{N}_\Omega\|_F^2 + \varphi(\mathbf{N}_\Omega) \quad (33)$$

where $\mathbf{R}^{k+1} = \mathbf{X} - \mathbf{U}^{k+1}\mathbf{V}^{k+1}$ represents the residual information in the $(k + 1)$ th iteration. In accordance to the HO theory, the solution to \mathbf{N}^{k+1} is:

$$\mathbf{N}^{k+1} = \delta_{\Lambda}(\mathbf{R}^{k+1}) \quad (34)$$

As shown in (16), the minimizer function $\delta_{\Lambda}(x)$ depends on the parameter e which controls robustness. That is to say, any entry in \mathbf{R}_{Ω} bigger than e , will be considered as outliers. For a smaller e , (19) will reduce the impact of all the outliers, but also mistakenly regard the ‘normal’ entries as outliers. While for a larger e , it is possible that some outliers remain in the data, leading to the recovery performance degradation. For ‘normal’ entries, they refer to the elements in the data corrupted by zero-mean Gaussian noise, which are not the gross errors. Analogous to [48], we take the following strategy to choose the value of e :

$$e^k = \min \left\{ \zeta \sigma^k, e^{k-1} \right\} \quad (35)$$

where $\zeta > 0$ is a constant and σ^k is the robust normalized median absolute deviation of the vectorized \mathbf{R}_{Ω} , i.e.,

$$\sigma^k = 1.4815 \times \text{Med}(|\text{vec}(\mathbf{R}_{\Omega}^k) - \text{Med}(\text{vec}(\mathbf{R}_{\Omega}^k))|) \quad (36)$$

with $\text{Med}(\cdot)$ being the sample median operator [49]. Besides, in order to achieve reliable recovery, (3) is first handled to provide a good initialization for the calculation of (21), although it is vulnerable to outliers, which is a similar strategy adopted in [37]. Actually, the solutions to \mathbf{U} and \mathbf{V} in (3) are same as those of (21) and (22), respectively, when $\mathbf{N} = \mathbf{0}$ at each iteration. Hence, the initialized estimates of \mathbf{U} and \mathbf{V} in (3) are computed as:

$$\mathbf{u}_i^{k+1} = \left(\left(\mathbf{V}_{\mathcal{J}_i}^k \right)^T \right)^{\dagger} \mathbf{X}_{i, \mathcal{J}_i}^T, \quad i = 1, \dots, m \quad (37)$$

$$\mathbf{v}_j^{k+1} = \left(\mathbf{U}_{\mathcal{I}_j}^{k+1} \right)^{\dagger} \mathbf{X}_{\mathcal{I}_j, j}, \quad j = 1, \dots, n \quad (38)$$

Algorithm 1 summarizes the detailed optimization procedure. As stated in [29], the convergence rate to finding the solutions to (37) and (38) is fast and it is not necessary to obtain the optimal solutions since (37) and (38) are only used to provide initialization for (30) and (34), not the final results. Thus, a small number of I_p is enough, specifically, $I_p = 3$ is employed in our algorithm. In addition, Algorithm 1 will be terminated when its iteration number exceeds I_m and/or $\kappa = \|\mathbf{X}_{\Omega} - \mathbf{M}_{\Omega}^k - \mathbf{N}^k\|_F / \|\mathbf{X}_{\Omega}\|_F$ is less than the pre-set threshold value $\eta > 0$. Note also that, the calculation of \mathbf{u}_i with $i = 1, \dots, m$ in (30) and \mathbf{v}_j with $j = 1, \dots, n$ in (32) are independent, and hence distributed and parallel realizations can be exploited to shorten computational time. Finally, the convergence of Algorithm 1 is analyzed in the following proposition. We first define $\mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{N}) = \frac{1}{2} \|\mathbf{X}_{\Omega} - (\mathbf{U}\mathbf{V})_{\Omega} - \mathbf{N}\|_F^2 + \varphi(\mathbf{N})$ as the objective function.

Proposition 1: The sequence $\{\mathcal{L}(\mathbf{U}^k, \mathbf{V}^k, \mathbf{N}^k)\}$ produced by Algorithm 1 is convergent.

Proof: Since e^k is updated at each iteration, we provide the proof via discussing the following two cases. First, when updating $e^k \rightarrow e^{k+1}$, $\varphi(\mathbf{N})$ in $\mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{N})$ is the only component related to e , and to make the development

clearer, we write $\varphi(\mathbf{N}, e) = \varphi(\mathbf{N})$. Besides, $\varphi(\mathbf{N}, e)$ is separable, namely, $\varphi(\mathbf{N}, e) = \sum_{(i,j)} \varphi(\mathbf{N}_{i,j}, e)$. Then, we have $\frac{\partial \varphi(\mathbf{N}_{i,j}, e)}{\partial e} > 0$ via taking the partial derivative of $\varphi(\mathbf{N}_{i,j}, e)$ w.r.t. e :

$$\frac{\partial \varphi(\mathbf{N}_{i,j}, e)}{\partial e} = \begin{cases} |\mathbf{N}_{i,j}|, & |\mathbf{N}_{i,j}| < e \\ e, & |\mathbf{N}_{i,j}| \geq e \end{cases} \quad (39)$$

and we obtain:

$$\frac{\partial \mathcal{L}}{\partial e} = \frac{\partial \sum_{i,j} \varphi(\mathbf{N}_{i,j}, e)}{\partial e} = \sum_{i,j} \frac{\partial \varphi(\mathbf{N}_{i,j}, e)}{\partial e} > 0 \quad (40)$$

Thus, $\mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{N})$ increases monotonically w.r.t. e , and the updating rule in (35) makes e non-increasing. Therefore, $\{\mathcal{L}(\mathbf{U}^k, \mathbf{V}^k, \mathbf{N}^k)\}$ is non-increasing when updating $e^k \rightarrow e^{k+1}$.

Second, after obtaining e^{k+1} , since alternating minimization is used at each step in Algorithm 1, we have:

$$\begin{aligned} & \mathcal{L}(\mathbf{U}^{k+1}, \mathbf{V}^{k+1}, \mathbf{N}^{k+1}) \\ &= \min_{\mathbf{V}} \frac{1}{2} \left\| \mathbf{X}_{\Omega} - \left(\mathbf{U}^{k+1} \mathbf{V} \right)_{\Omega} - \mathbf{N}^{k+1} \right\|_F^2 + \varphi(\mathbf{N}^{k+1}) \\ &\leq \min_{\mathbf{U}} \frac{1}{2} \left\| \mathbf{X}_{\Omega} - \left(\mathbf{U} \mathbf{V}^k \right)_{\Omega} - \mathbf{N}^{k+1} \right\|_F^2 + \varphi(\mathbf{N}^{k+1}) \\ &\leq \min_{\mathbf{N}} \frac{1}{2} \left\| \mathbf{X}_{\Omega} - \left(\mathbf{U}^k \mathbf{V}^k \right)_{\Omega} - \mathbf{N} \right\|_F^2 + \varphi(\mathbf{N}) \\ &\leq \frac{1}{2} \left\| \mathbf{X}_{\Omega} - \left(\mathbf{U}^k \mathbf{V}^k \right)_{\Omega} - \mathbf{N}^k \right\|_F^2 + \varphi(\mathbf{N}^k) \\ &= \mathcal{L}(\mathbf{U}^k, \mathbf{V}^k, \mathbf{N}^k) \end{aligned} \quad (41)$$

which means that $\mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{N})$ is non-increasing. Since $\mathcal{L}(\mathbf{U}, \mathbf{V}, \mathbf{N})$ is lower bounded by 0, thus the sequence $\{\mathcal{L}(\mathbf{U}^k, \mathbf{V}^k, \mathbf{N}^k)\}$ generated by Algorithm 1 is convergent. ■

Algorithm 1 HOAT

Input: Incomplete matrix \mathbf{X}_{Ω} , index set Ω , maximum allowable iteration numbers I_p and I_m , and tolerance parameter η

Initialize: Generate a standard Gaussian matrix $\mathbf{V}^0 \in \mathbb{R}^{r \times n}$, and determine $\{\mathcal{J}_i\}_{i=1}^m$ and $\{\mathcal{I}_j\}_{j=1}^n$ according to Ω .

for $p = 1, 2, \dots, I_p$ **do**

 // Fix \mathbf{V}^{p-1} , optimize \mathbf{U}

 Find \mathbf{U}^p according to (37)

 // Fix \mathbf{U}^p , optimize \mathbf{V}

 Find \mathbf{V}^p according to (38)

end for

Set $\mathbf{U}^0 = \mathbf{U}^p$ and $\mathbf{V}^0 = \mathbf{V}^p$

for $k = 1, 2, \dots, I_m$ **do**

 Calculate e^k according to (35)

 // Fix \mathbf{U}^{k-1} and \mathbf{V}^{k-1} , optimize \mathbf{N}

 Find \mathbf{N}^k according to (34)

 // Fix \mathbf{V}^{k-1} and \mathbf{N}^k , optimize \mathbf{U}

 Find \mathbf{U}^k according to (30)

 // Fix \mathbf{U}^k and \mathbf{N}^k , optimize \mathbf{V}

 Find \mathbf{V}^k according to (32)

 Stop, if a termination condition is satisfied.

end for

Output: $\mathbf{M} = \mathbf{U}^k \mathbf{V}^k$.

B. Robust Matrix Completion via Multiplicative Form

For the multiplicative form of truncated-quadratic function in (11), we derive its minimizer function as (see Appendix B):

$$\delta_M(x) = \begin{cases} 1, & |x| < e \\ 0, & |x| \geq e \end{cases} \quad (42)$$

According to (11), (18) is equal to:

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{W}} \sum_{(i,j) \in \Omega} \frac{1}{2} \mathbf{W}_{i,j} (\mathbf{X}_{i,j} - (\mathbf{UV})_{i,j})^2 + \varphi(\mathbf{W}_{i,j}) \quad (43)$$

where $\mathbf{W}_{i,j}$ is a new auxiliary variable to assign the weights to different entries, and its value depends on the minimizer function $\delta_M(\cdot)$ in (42). In addition, we define $\varphi(\mathbf{W}) = \sum_{(i,j)} \varphi(\mathbf{W}_{i,j})$, then (43) can be rewritten as:

$$\min_{\mathbf{U}, \mathbf{V}, \mathbf{W}} \frac{1}{2} \left\| \sqrt{\mathbf{W}_\Omega} \circ (\mathbf{X}_\Omega - (\mathbf{UV})_\Omega) \right\|_F^2 + \varphi(\mathbf{W}_\Omega) \quad (44)$$

where $\sqrt{\cdot}$ is the element-wise square root operator, and \circ denotes the entry-wise product. In particular, $\mathbf{W}_\Omega^c = \mathbf{0}$. BCD is also used to seek the solutions to (44), and thus we can decompose (44) into three subproblems:

$$\mathbf{U}^{k+1} = \arg \min_{\mathbf{U}} \frac{1}{2} \left\| \sqrt{\mathbf{W}_\Omega^k} \circ (\mathbf{X}_\Omega - (\mathbf{UV}^k)_\Omega) \right\|_F^2 \quad (45)$$

$$\mathbf{V}^{k+1} = \arg \min_{\mathbf{V}} \frac{1}{2} \left\| \sqrt{\mathbf{W}_\Omega^k} \circ (\mathbf{X}_\Omega - (\mathbf{U}^{k+1}\mathbf{V})_\Omega) \right\|_F^2 \quad (46)$$

$$\mathbf{W}^{k+1} = \arg \min_{\mathbf{W}} \frac{1}{2} \left\| \sqrt{\mathbf{W}_\Omega} \circ (\mathbf{X}_\Omega - (\mathbf{U}^{k+1}\mathbf{V}^{k+1})_\Omega) \right\|_F^2 + \varphi(\mathbf{W}_\Omega) \quad (47)$$

Note that (45) and (46) are weighted least squares problems, whose computational complexity is high since their weights are updated via (47) at each iteration. However, it is easy to observe that the entry of \mathbf{W}^k is either '0' or '1', thus we can set $\tilde{\Omega}^k = \mathbf{W}^k$ as the current observation index set of \mathbf{X} , that is to say, the observations contaminated by outliers are taken as missing entries. Accordingly, (45), (46) and (47) are reformulated as:

$$\mathbf{U}^{k+1} = \arg \min_{\mathbf{U}} \frac{1}{2} \left\| \mathbf{X}_{\tilde{\Omega}^k} - (\mathbf{UV}^k)_{\tilde{\Omega}^k} \right\|_F^2 \quad (48)$$

$$\mathbf{V}^{k+1} = \arg \min_{\mathbf{V}} \frac{1}{2} \left\| \mathbf{X}_{\tilde{\Omega}^k} - (\mathbf{U}^{k+1}\mathbf{V})_{\tilde{\Omega}^k} \right\|_F^2 \quad (49)$$

$$\tilde{\Omega}^{k+1} := \arg \min_{\mathbf{W}} \frac{1}{2} \left\| \sqrt{\mathbf{W}_\Omega} \circ (\mathbf{X}_\Omega - (\mathbf{U}^{k+1}\mathbf{V}^{k+1})_\Omega) \right\|_F^2 + \varphi(\mathbf{W}_\Omega) \quad (50)$$

To solve (48)-(50), we first recall that $\tilde{\mathcal{J}}_i^k$ and $\tilde{\mathcal{I}}_j^k$ are the set of column indices of the observed entries in the i th row of $\mathbf{X}_{\tilde{\Omega}^k}$ and the set of row indices of the observed entries in the j th column of $\mathbf{X}_{\tilde{\Omega}^k}$ at the k th iteration, respectively. Problem (48) can be decomposed as the following m subproblems:

$$\mathbf{u}_i^{k+1} = \arg \min_{\mathbf{u}_i} \frac{1}{2} \left\| \mathbf{X}_{i, \tilde{\mathcal{J}}_i^k} - \mathbf{u}_i^T \mathbf{V}_{\tilde{\mathcal{J}}_i^k}^k \right\|_2^2 \quad (51)$$

where $i = 1, \dots, m$, and the solution is:

$$\mathbf{u}_i^{k+1} = \left(\left(\mathbf{V}_{\tilde{\mathcal{J}}_i^k}^k \right)^T \right)^\dagger \mathbf{X}_{i, \tilde{\mathcal{J}}_i^k}^T \quad (52)$$

whose computational complexity is $\mathcal{O}(|\tilde{\mathcal{J}}_i^k|r^2)$.

Similarly, (49) can be separated into n independent subproblems:

$$\mathbf{v}_j^{k+1} = \arg \min_{\mathbf{v}_j} \frac{1}{2} \left\| \mathbf{X}_{\tilde{\mathcal{I}}_j^k, j} - \mathbf{U}_{\tilde{\mathcal{I}}_j^k}^{k+1} \mathbf{v}_j \right\|_2^2 \quad (53)$$

where $j = 1, \dots, n$, and its solution is:

$$\mathbf{v}_j^{k+1} = \left(\mathbf{U}_{\tilde{\mathcal{I}}_j^k}^{k+1} \right)^\dagger \mathbf{X}_{\tilde{\mathcal{I}}_j^k, j} \quad (54)$$

whose computational complexity is $\mathcal{O}(|\tilde{\mathcal{I}}_j^k|r^2)$. Given \mathbf{U}^{k+1} and \mathbf{V}^{k+1} , (50) can be rewritten as:

$$\mathbf{W}^{k+1} = \arg \min_{\mathbf{W}} \frac{1}{2} \left\| \sqrt{\mathbf{W}_\Omega} \circ \mathbf{R}_\Omega^{k+1} \right\|_F^2 + \varphi(\mathbf{W}_\Omega) \quad (55)$$

According to (13), its solution is:

$$\tilde{\Omega}^{k+1} = \mathbf{W}^{k+1} = \delta_M \left(\mathbf{R}_\Omega^{k+1} \right) \quad (56)$$

Algorithm 2 summarizes the matrix completion procedure based on HOMET. We take the same strategy as Algorithm 1 to terminate Algorithm 2 and determine its parameters including I_p , I_m , η and e . In addition, we use the solution to (3) as the initialization for (48) and (50). It is worth noting that the processes of finding the solutions to different \mathbf{u}_i in (52) for $i = 1, \dots, m$ and \mathbf{v}_j in (54) for $j = 1, \dots, n$, are independent of each other, (56) is entry-wise operator, and hence those steps can be calculated in parallel to reduce the computational time. Furthermore, we denote $\mathcal{C}(\mathbf{U}, \mathbf{V}, \mathbf{W}) = \frac{1}{2} \left\| \sqrt{\mathbf{W}_\Omega} \circ (\mathbf{X}_\Omega - (\mathbf{UV})_\Omega) \right\|_F^2 + \varphi(\mathbf{W}_\Omega)$, and the convergence analysis of Algorithm 2 is provided in Proposition 2.

Algorithm 2 HOMET

Input: Incomplete matrix \mathbf{X}_Ω , index set Ω , maximum allowable iteration numbers I_p and I_m , and tolerance parameter η

Initialize: Generate a standard Gaussian matrix $\mathbf{V}^0 \in \mathbb{R}^{r \times n}$, and find $\{\mathcal{J}_i\}_{i=1}^m$ and $\{\mathcal{I}_j\}_{j=1}^n$ according to Ω .

for $p = 1, 2, \dots, I_p$ **do**

 // Fix \mathbf{V}^{p-1} , optimize \mathbf{U}

 Find \mathbf{U}^p according to (37)

 // Fix \mathbf{U}^p , optimize \mathbf{V}

 Find \mathbf{V}^p according to (38)

end for

Set $\mathbf{U}^0 = \mathbf{U}^p$ and $\mathbf{V}^0 = \mathbf{V}^p$

for $k = 1, 2, \dots, I_m$ **do**

 Calculate e^k according to (35)

 // Fix \mathbf{U}^{k-1} and \mathbf{V}^{k-1} , optimize $\tilde{\Omega}$

 Find $\tilde{\Omega}^k$ according to (56)

 Update $\tilde{\mathcal{J}}_i^k$ and $\tilde{\mathcal{I}}_j^k$ according to $\tilde{\Omega}^k$

 // Fix \mathbf{V}^{k-1} and $\tilde{\Omega}^k$, optimize \mathbf{U}

 Find \mathbf{U}^k according to (52)

 // Fix \mathbf{U}^k and $\tilde{\Omega}^k$, optimize \mathbf{V}

 Find \mathbf{V}^k according to (54)

 Stop, if a termination condition is satisfied.

end for

Output: $\mathbf{M} = \mathbf{U}^k \mathbf{V}^k$.

Proposition 2: The sequence $\{\mathcal{C}(\mathbf{U}^k, \mathbf{V}^k, \mathbf{W}^k)\}$ generated by Algorithm 2 converges.

Proof: Similar to Proposition 1, since $\varphi(\mathbf{W})$ in $\mathcal{C}(\mathbf{U}, \mathbf{V}, \mathbf{W})$ is the only component related to e and is separable, we first obtain:

$$\frac{\partial \varphi(\mathbf{W}_{i,j}, e)}{\partial e} = \begin{cases} (1 - \mathbf{W}_{i,j})e, & 0 \leq \mathbf{W}_{i,j} < 1 \\ 0, & y \geq 1 \end{cases} \quad (57)$$

and then we have:

$$\frac{\partial \mathcal{L}}{\partial e} = \frac{\partial \sum \varphi(\mathbf{N}_{i,j}, e)}{\partial e} = \sum \frac{\partial \varphi(\mathbf{N}_{i,j}, e)}{\partial e} \geq 0 \quad (58)$$

Thus, for the non-increasing e^{k+1} in (35), $\mathcal{C}(\mathbf{U}, \mathbf{V}, \mathbf{W})$ is non-increasing.

Moreover, when e^k has been updated, since BCD is utilized to find the solution to (44), hence we have:

$$\begin{aligned} & \mathcal{C}(\mathbf{U}^{k+1}, \mathbf{V}^{k+1}, \mathbf{W}^{k+1}) \\ &= \min_{\mathbf{V}} \frac{1}{2} \left\| \sqrt{\mathbf{W}_{\Omega}^{k+1}} \circ (\mathbf{X}_{\Omega} - (\mathbf{U}^{k+1} \mathbf{V})_{\Omega}) \right\|_F^2 + \varphi(\mathbf{W}_{\Omega}^{k+1}) \\ &\leq \min_{\mathbf{U}} \frac{1}{2} \left\| \sqrt{\mathbf{W}_{\Omega}^{k+1}} \circ (\mathbf{X}_{\Omega} - (\mathbf{U} \mathbf{V}^k)_{\Omega}) \right\|_F^2 + \varphi(\mathbf{W}_{\Omega}^{k+1}) \\ &\leq \min_{\mathbf{W}} \frac{1}{2} \left\| \sqrt{\mathbf{W}_{\Omega}} \circ (\mathbf{X}_{\Omega} - (\mathbf{U}^k \mathbf{V}^k)_{\Omega}) \right\|_F^2 + \varphi(\mathbf{W}_{\Omega}) \\ &\leq \frac{1}{2} \left\| \sqrt{\mathbf{W}_{\Omega}^k} \circ (\mathbf{X}_{\Omega} - (\mathbf{U}^k \mathbf{V}^k)_{\Omega}) \right\|_F^2 + \varphi(\mathbf{W}_{\Omega}^k) \\ &= \mathcal{C}(\mathbf{U}^k, \mathbf{V}^k, \mathbf{W}^k) \end{aligned} \quad (59)$$

implying that $\mathcal{C}(\mathbf{U}, \mathbf{V}, \mathbf{W})$ is non-increasing with a lower bound, thus the sequence $\{\mathcal{C}(\mathbf{U}^k, \mathbf{V}^k, \mathbf{W}^k)\}$ produced via Algorithm 2 converges. ■

C. Computational Complexity

Although both Algorithms 1 and 2 introduce an auxiliary variable and utilize BCD to solve their corresponding optimization problems, they have different computational costs. For Algorithm 1, the complexity of least squares contains the cost of solving (21) by (30) and the cost of solving (22) by (32). The resultant complexity is $\sum_{i=1}^m \mathcal{O}(|\mathcal{I}_i| r^2) + \sum_{j=1}^n \mathcal{O}(|\mathcal{J}_j| r^2) = \mathcal{O}(2|\Omega| r^2)$ at each iteration. Besides, the cost of solving (23) is $\mathcal{O}(|\Omega|)$. Assuming that Algorithm 1 requires K_1 iterations to converge, its total computational complexity is $\mathcal{O}(K_1(2r^2 + 1)|\Omega|)$. Similarly, for Algorithm 2, the complexity of least squares is $\sum_{i=1}^m \mathcal{O}(|\tilde{\mathcal{I}}_i^k| r^2) + \sum_{j=1}^n \mathcal{O}(|\tilde{\mathcal{J}}_j^k| r^2) = \mathcal{O}(2|\tilde{\Omega}^k| r^2)$ in the k th iteration, which is not bigger than $\mathcal{O}(2|\Omega| r^2)$ due to $|\tilde{\Omega}^k| \leq |\Omega|$ at each iteration, while the cost of solving (47) is $\mathcal{O}(|\Omega|)$. Accordingly, the total computational complexity of Algorithm 2 is $\sum_{k=1}^{K_2} \mathcal{O}(2|\tilde{\Omega}^k| r^2 + |\Omega|)$, where K_2 is the iteration number for convergence. As analyzed in [45], the iteration number of the multiplicative form is less than that of the additive form, and hence the running time of the multiplicative form will be less than that of the additive form.

IV. RESULTS

In this section, compared with the state-of-the-art robust matrix completion methods, namely, ℓ_1 -ADMM [29], RMC-Huber [33], RMF-MM [34], VBMFL₁ [35] and HQ-PF [37], we perform experiments on synthetic data, real gray-scale and hyperspectral images, to show the effectiveness of the two proposed algorithms. For the parameters in the competing algorithms, the recommended setting is adopted, and if it is not available, we select the best parameters through experiments. In addition, we set $I_m = 50$ and $\eta = 1 \times 10^{-6}$ for the developed methods. All the experiments are conducted on a computer with 3.2 GHz CPU and 16 GB memory.

A. Results of Synthetic Random Data

The experimental strategy in [29], [32], and [37] is adopted. We first generate two random matrices $\mathbf{U} \in \mathbb{R}^{r \times r}$ and $\mathbf{V} \in \mathbb{R}^{r \times n}$, whose entries satisfy the standard Gaussian distribution, and then construct the synthetic matrix as $\mathbf{X} = \mathbf{U}\mathbf{V}$. For convenience, we set $m = n$, and $r = m/50$. Unless stated otherwise, $m = 400$, and the observation ratio is 50%. The incomplete matrix \mathbf{X}_{Ω} is perturbed via Gaussian mixture model (GMM). The probability density function of GMM is:

$$p_v(v) = \frac{1-c}{\sqrt{2\pi}\sigma_1} \exp\left(-\frac{v^2}{2\sigma_1^2}\right) + \frac{c}{\sqrt{2\pi}\sigma_2} \exp\left(-\frac{v^2}{2\sigma_2^2}\right) \quad (60)$$

where σ_1^2 and σ_2^2 are variances with $\sigma_1^2 \ll \sigma_2^2$. Besides, c controls the ratio of outliers, namely, a large value of c gives rise to higher ratio of outliers. In our experiments, to model gross errors, we set $\sigma_2^2 = 100\sigma_1^2$ and $c = 0.1$. Thus, the signal-to-noise ratio (SNR) is given by:

$$\text{SNR} = \frac{\|\mathbf{X}_{\Omega}\|_F^2}{|\Omega|((1-c)\sigma_1^2 + c\sigma_2^2)} \quad (61)$$

In addition, we set SNR = 10 dB unless stated otherwise. To evaluate the recovery performance, the root mean square error (RMSE) is employed, defined as:

$$\text{RMSE} = \frac{\|\mathbf{X} - \mathbf{M}\|_F}{\sqrt{m \times n}} \quad (62)$$

which is calculated based on 100 independent runs. It is easy to understand that a smaller value of RMSE implies a better recovery performance.

Fig. 2 plots RMSE versus iteration number by the seven methods, namely, HOAT, HOMT, VBMFL₁, RMC-Huber, ℓ_1 -ADMM, RMF-MM and HQ-PF, at SNR = 10 dB. It is observed that the recovery accuracy of our algorithms and HQ-PF is higher than that of the remaining schemes. It is because the penalty functions of the former are non-convex, while those of the latter are convex. Compared with ℓ_2 -norm, although ℓ_1 -norm and Huber function can reduce the influence of outliers, their ability to resist outliers is restricted especially for large outliers. Besides, the reason why HOAT and HOMT are superior to HQ-PF is that the latter changes the weights of 'normal' data, even though they all utilize the *dual* function and Frobenius norm to resist outliers and suppress Gaussian noise, respectively. Compared with the competing methods,

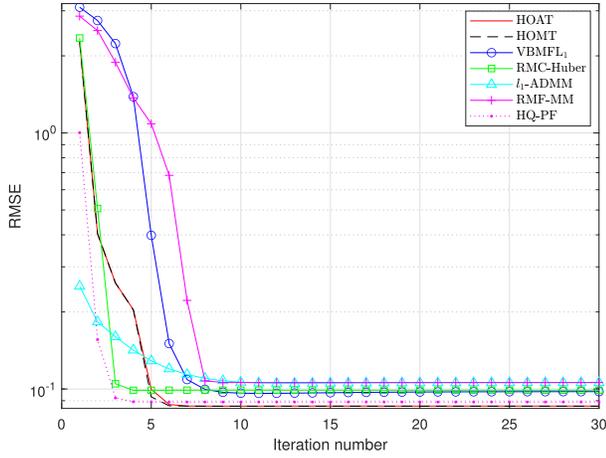


Fig. 2. RMSE versus iteration number.

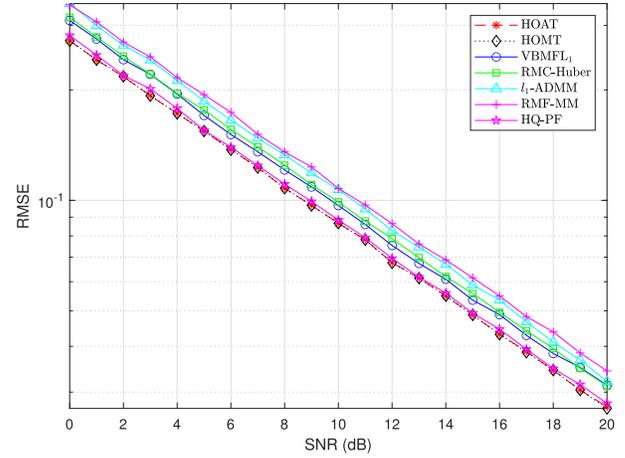


Fig. 4. RMSE versus SNR in GMM noise.

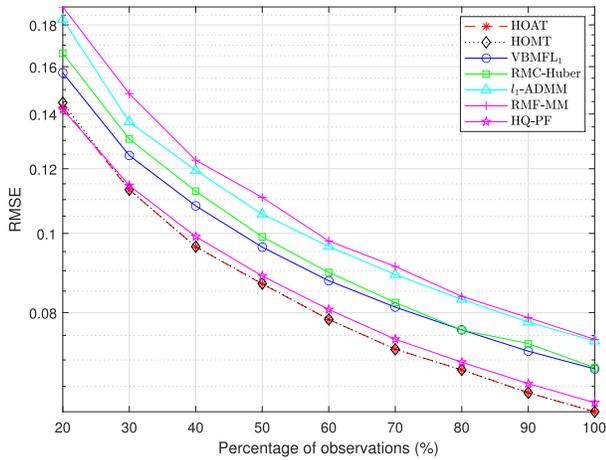
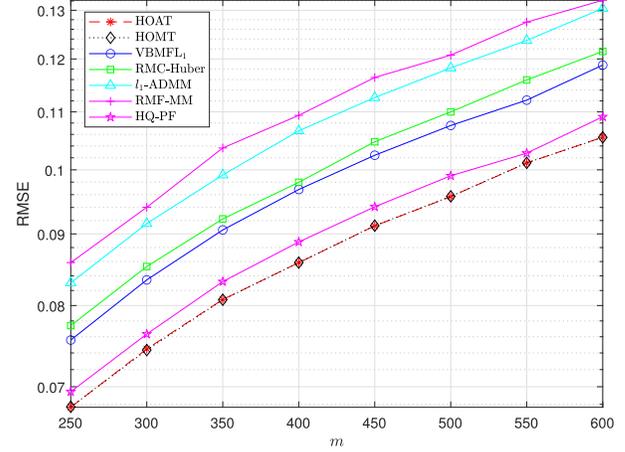


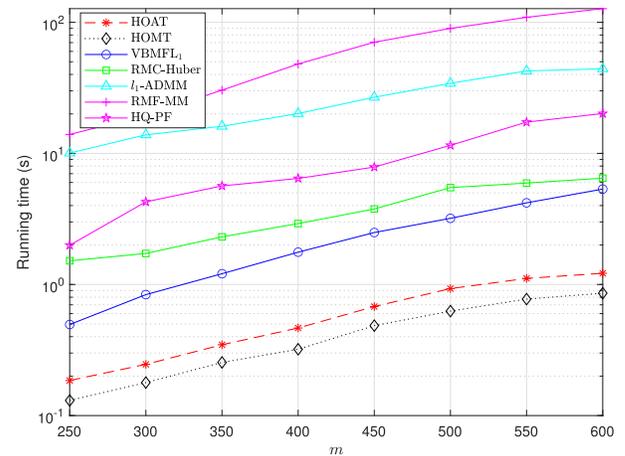
Fig. 3. RMSE versus percentage of observations.

Fig. 5. RMSE versus m .

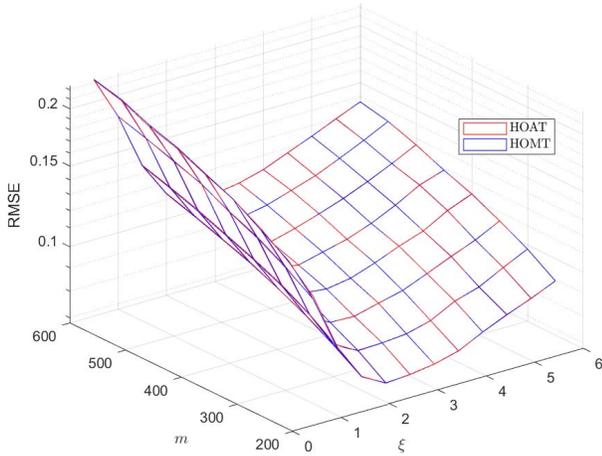
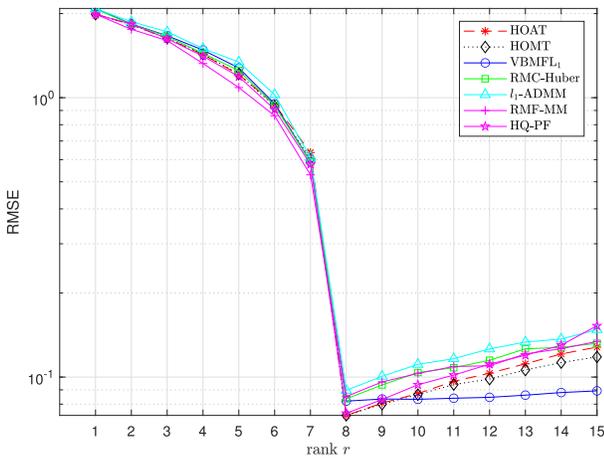
the proposed algorithms not only resist gross errors, but also do not change the weights of ‘normal’ data.

Fig. 3 depicts the RMSE versus percentage of observations. We easily see that the HOAT, HOMET and HQ-PF have better recovery performance for all observation percentages. It is because they adopt non-convex cost functions to resist gross errors, and researches in [33] and [40] show that non-convex penalty functions have a higher capability to resist big outliers than convex penalty functions. The impact of SNR on different algorithms is plotted in Fig. 4. We see that the RMSEs of all the algorithms decrease as SNR increases, and our algorithms perform the best.

Then, the relationship between the RMSE and matrix column m is investigated. As can be seen from Fig. 5, our algorithms have the best recovery performance for all m , although the RMSEs of all the methods increase with m . Compared with HOAT, HOMET and HQ-PF, large outliers still affect the recovery accuracy of VBMFL₁, RMC-Huber, ℓ_1 -ADMM and RMF-MM, since they utilize convex functions, i.e., ℓ_1 -norm and Huber function, as the penalties. In addition, we present the average runtime of different approaches in Fig. 6. It is easy to see that the computational cost of HOMET

Fig. 6. Running time versus m .

is the lowest among all the algorithms, while RMF-MM is the most time-consuming, which is approximately two orders of magnitude slower than HOAT and HOMET. Besides, the two proposed algorithms are at least an order of magnitude faster than HQ-PF though they all use non-convex cost functions and

Fig. 7. RMSE surface versus ζ and m .Fig. 8. RMSE versus rank r .

alternating minimization to find the solutions. In particular, the running time of HOAT is longer than that of HOMT, which is consistent with our analysis in Section III-C.

Fig. 7 depicts the selection of ζ in (35). We see that the two proposed approaches can work well in a wide range of ζ . In particular, when ζ is very small, our algorithms have larger RMSEs, because a smaller ζ will cause them wrongly take the ‘normal’ values as outliers, leading to a large degradation in recovery performance. While a much bigger value of ζ also gives rise to performance degradation since it may consider outliers as ‘normal’ values. When $2.5 \leq \zeta \leq 4$, HOAT and HOMT have a higher recovery accuracy, and a larger value of ζ may contain more useful details if all the outliers are excluded. Besides, it is observed that the selection of ζ is not sensitive to the matrix size m .

Finally, since our algorithms require knowing the rank information, it is necessary to provide a method to estimate its value. Similar to [29], cross-validation is used. In our experiments, the matrix column length is $m = 400$ and the true rank is set to 8. Furthermore, we randomly choose 95% observed entries as the training set and the remaining 5% are considered as the test data. Fig. 8 plots the RMSE versus rank r . We can see that all the methods can correctly estimate the matrix rank via cross-validation. It is worth noting that

when the estimated rank is less than the true rank, it will bring about a big recovery error.

B. Application on Image Inpainting

In some cases, the received images are incomplete because of the damage of photosensitive devices or the occlusion of other objects. Besides, the images may be contaminated by Gaussian noise and impulsive noise during the process of wireless transmission. In this section, we apply the robust matrix completion algorithms on image inpainting, and the images in [50] and [51] are tested. To model outliers, a mixture of zero-mean Gaussian noise and impulsive noise is adopted. The standard deviation of zero-mean Gaussian noise is set to be 0.02, and the impulsive noise is generated by the built-in command of ‘imnoise (\mathbf{I} , ‘salt & pepper’, ρ)’ in MATLAB, where \mathbf{I} is the incomplete matrix, and ρ is the normalized noise intensity. The relationship between ρ and SNR is $\rho = 1/\text{SNR}$ and $\text{SNR} = 10$ dB. Moreover, to measure the recovery performance of different algorithms, two evaluation indices, namely, peak SNR (PSNR) and structural similarity index measure (SSIM), are employed, and we directly use the built-in commands ‘psnr (recovered, original)’ and ‘ssim (recovered, original)’ in MATLAB for their calculation, respectively. In particular, all the algorithms use the same rank, and $\zeta = 4$ in the proposed algorithms. Furthermore, we investigate the impact of three kinds of masks [50], that is, random mask, text mask and block mask, on robust matrix completion, and all the methods are tested on 10 dB Gaussian noise, and the mixture noise for each mask.

We first address a relatively easy robust matrix completion problem, where the missing entries of an image are uniformly distributed. The image called Scenery in the top left of Fig. 9, is used, whose dimensions are 300×300 . Then, 20% of pixels of the image are randomly removed and two kinds of noise, i.e., Gaussian noise and the mixture noise, are added to the incomplete image in order to generate two different corrupted images. Table I tabulates the average recovery performance and running time for the two noise types. To further evaluate the recovery of different algorithms, their restored images in the mixture noise are shown in Fig. 9. It is observed that compared to VBMFL₁, RMC-Huber, ℓ_1 -ADMM and RMF-MM, the non-convex penalty algorithms, i.e., HOAT, HOMT and HQ-PF, achieve better reconstruction performance. Besides, although HQ-PF has a comparable recovery accuracy with HOAT and HOMT, it requires longer runtime. In particular, HOAT and HOMT outperform the competing algorithms in Gaussian noise because when there are no outliers, the penalty function used in the former is equal to ℓ_2 -norm for a bigger value of e , compared with the remaining algorithms.

The second is the text mask where the part covered by text in the image is considered as missing entries, which is more common than the first scenario, but its recovery is more difficult because the concealed regions may contain important details. As shown in Fig. 10, the Windows image is employed and corrupted by the text ‘Matrix’ ‘Completion’ as the incomplete image, whose dimensions are 349×366 . Again, the same intensity of Gaussian noise and mixture noise are

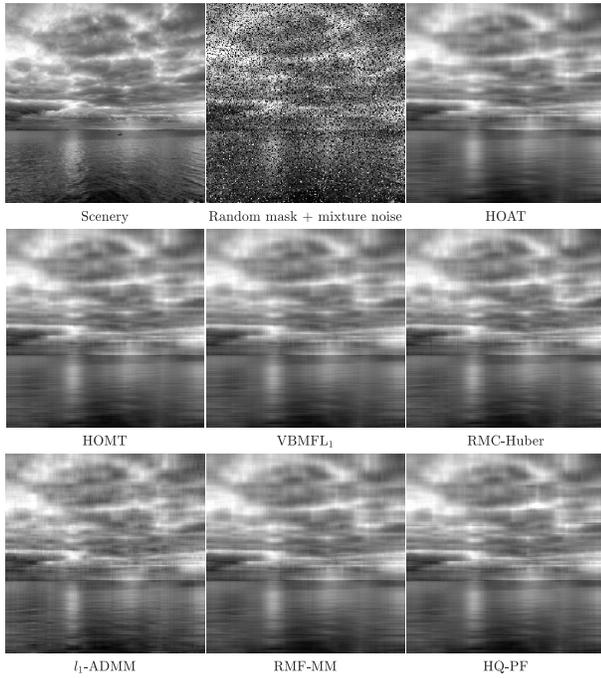


Fig. 9. Image recovery results with random mask by different algorithms.

TABLE I
PERFORMANCE COMPARISON FOR RANDOM MASK:
AVERAGE PSNR, SSIM AND RUNNING TIME

Method	Gaussian noise			Mixture noise		
	PSNR	SSIM	Runtime	PSNR	SSIM	Runtime
HOAT	23.250	0.3944	1.099	28.144	0.7504	3.196
HOMT	23.254	0.3949	0.938	28.036	0.7497	2.672
VBMFL ₁ [35]	23.157	0.3886	3.191	27.862	0.7390	12.466
RMC-Huber [33]	22.039	0.3264	9.926	27.817	0.7356	38.693
ℓ_1 -ADMM [29]	21.459	0.2959	21.799	26.720	0.6839	163.228
RMF-MM [34]	21.576	0.3006	94.867	27.703	0.7270	289.060
HQ-PF [37]	23.158	0.3321	29.999	28.128	0.7352	67.798

used to contaminate the incomplete image. Fig. 10 shows the recovered images by the seven approaches in the presence of the mixture noise. Table II shows their average PSNR, SSIM and running time for the two noise models. We easily observe that the HOAT and HOMT are superior to the remaining methods in the presence of Gaussian noise and mixture noise. Note that the competing algorithms cannot restore the light on the wall in Fig. 10. In addition, HOMT is comparable with HOAT in terms of PSNR and SSIM, but the former needs less complexity.

Block mask is also difficult to tackle since the resultant missing pattern is not random as well. The Stripe image, whose dimensions are 272×271 , is covered by different shaped blocks and corrupted via the Gaussian noise or mixture noise. Fig. 11 depicts the recovered images in the presence of the mixture noise, and Table III shows their average recovery performance and runtime. Again, compared with the VBMFL₁, RMC-Huber, ℓ_1 -ADMM, RMF-MM and HQ-PF,

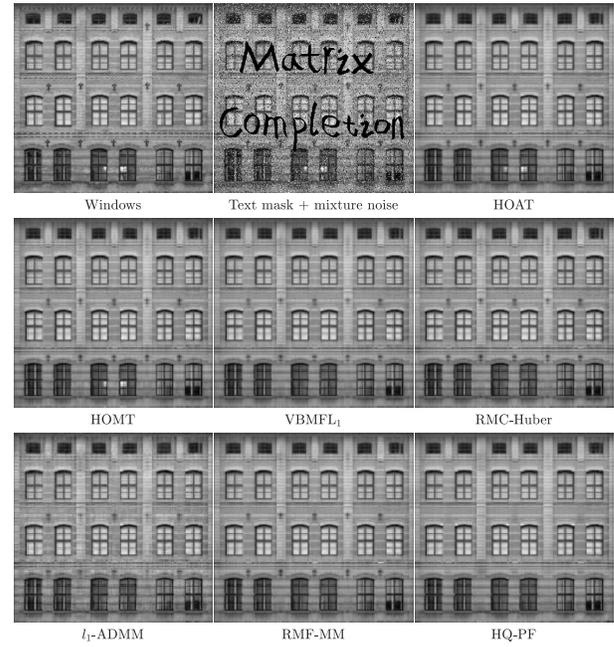


Fig. 10. Image recovery results with text mask by different algorithms.

TABLE II
PERFORMANCE COMPARISON FOR TEXT MASK:
AVERAGE PSNR, SSIM AND RUNNING TIME

Method	Gaussian noise			Mixture noise		
	PSNR	SSIM	Runtime	PSNR	SSIM	Runtime
HOAT	24.202	0.6520	4.6316	28.116	0.8523	1.129
HOMT	24.200	0.6517	4.139	28.083	0.8522	0.729
VBMFL ₁ [35]	23.920	0.6387	6.663	27.747	0.8477	5.889
RMC-Huber [33]	23.064	0.5933	16.632	27.804	0.8469	19.456
ℓ_1 -ADMM [29]	22.323	0.5509	44.026	26.597	0.8073	27.982
RMF-MM [34]	22.532	0.5658	389.137	27.670	0.8435	148.132
HQ-PF [37]	24.142	0.6024	127.571	28.045	0.8449	40.394

TABLE III
PERFORMANCE COMPARISON FOR BLOCK MASK:
AVERAGE PSNR, SSIM AND RUNNING TIME

Method	Gaussian noise			Mixture noise		
	PSNR	SSIM	Runtime	PSNR	SSIM	Runtime
HOAT	25.043	0.7442	1.032	29.150	0.9261	4.048
HOMT	25.040	0.7441	0.948	29.175	0.9270	3.654
VBMFL ₁ [35]	24.429	0.7101	1.993	28.980	0.9253	4.116
RMC-Huber [33]	24.025	0.6958	14.090	28.859	0.9246	32.518
ℓ_1 -ADMM [29]	23.206	0.6521	45.016	27.380	0.8906	186.702
RMF-MM [34]	23.366	0.6637	77.555	28.837	0.9221	175.011
HQ-PF [37]	25.018	0.7060	37.762	29.070	0.9090	57.555

our algorithms are more robust to Gaussian noise and mixture noise, producing the best recovery performance in terms of PSNR as well as SSIM, and requiring the least computational time.

C. Application on Hyperspectral Imaging

Hyperspectral imaging (HSI) plays an important role in many applications including earth climate, agriculture and

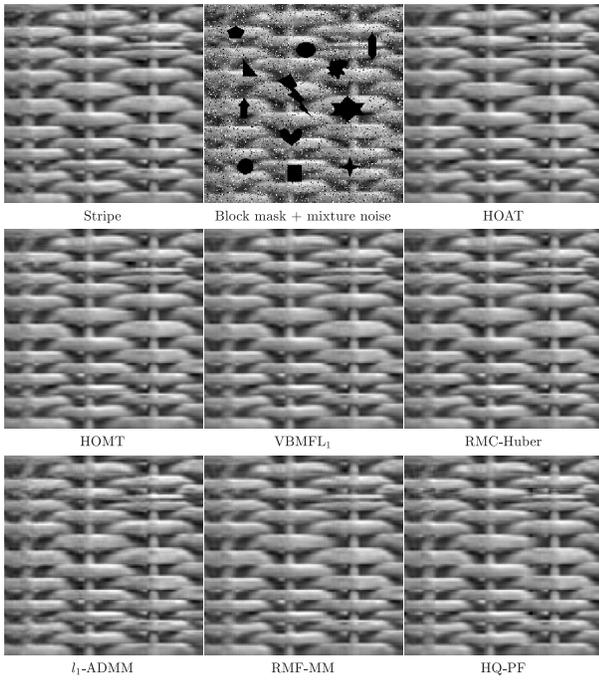


Fig. 11. Image recovery results with block mask by different algorithms.

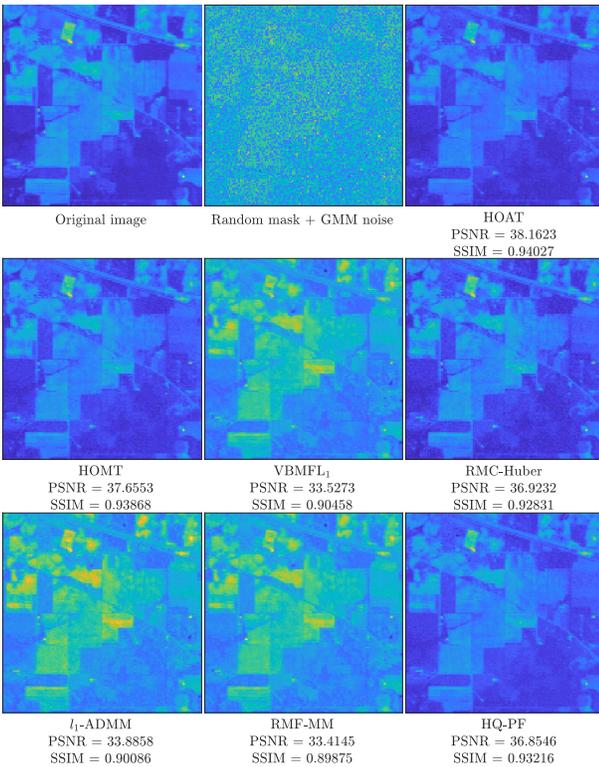


Fig. 12. HSI restoration results for the 10th band.

urban planning [4], [5], [52]. However, it may suffer from annoying performance degradation during the acquisition process such as missing data and being contaminated by Gaussian noise and impulsive noise, due to photon effects and calibration error. Hence, there is a need to improve the

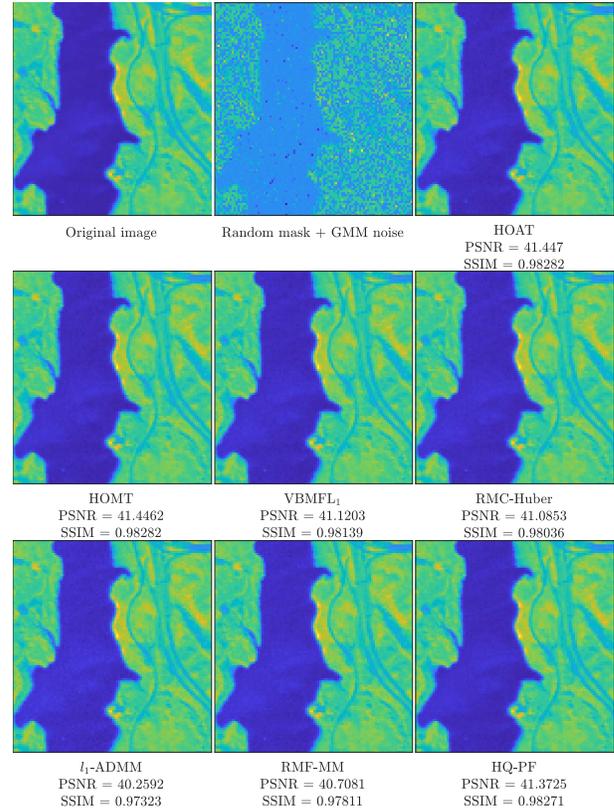


Fig. 13. HSI restoration results for the 80th band.

HSI quality prior to its subsequent processing. In this section, Indian Pines [52] and Jasper Ridge¹ datasets are used, whose dimensions are 145×145 pixels per band with 200 bands and 100×100 pixels per band with 198 bands, respectively. Then, we reconstruct a matrix $\mathbf{X} \in \mathbb{R}^{21025 \times 200} / \mathbb{R}^{10000 \times 198}$, whose columns comprise vectorized bands of the Indian Pines/Jasper Ridge data. Furthermore, 40% of pixels in \mathbf{X} are randomly missing, 10 dB impulsive noise produced by GMM is added to the incomplete data, and $\zeta = 3$ is chosen in the proposed algorithms. Fig. 12 displays the restoration results of different methods for the Indian Pines data. It is seen that compared with VBMFL₁, ℓ_1 -ADMM, and RMF-MM, the four methods, namely, HOAT, HOMT, RMC-Huber and HQ-PF, achieve better recovery results and our algorithms outperform the RMC-Huber and HQ-PF in terms of PSNR and SSIM. Moreover, the recovered images for Jasper Ridge data are shown in Fig. 13. Again, the HOAT and HOMT are superior to the competing algorithms due to higher PSNR and SSIM.

V. CONCLUSION

In this paper, the truncated-quadratic function, which is non-convex and non-smooth, is employed to resist outliers and HO theory is adopted to convert the loss function into two forms, that is, additive and multiplicative models, via introducing an auxiliary variable, giving rise to two easy-to-solve optimization problems. Accordingly, we propose two robust matrix completion algorithms, called HOAT and HOMT. In particular,

¹<http://lesun.weebly.com/hyperspectral-data-set.html>

the elements contaminated by outliers in HOMT are considered as the missing entries, which can reduce its computational cost. Besides, BCD is utilized to find their solutions, and we show that the computational complexity of the additive form is higher than that of the multiplicative form, implying that the latter is preferred, if possible. Furthermore, extensive experimental results based on synthetic and real-world data demonstrate the superiority of the HOAT and HOMT over the competing algorithms in terms of RMSE or PSNR/SSIM, and running time.

APPENDIX A
DERIVATION OF MINIMIZER FUNCTION
OF ADDITIVE FORM

Based on (10), we obtain the *dual* function of (15) [43]:

$$\varphi(y) = \sup_x -\frac{(x-y)^2}{2} + \phi(x) \quad (63)$$

which is determined by $\phi(x)$. Substituting (15) in (63) yields:

$$\varphi(y) = \begin{cases} \sup_x -\frac{(x-y)^2}{2} + \frac{x^2}{2}, & |x| < e \\ \sup_x -\frac{(x-y)^2}{2} + \frac{e^2}{2}, & |x| \geq e \end{cases} \quad (64)$$

and hence,

$$\varphi(y) = \begin{cases} -\frac{(|y|-e)^2}{2} + \frac{e^2}{2}, & |y| < e \\ \frac{e^2}{2}, & |y| \geq e \end{cases} \quad (65)$$

Then, plugging (65) into (10) leads to:

$$\begin{aligned} & \inf_y \frac{(x-y)^2}{2} + \varphi(y) \\ &= \begin{cases} \inf_y \frac{2(e-x)y + x^2}{2}, & 0 \leq y < e \\ \inf_y \frac{-2(e+x)y + x^2}{2}, & -e < y < 0 \\ \inf_y \frac{(x-y)^2}{2} + \frac{e^2}{2}, & |y| \geq e \end{cases} \\ &= \begin{cases} \frac{x^2}{2}, & |x| < e \\ \frac{e^2}{2}, & |x| \geq e \end{cases} = \phi(x) \end{aligned} \quad (66)$$

We denote the minimizer function associated with (66) as $\delta_A(x)$:

$$\begin{aligned} \delta_A(x) &:= \arg \inf_y \frac{(x-y)^2}{2} + \varphi(y) \\ &= \begin{cases} \arg \inf_y \frac{2(e-x)y + x^2}{2}, & 0 \leq y < e \\ \arg \inf_y \frac{-2(e+x)y + x^2}{2}, & -e < y < 0 \\ \arg \inf_y \frac{(x-y)^2}{2} + \frac{e^2}{2}, & |y| \geq e \end{cases} \\ &= \begin{cases} 0, & |x| < e \\ x, & |x| \geq e \end{cases} \end{aligned} \quad (67)$$

APPENDIX B
DERIVATION OF MINIMIZER FUNCTION OF
MULTIPLICATIVE FORM

First, the *dual* function of (15) for multiplicative form is [44]:

$$\varphi(y) = \sup_x -\frac{yx^2}{2} + \phi(x) \quad (68)$$

Then, substituting (15) into (68), we have:

$$\varphi(y) = \begin{cases} \sup_x -\frac{yx^2}{2} + \frac{x^2}{2}, & |x| < e \\ \sup_x -\frac{yx^2}{2} + \frac{e^2}{2}, & |x| \geq e \end{cases} \quad (69)$$

and it is easy to obtain:

$$\varphi(y) = \begin{cases} \frac{1-y}{2}e^2, & 0 \leq y < 1 \\ 0, & y \geq 1 \end{cases} \quad (70)$$

Hence,

$$\begin{aligned} & \inf_y \frac{yx^2}{2} + \varphi(y) \\ &= \begin{cases} \inf_y \frac{yx^2}{2} + \frac{1-y}{2}e^2, & 0 \leq y < 1 \\ \inf_y \frac{yx^2}{2}, & y \geq 1 \end{cases} \\ &= \begin{cases} \frac{x^2}{2}, & |x| < e \\ \frac{e^2}{2}, & |x| \geq e \end{cases} = \phi(x) \end{aligned} \quad (71)$$

The minimizer function $\delta_M(x)$ related to (71) is:

$$\begin{aligned} \delta_M(x) &:= \begin{cases} \arg \inf_y \frac{yx^2}{2} + \frac{1-y}{2}e^2, & 0 \leq y < 1 \\ \arg \inf_y \frac{yx^2}{2}, & y \geq 1 \end{cases} \\ &= \begin{cases} 1, & |x| < e \\ 0, & |x| \geq e \end{cases} \end{aligned} \quad (72)$$

REFERENCES

- [1] M. A. Davenport and J. Romberg, "An overview of low-rank matrix recovery from incomplete observations," *IEEE J. Sel. Topics Signal Process.*, vol. 10, no. 4, pp. 608–622, Jun. 2016.
- [2] Y. Chen and Y. Chi, "Harnessing structures in big data via guaranteed low-rank matrix estimation: Recent theory and fast algorithms via convex and nonconvex optimization," *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 14–31, Jul. 2018.
- [3] Y. Chi, Y. M. Lu, and Y. Chen, "Nonconvex optimization meets low-rank matrix factorization: An overview," *IEEE Trans. Signal Process.*, vol. 67, no. 20, pp. 5239–5269, Oct. 2019.
- [4] W. He, H. Zhang, L. Zhang, and H. Shen, "Hyperspectral image denoising via noise-adjusted iterative low-rank matrix approximation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 3050–3061, Jun. 2015.
- [5] H. Zhang, W. He, L. Zhang, H. Shen, and Q. Yuan, "Hyperspectral image restoration using low-rank matrix recovery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4729–4743, Aug. 2014.

- [6] A. Ramlatchan, M. Yang, Q. Liu, M. Li, J. Wang, and Y. Li, "A survey of matrix completion methods for recommendation systems," *Big Data Mining Anal.*, vol. 1, no. 4, pp. 308–323, 2018.
- [7] Y. Zhou, D. Wilkinson, R. Schreiber, and R. Pan, "Large-scale parallel collaborative filtering for the Netflix prize," in *Proc. 4th Int. Conf. Algorithmic Appl. Manag.*, Shanghai, China, 2008, pp. 337–348.
- [8] Q. Liu, X. Li, H. Cao, and Y. Wu, "From simulated to visual data: A robust low-rank tensor completion approach using ℓ_p -regression for outlier resistance," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3462–3474, Jun. 2022.
- [9] H. Xu, J. Zheng, X. Yao, Y. Feng, and S. Chen, "Fast tensor nuclear norm for structured low-rank visual inpainting," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 538–552, Feb. 2022.
- [10] A. Argyriou, T. Evgeniou, and M. Pontil, "Convex multi-task feature learning," *Mach. Learn.*, vol. 73, no. 3, pp. 243–272, 2008.
- [11] Z. Liu, A. Hansson, and L. Vandenberghe, "Nuclear norm system identification with missing inputs and outputs," *Syst. Control Lett.*, vol. 62, no. 8, pp. 605–612, Aug. 2013.
- [12] F. Taherkhani, H. Kazemi, and N. M. Nasrabadi, "Matrix completion for graph-based deep semi-supervised learning," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 5058–5065.
- [13] S. Ahmadi and M. Rezaghi, "Generalized low-rank approximation of matrices based on multiple transformation pairs," *Pattern Recognit.*, vol. 108, Dec. 2020, Art. no. 107545.
- [14] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Found. Comput. Math.*, vol. 9, no. 6, p. 717, Apr. 2009.
- [15] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.
- [16] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Jan. 2010.
- [17] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and Bregman iterative methods for matrix rank minimization," *Math. Program.*, vol. 128, nos. 1–2, pp. 321–353, 2011.
- [18] K.-C. Toh and S. Yun, "An accelerated proximal gradient algorithm for nuclear norm regularized least squares problems," *Pacific J. Optim.*, vol. 6, pp. 615–640, Jan. 2010.
- [19] R. Meka, P. Jain, and I. S. Dhillon, "Guaranteed rank minimization via singular value projection," 2009, *arXiv:0909.5457*.
- [20] J. Tanner and K. Wei, "Normalized iterative hard thresholding for matrix completion," *SIAM J. Sci. Comput.*, vol. 35, no. 5, pp. 104–125, Jan. 2013.
- [21] X. Jiang, Z. Zhong, X. Liu, and H. C. So, "Robust matrix completion via alternating projection," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 579–583, May 2017.
- [22] Z. Wen, W. Yin, and Y. Zhang, "Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm," *Math. Programm. Comput.*, vol. 4, no. 4, pp. 333–361, Dec. 2012.
- [23] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proc. 45th Annu. ACM Symp. Symp. Theory Comput. (STOC)*, 2013, pp. 665–674.
- [24] Z. Zhu, Q. Li, G. Tang, and M. B. Wakin, "Global optimality in low-rank matrix optimization," *IEEE Trans. Signal Process.*, vol. 66, no. 13, pp. 3614–3628, Jul. 2018.
- [25] Z. Zhu, Q. Li, G. Tang, and M. B. Wakin, "The global optimization geometry of low-rank matrix optimization," *IEEE Trans. Inf. Theory*, vol. 67, no. 2, pp. 1308–1331, Feb. 2021.
- [26] A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma, "Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts," *IEEE Signal Process. Mag.*, vol. 29, no. 4, pp. 61–80, Jul. 2012.
- [27] A. M. Zoubir, V. Koivunen, E. Ollila, and M. Muma, *Robust Statistics for Signal Processing*. Cambridge, MA, USA: Cambridge Univ. Press, 2018.
- [28] V. S. Bhadouria and D. Ghoshal, "A study on genetic expression programming-based approach for impulse noise reduction in images," *Signal, Image Video Process.*, vol. 10, no. 3, pp. 575–584, Mar. 2016.
- [29] W.-J. Zeng and H. C. So, "Outlier-robust matrix completion via ℓ_p -minimization," *IEEE Trans. Signal Process.*, vol. 66, no. 5, pp. 1125–1140, Mar. 2018.
- [30] F. Wen, R. Ying, P. Liu, and R. C. Qiu, "Robust PCA using generalized nonconvex regularization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1497–1510, Jun. 2020.
- [31] X. Zhang, J. Zheng, D. Wang, and L. Zhao, "Exemplar-based denoising: A unified low-rank recovery framework," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2538–2549, Aug. 2020.
- [32] L. Zhao, P. Babu, and D. P. Palomar, "Efficient algorithms on robust low-rank matrix completion against outliers," *IEEE Trans. Signal Process.*, vol. 64, no. 18, pp. 4767–4780, Sep. 2016.
- [33] M. Muma, W.-J. Zeng, and A. M. Zoubir, "Robust M-estimation based matrix completion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 5476–5480.
- [34] Z. Lin, C. Xu, and H. Zha, "Robust matrix factorization by majorization minimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 208–220, Jan. 2018.
- [35] Q. Zhao, D. Meng, Z. W. X. Zuo, and Y. Yan, " L_1 -norm low-rank matrix factorization by variational Bayesian method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 4, pp. 825–839, Apr. 2015.
- [36] T. T. Mai and P. Alquier, "A Bayesian approach for noisy matrix completion: Optimal rate under general sampling distribution," *Electron. J. Statist.*, vol. 9, no. 1, pp. 823–841, Jan. 2015.
- [37] Y. He, F. Wang, Y. Li, J. Qin, and B. Chen, "Robust matrix completion via maximum correntropy criterion and half-quadratic optimization," *IEEE Trans. Signal Process.*, vol. 68, pp. 181–195, 2020.
- [38] M. Nikolova, "Thresholding implied by truncated quadratic regularization," *IEEE Trans. Signal Process.*, vol. 48, no. 12, pp. 3437–3450, Dec. 2000.
- [39] M. Nikolova, "Analysis of the recovery of edges in images and signals by minimizing nonconvex regularized least-squares," *SIAM Multiscale Modeling Simulation*, vol. 4, no. 3, pp. 960–991, 2005.
- [40] R. He, T. Tan, and L. Wang, "Robust recovery of corrupted low-rank matrix by implicit regularizers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 4, pp. 770–783, Apr. 2014.
- [41] F. D. Mandanas and C. L. Kotropoulos, "Robust multidimensional scaling using a maximum correntropy criterion," *IEEE Trans. Signal Process.*, vol. 65, no. 4, pp. 919–932, Feb. 2017.
- [42] Y. Sun, P. Babu, and D. P. Palomar, "Majorization-minimization algorithms in signal processing, communications, and machine learning," *IEEE Trans. Signal Process.*, vol. 65, no. 3, pp. 794–816, Feb. 2017.
- [43] D. Geman and C. Yang, "Nonlinear image recovery with half-quadratic regularization," *IEEE Trans. Signal Process.*, vol. 4, no. 7, pp. 932–946, Jul. 1995.
- [44] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Process.*, vol. 6, no. 2, pp. 298–311, Feb. 1997.
- [45] M. Nikolova and M. K. Ng, "Analysis of half-quadratic minimization methods for signal and image recovery," *SIAM J. Sci. Comput.*, vol. 27, no. 3, pp. 937–966, 2005.
- [46] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.
- [47] Z. Q. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," *J. Optim. Theory Appl.*, vol. 72, no. 1, pp. 7–35, 1992.
- [48] J. Liu, P. C. Cosman, and B. D. Rao, "Robust linear regression via ℓ_0 regularization," *IEEE Trans. Signal Process.*, vol. 66, no. 3, pp. 698–713, Feb. 2018.
- [49] R. A. Maronna, D. R. Martin, and V. J. Yohai, *Robust Statistics: Theory and Methods*. Hoboken, NJ, USA: Wiley, 2006.
- [50] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, "Fast and accurate matrix completion via truncated nuclear norm regularization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2117–2130, Sep. 2013.
- [51] L. Chen, X. Jiang, X. Liu, and Z. Zhou, "Robust low-rank tensor recovery via nonconvex singular value minimization," *IEEE Trans. Image Process.*, vol. 29, pp. 9044–9059, 2020.
- [52] W. He, H. Zhang, L. Zhang, and H. Shen, "Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 178–188, Jan. 2016.



Zhi-Yong Wang was born in Henan, China. He received the B.S. degree in mechanical engineering from Zhengzhou University, Zhengzhou, China, in 2017, and the M.S. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2020. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, City University of Hong Kong under the supervision of Prof. Hing Cheung So.

His research interests include sparse recovery, robust signal processing and low-rank approximation, and their applications in image inpainting and video restoration.



Xiao Peng Li received the B.Eng. degree in electronic science and technology from Yanshan University, Qinhuangdao, China, in 2015, and the M.Sc. degree in electronic information engineering from the City University of Hong Kong, Hong Kong, China, in 2018, where he is pursuing the Ph.D. degree in electrical engineering under the supervision of Prof. Hing Cheung So.

From 2018 to 2019, he was a research assistant with Department of Information Engineering, Shenzhen University, Shenzhen, China. His research interests include deep neural networks, sparse recovery, matrix processing, tensor processing and their applications on image recovery, video restoration, hyperspectral unmixing, and stock market analysis.



Hing Cheung So (Fellow, IEEE) was born in Hong Kong. He received the B.Eng. degree from the City University of Hong Kong and the Ph.D. degree from The Chinese University of Hong Kong, in 1990 and 1995, respectively, all in electronic engineering.

From 1990 to 1991, he was an Electronic Engineer at the Research and Development Division, Everex Systems Engineering Ltd., Hong Kong. From 1995 to 1996, he was a Post-Doctoral Fellow at The Chinese University of Hong Kong. From 1996 to 1999, he was a Research Assistant

Professor at the Department of Electronic Engineering, City University of Hong Kong, where he is currently a Professor. His research interests include detection and estimation, fast and adaptive algorithms, multidimensional harmonic retrieval, robust signal processing, source localization, and sparse approximation. He was an Elected Member in Signal Processing Theory and Methods Technical Committee from 2011 to 2016 of the IEEE Signal Processing Society where he was the Chair in the awards subcommittee from 2015 to 2016. He was on the Editorial Boards of *IEEE Signal Processing Magazine* from 2014 to 2017 and *IEEE TRANSACTIONS ON SIGNAL PROCESSING* from 2010 to 2014. He has been on the Editorial Boards of *Signal Processing*, since 2010, and *Digital Signal Processing*, since 2011. He was the Lead Guest Editor for Special Issue on Advances in Time/Frequency Modulated Array Signal Processing of *IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING*, in 2017.