Journal of Industrial and Management Optimization

https://www.aimspress.com/journal/jimo

JIMO, 22(12): 6374–6398.
DOI:10.3934/jimo.2026214
Received: 31 July 2026
Revised: 28 October 2026
Accepted: 03 November 2026
Published: 10 November 2026

Research article

An alternating proximal gradient algorithm with partial SVD for low-rank and mix-sparse matrix decomposition

Xian Zhang[1], **Xin Wang**[2,*] **and Lihua Wang**[3]

[1] School of Science, Guiyang University, Guiyang, 550025, China
[2] School of Mathematics and Statistics, Beijing Jiaotong University, Beijing, 100044, China
[3] School of Mathematics and Statistics, Guizhou University, Guiyang, 550025, China

* **Correspondence:** Email: xinwang3@bjtu.edu.cn.

Abstract: Low-rank and mix-sparse matrix decomposition has wide applications in computer vision, compressed sensing, and machine learning. In this paper, we study a nonconvex low-rank and mix sparse optimization problem, where the low-rank part is induced by the $\ell_p(0 < p < 1)$ quasi-norm and the mix-sparse part is induced by a penalty composed of a combination of ℓ_0 norm and $\ell_{2,0}$ norm. To solve this nonconvex optimization problem, we propose an alternating proximal gradient algorithm with partial singular value decomposition (APGP). The significant advantage of the APGP is that the partial-SVD strategy reduces the computational cost of low-rank updates, while the mix-sparse update admits a closed-form solution with low storage requirements. Furthermore, we establish the global convergence of the whole iterative sequence generated by the proposed APGP algorithm. Numerical experiments on synthetic data and background subtraction demonstrate that APGP performs competitively with representative algorithms.

Keywords: Low-rank and mix-sparse decomposition; alternating proximal gradient algorithm; partial SVD; background subtraction
Mathematics Subject Classification: 90C06, 90C26, 90C90

1. Introduction

In this paper, we consider the following general low-rank and mix-sparse decomposition problem:

$$\min_{L,S} \ \frac{\varrho}{2}\|D - L - S\|_F^2 + \mathcal{F}(L) + \mathcal{H}(S) \tag{1.1}$$

where $D \in \mathbb{R}^{m\times n}$ is the observed corrupted data matrix, $L \in \mathbb{R}^{m\times n}$ denotes the low-rank component, $S \in \mathbb{R}^{m\times n}$ denotes the perturbation component with a mix-sparse structure, and $\varrho > 0$ is a fidelity

parameter. The functions $\mathcal{F}$ and $\mathcal{H}$ are proper and lower semicontinuous regularization that promote the desired structures of L and S, respectively. Specifically, we use the ℓ_p quasi-norm [1,2]

$$\mathcal{F}(L) = \lambda_1 \|L\|_p^p = \lambda_1 \sum_{i=1}^{\min\{m,n\}} \sigma_i^p(L), \quad 0 < p < 1$$

to approximate the rank of L. To model mixed sparse corruptions, we define $\mathcal{H}(S) = \lambda_2\|S\|_0 + \lambda_3\|S\|_{2,0}$, where $\|S\|_0$ counts the number of nonzero entries of S and $\|S\|_{2,0} = \sum_j^n \mathbf{1}(\|S_{\cdot j}\|_2 \neq 0)$ counts its nonzero groups [3,4]. Here λ_1, λ_2, $\lambda_3 > 0$ are regularization parameters. Problem (1.1) has wide applications in many areas involving computer vision, compressed sensing, statistics, image processing, and machine learning, for example, video surveillance [5], remote sensing [6,7], image denoising and inpainting [8–10], sensor network localization [11,12], and machine learning [13].

Several classical low-rank and sparse matrix decomposition models can be derived from (1.1). As $p \to 0$ and $\lambda_3 = 0$, the ℓ_p quasi-norm approaches the rank function, and the mix-sparse regularization reduces to the element-wise sparsity term. The resulting model is a penalized form of robust principal component analysis (RPCA):

$$\min_{L,S} \ \frac{\varrho}{2}\|D - L - S\|_F^2 + \lambda_1 \mathrm{rank}(L) + \lambda_2\|S\|_0, \tag{1.2}$$

where $\mathrm{rank}(L) = \|\sigma(L)\|_0$, where $\sigma(L)$ denotes the singular values vector of L. Since both the rank function and the ℓ_0 norm are nonconvex and difficult to handle directly, their convex relaxations are often considered by replacing them with the nuclear norm and the ℓ_1 norm, respectively. The resulting convex models can be efficiently solved by the augmented Lagrangian multiplier (ALM) method [14], the inexact augmented Lagrangian multiplier (IALM) method [15], the alternating direction method of multipliers (ADMM) [16], and the accelerated proximal gradient (APG) method [17–19]. Related convergence analyses and exact recovery conditions have also been extensively studied [14,20].

However, convex relaxations based on the nuclear norm and the ℓ_1 norm often suffer from over-penalization and high computational complexity. Consequently, many studies have adopted nonconvex surrogate functions to approximate the rank function [21–23] and the ℓ_0 norm [16,24,25], respectively. For the nonconvex relaxation of the rank function, Kang and Peng [26] proposed a nonconvex rank approximation method, which provides a tighter approximation to the rank function than the nuclear norm. Xie and Gu [23] developed the weighted Schatten-p norm minimization (WSNM) method, which assigns adaptive weights to different singular values and extends conventional nuclear norm minimization to a weighted Schatten-p norm minimization framework. Fang and Xu [27] proposed an efficient robust principal component analysis (eRPCA) algorithm via block Krylov iteration and CUR decomposition. For the nonconvex relaxation of the ℓ_0 norm, existing studies mainly include non-Lipschitz ℓ_p regularized RPCA [24] and Lipschitz nonconvex regularized RPCA [28]. Yang and Pong [16] used the ℓ_p norm as a nonconvex relaxation of the ℓ_0 norm and solved it by ADMM. In addition, adaptive weighted RPCA and adaptive double-weighted RPCA methods were proposed in [25,29] to improve low-rank matrix recovery by adaptively weighting singular values.

On the other hand, as $p \to 0$, $\lambda_2 = 0$, problem (1.1) reduces to a low-rank and group-sparse decomposition problem:

$$\min_{L,S} \ \frac{\varrho}{2}\|D - L - S\|_F^2 + \lambda_1 \mathrm{rank}(L) + \lambda_3\|S\|_{2,0}. \tag{1.3}$$

Existing studies mainly focus on its convex relaxation, especially the $\ell_{2,1}$ norm, including theoretical analysis and numerical algorithms [30–33]. In terms of algorithms, the approximated robust principal component analysis (ARPCA) method has been applied to background separation tasks [34]; the randomly permuted alternating direction method of multipliers (RP-ADMM) was proposed in [35] to enhance the robustness of the model under mix-noise scenarios. These methods improve, to some extent, the ability of low-rank group-sparse models to characterize structured sparse perturbations. However, the rank + $\ell_{2,0}$ model mainly describes group-sparse perturbations and has limited ability to characterize isolated sparse noise, whereas the rank + ℓ_0 model can describe element-wise sparse noise but is insufficient for characterizing structured sparse anomalies.

The proximal mapping of the convex sparse-group-Lasso regularization can be expressed as the composition of the proximal mappings of the Lasso regularization and group Lasso regularization [36]. Recently, proximal point methods, proximal gradient algorithms, and related composite optimization techniques have been further developed for solving large-scale nonsmooth and nonconvex optimization problems, providing effective tools for handling structured regularization models [18, 19, 37]. However, the corresponding nonconvex $\ell_0 + \ell_{2,0}$ regularization is discontinuous and requires a different analysis. This motivates us to derive a closed-form solution and to develop an efficient alternating algorithm for model (1.1).

The main contributions of this paper are summarized as follows:

1 We propose a new model (1.1) that simultaneously characterizes element-wise sparsity and group sparsity. For the mix-sparse regularization term $\lambda_2\|\cdot\|_0 + \lambda_3\|\cdot\|_{2,0}$, we derive the analytical solution of the corresponding proximal problem, which enables an efficient closed-form update for the sparse component.
2 To solve problem (1.1), we propose an alternating proximal gradient algorithm with partial singular value decomposition (SVD), abbreviated as APGP. By exploiting the specific structures of the low-rank and mix-sparse regularization terms, APGP incorporates a partial-SVD strategy to reduce the computational cost of low-rank updates, while the mix-sparse update admits a closed-form solution with low storage requirements. Furthermore, the theoretical results indicate that the whole sequence generated by APGP globally converges to a critical point of problem (1.1).
3 Numerical experiments on synthetic recovery and background subtraction demonstrate that APGP performs competitively against representative algorithms in terms of recovery accuracy, computational efficiency, and robustness.

The outline of this paper is as follows. Section 2 presents the notations and preliminaries, including the analytical solution for the proximal operator associated with $\lambda_2\|\cdot\|_0 + \lambda_3\|\cdot\|_{2,0}$. Section 3 develops the partial-SVD-based APGP and analyzes its convergence properties. Section 4 reports numerical experiments on synthetic data and background subtraction. Finally, Section 5 gives the conclusion.

2. Notations and preliminaries

In this paper, we use $\mathbb{R}^{m\times n}$ to denote the set of all $m \times n$ matrices. For any matrix X, $X_{\cdot j}$ denotes its jth column, $\mathrm{Tr}(A)$ denotes its trace, and $\mathrm{diag}(X)$ denotes the column-vector of its diagonal elements. The Frobenius norm of X is defined as $\|X\|_F = (\sum_{i,j} X_{ij}^2)^{1/2}$. For $X,\ Y \in \mathbb{R}^{m\times n}$, we define $\|(X,Y)\|_F := \sqrt{\|X\|_F^2 + \|Y\|_F^2}$. The ℓ_0-norm denotes the number of nonzero elements of matrix X.

For a real-valued function $f : \mathbb{R}^{m\times n} \to \mathbb{R}$, denote its domain by $\mathrm{dom} f = \{X \in \mathbb{R}^{m\times n} : f(X) < \infty\}$. The function f is said to be proper if its value is not equal to $-\infty$ and $\mathrm{dom} f \neq \emptyset$, and it is said to be closed if it is lower semicontinuous. In what follows, for convenience, we use the notation $\mathcal{L}(L, S) := \frac{\varrho}{2}\|D - L - S\|_F^2$, $\mathcal{M}(L, S) := \mathcal{L}(L, S) + \mathcal{F}(L) + \mathcal{H}(S)$. $\nabla_L \mathcal{L}(\cdot, S)$ is Lipschitz continuous with modulus $L_{\mathcal{F}}$, that is, $\|\nabla_L \mathcal{L}(L_1, S) - \nabla_L \mathcal{L}(L_2, S)\| \leq L_{\mathcal{F}}\|L_1 - L_2\|_F,\ \forall L_1,\ L_2 \in \mathbb{R}^{m\times n}$. Likewise, $\nabla_S \mathcal{L}(L, \cdot)$ is Lipschitz continuous in S with modulus $L_{\mathcal{H}}$.

To establish the optimality condition for problem (1.1), we recall the subdifferential concepts for nonconvex functions in [16, 38].

Definition 2.1. *Let $f : \mathbb{R}^{m\times n} \to \mathbb{R}$ be proper and lower semicontinuous and $X \in \mathrm{dom} f$.*
(i) *The regular subdifferential of f at X is defined by*

$$\hat{\partial} f(X) := \left\{ D \in \mathbb{R}^{m\times n} : \liminf_{Y\neq X, Y\to X} \frac{f(Y) - f(X) - \langle D, Y - X\rangle}{\|Y - X\|_F} \geq 0 \right\}.$$

(ii) *The (limiting-) subdifferential of f at X is defined by*

$$\partial f(X) := \left\{ D \in \mathbb{R}^{m\times n} : \exists X^k \xrightarrow{f} X, D^k \in \hat{\partial} f(X^k) \text{ with } D^k \to D \right\}.$$

From [38, Theorem 10.1], we have $(0, 0) \in \partial \mathcal{M}(\bar{L}, \bar{S})$ at any local minimizer $(\bar{L}, \bar{S})$. By [38, Exercise 8.9], [38, Proposition 10.5], and [38, Exercise 10.10], it holds that $\partial \mathcal{M}(L, S) = (\nabla_L \mathcal{L}(L, S) + \partial \mathcal{F}(L), \nabla_S \mathcal{L}(L, S) + \partial \mathcal{H}(S))$ for all $(L, S) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n}$. Then the necessary optimality condition of problem (1.1) at the global minimizer is described as

$$(0, 0) \in \partial \mathcal{M}(\bar{L}, \bar{S}) = \left(\nabla_L \mathcal{L}(\bar{L}, \bar{S}) + \partial \mathcal{F}(\bar{L}), \nabla_S \mathcal{L}(\bar{L}, \bar{S}) + \partial \mathcal{H}(\bar{S})\right). \tag{2.1}$$

We say that $(L^*, S^*) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n}$ is a critical point of problem (1.1) if (L^*, S^*) satisfies (2.1) in place of $(\bar{L}, \bar{S})$.

We next recall the vector p-thresholding operator associated with the function $|x|^p$, which was established by Peng and Xiu [2]. This operator is useful for deriving explicit solutions to the ℓ_p-regularized subproblem.

Lemma 2.2. *Let $\tau > 0$, $0 < p < 1$. Define*

$$h_\tau(t) := \mathrm{Arg}\min_{x\geq 0} x^p + \frac{1}{2\tau}(x - t)^2, \quad t \in \mathbb{R}. \tag{2.2}$$

Then

$$h_\tau(t) := \begin{cases} h_{\tau,p}(t), & if\ \ t > t^*, \\ \{[2\tau(1-p)]^{1/(2-p)}, 0\}, & if\ \ t = t^*, \\ 0, & if\ \ t < t^*, \end{cases} \tag{2.3}$$

where $t^* = \frac{2-p}{2(1-p)}[2\tau(1-p)]^{1/(2-p)}$, and $h_{\tau,p}(t)$ is the unique root in $(\bar{x}, +\infty)$ of the following equation:

$$p\tau x^{p-1} + x - t = 0, \tag{2.4}$$

where $\bar{x} = [\tau(1-p)]^{1/(2-p)} > 0$, and $h_{\tau,p}(t)$ is differentiable and strictly increasing on $[t^*, +\infty)$.

When $p = 1/2$, it follows from [39, Corollary 2.5] and [40, Theorem 1] that $h_{\tau,1/2}(t) = \frac{2}{3}t\left(1+\cos\left(\frac{2\pi}{3}-\frac{2\phi(t)}{3}\right)\right)$, $\phi(t) = \arccos\left(\frac{\tau}{4}(\frac{t}{3})^{-3/2}\right)$. Consequently, the global solution $x^* = h_\tau(t)$ of problem (2.2) has analytic expression

$$h_\tau(t) := \begin{cases} h_{\tau,1/2}(t), & if \ \ t > t^*, \\ \{\tau^{2/3}, 0\}, & if \ \ t = t^*, \\ 0, & if \ \ t < t^*, \end{cases}$$

where $t^* = \frac{3}{2}\tau^{2/3}$. For a general value of p, as shown in [2], equation (2.4) admits a unique root in $(\bar{x}, +\infty)$. Hence, $h_{\tau,p}(t)$ can be simply calculated by the Newton iteration

$$x_{k+1} = x_k - \frac{p\tau x_k^{p-1} + x_k - t}{p(p-1)\tau x_k^{p-2} + 1}$$

with the initial point $x_0 = 1.5\bar{x}$.

Inspired by the mixed $\ell_0 + \ell_{2,0}$ regularization studied in [3], we present the proximal mapping for the matrix-valued sparse component in the following lemma.

Lemma 2.3. *For any $Z \in \mathbb{R}^{m\times n}$, $\tau > 0$, and $\varphi : \mathbb{R}^{m\times n} \to \mathbb{R}$, the proximal operator of φ is defined by:*

$$Prox_\tau^\varphi(Z) = \mathrm{Arg}\min_{X\in\mathbb{R}^{m\times n}} \left\{ f(X) := \varphi(X) + \frac{1}{2\tau}\|X - Z\|_F^2 \right\}.$$

Then

$$[Prox_\tau^{\lambda_2\ell_0+\lambda_3\ell_{2,0}}(Z)]_{\cdot j} : = \begin{cases} \mathbf{0}, & ||\bar{Z}_{\cdot j}|| < \sqrt{2\tau(\lambda_3 + \lambda_2\|\bar{Z}_{\cdot j}\|_0)}, \\ \{\mathbf{0}, \bar{Z}_{\cdot j}\}, & ||\bar{Z}_{\cdot j}|| = \sqrt{2\tau(\lambda_3 + \lambda_2\|\bar{Z}_{\cdot j}\|_0)}, \\ \bar{Z}_{\cdot j}, & ||\bar{Z}_{\cdot j}|| > \sqrt{2\tau(\lambda_3 + \lambda_2\|\bar{Z}_{\cdot j}\|_0)}, \end{cases} \quad j = 1, \cdots, n. \tag{2.5}$$

where

$$\begin{aligned} \bar{Z}_{\cdot j} : &= [Prox_\tau^{\lambda_2\ell_0}(Z, \sqrt{2\tau\lambda_2})]_{\cdot j}, \\ Prox_\tau^{\lambda_2\ell_0}(Z, \sqrt{2\tau\lambda_2})_{ij} : &= \begin{cases} 0, & |Z_{ij}| < \sqrt{2\tau\lambda_2}, \\ \{0, Z_{ij}\}, & |Z_{ij}| = \sqrt{2\tau\lambda_2}, \\ Z_{ij}, & |Z_{ij}| > \sqrt{2\tau\lambda_2}, \end{cases} \quad i = 1, \cdots, m, j = 1, \cdots, n. \end{aligned}$$

Finally, we recall the Kurdyka-Łojasiewicz (KŁ) property, which plays an important role in the convergence analysis of nonconvex algorithms. It is well known that many important classes of functions, including proper closed semialgebraic functions, satisfy the KŁ property; see [1, 41].

Definition 2.4. *(KŁ property) A proper closed function h is said to satisfy the KŁ property at $\bar{X} \in \mathrm{dom}\partial h$ if there exist $\eta \in (0, \infty]$, a neighborhood $\mathcal{U}$ of $\bar{X}$, and a continuous and concave function $\varphi : [0, \eta) \to \mathbb{R}_+$ such that*

(i) φ *is continuously differentiable in* $(0,\eta)$ *with* $\varphi(0)=0$ *and* $\varphi'(t)>0$ *for all* $t\in(0,\eta)$*;*
(ii) *For any* $X\in\mathcal{U}$ *with* $h(\bar{X})<h(X)<h(\bar{X})+\eta$*, it holds that*

$$\varphi'(h(X)-h(\bar{X}))\mathrm{dist}(0,\partial h(X))\geq 1.$$

A proper closed function h *is called a KŁ function if it satisfies the KŁ property at all points in* $\mathrm{dom}\partial h$.

Lemma 2.5. *[1] (Uniformized KŁ property) Suppose that* h *is a proper closed function and let* Γ *be a compact set. If* h *is a constant on* Γ *and satisfies the KŁ property at each point of* Γ*, then there exist* $\epsilon>0$*,* $\eta>0$ *and a continuous and concave function* $\varphi:[0,\eta)\to\mathbb{R}_+$ *such that*
(i) φ *is continuously differentiable in* $(0,\eta)$ *with* $\varphi(0)=0$ *and* $\varphi'(t)>0$ *for all* $t\in(0,\eta)$*;*
(ii) *For any* $\bar{X}\in\Gamma$ *and any* $X\in\mathcal{U}$ *with* $\mathrm{dist}(X,\Gamma)<\epsilon$ *and* $h(\bar{X})<h(X)<h(\bar{X})+\eta$*, it holds that*

$$\varphi'(h(X)-h(\bar{X}))\mathrm{dist}(0,\partial h(X))\geq 1. \tag{2.6}$$

3. APGP algorithm and convergence analysis

In this section, we propose an APGP for solving problem (1.1). Then we establish the convergence results of APGP under several general assumptions.

3.1. Scheme of APGP algorithm for problem (1.1)

To solve problem (1.1), we develop an APGP. The proposed algorithm is motivated by the two-block structure of the objective function with respect to the coupled variables L and S. Specifically, we adopt an alternating strategy that updates one block while fixing the other. Given an initial point $(L^0,S^0)\in\mathbb{R}^{m\times n}\times\mathbb{R}^{m\times n}$, the iterative sequence $\{(L^k,S^k)\}$ is produced by solving the following two subproblems:

$$\begin{cases} L^{k+1} & \in \arg\min\limits_{L\in\mathbb{R}^{m\times n}} \langle L-L^k,\nabla_L\mathcal{L}(L^k,S^k)\rangle+\frac{1}{2\gamma_1}\|L-L^k\|_F^2+\mathcal{F}(L)\\ S^{k+1} & \in \arg\min\limits_{S\in\mathbb{R}^{m\times n}} \langle S-S^k,\nabla_S\mathcal{L}(L^{k+1},S^k)\rangle+\frac{1}{2\gamma_2}\|S-S^k\|_F^2+\mathcal{H}(S)\end{cases} \tag{3.1}$$

It is easy to verify that the L-subproblem in (3.1) can be reformulated as

$$L^{k+1}\in\arg\min_{L\in\mathbb{R}^{m\times n}}\frac{1}{2\gamma_1}\|L-(L^k-\gamma_1\nabla_L\mathcal{L}(L^k,S^k))\|_F^2+\mathcal{F}(L). \tag{3.2}$$

Let $\bar{L}^k=L^k-\gamma_1\nabla_L\mathcal{L}(L^k,S^k)$, and suppose that the singular value decomposition of $\bar{L}^k$ is given by $\bar{L}^k=U\mathrm{Diag}(\sigma)V^\top$, where $\sigma=(\sigma_1,\cdots,\sigma_{\min\{m,n\}})^\top$. According to Lemma 2.2, it is easy to show that the closed-form solutions of (3.2) can be described as

$$L^{k+1}\in\mathcal{T}_L(\bar{L}^k):=U\mathrm{Diag}(\mathrm{h}_\tau(\sigma_1),\cdots,\mathrm{h}_\tau(\sigma_{\min\{\mathrm{m},\mathrm{n}\}})V^\top, \tag{3.3}$$

where $\tau=\lambda_1\gamma_1$ and $h_\tau(t)$ is the vector p-thresholding operator defined in (2.3).

Moreover, let $\bar{S}^k=S^k-\gamma_2\nabla_S\mathcal{L}(L^{k+1},S^k)$. Then the S-subproblem in (3.1) can be rewritten as

$$S^{k+1}\in\arg\min_{S\in\mathbb{R}^{m\times n}}\tfrac{1}{2\gamma_2}\|S-(S^k-\gamma_2\nabla_S\mathcal{L}(L^{k+1},S^k))\|_F^2+\mathcal{H}(S). \tag{3.4}$$

By Lemma 2.3, the solution of (3.4) has the following closed-form expression:

$$S^{k+1} \in Prox_{\gamma_2}^{\lambda_2\ell_0+\lambda_3\ell_{2,0}}(\bar{S}^k), \tag{3.5}$$

where $Prox_{\gamma_2}^{\lambda_2\ell_0+\lambda_3\ell_{2,0}}(\cdot)$ is given in (2.5).

Then we present the basic framework of APGP for solving (1.1) as follows.

Algorithm 1 APGP for solving (1.1)

Input: Initial point S^0, $L^0 \in \mathbb{R}^{m\times n}$, and the given parameters $\lambda_1, \lambda_2, \lambda_3 > 0, 0 < \gamma_1 < \frac{1}{L_{\mathcal{F}}}, 0 < \gamma_2 < \frac{1}{L_{\mathcal{H}}}$, $xtol = 10^{-4}$. Set $k = 0$.

Step 1. Compute

$$\begin{cases} L^{k+1} & \in \arg\min\limits_{L\in\mathbb{R}^{m\times n}} \langle L - L^k, \nabla_L \mathcal{L}(L^k, S^k)\rangle + \frac{1}{2\gamma_1}\|L - L^k\|_F^2 + \mathcal{F}(L), \\ S^{k+1} & \in \arg\min\limits_{S\in\mathbb{R}^{m\times n}} \langle S - S^k, \nabla_S \mathcal{L}(L^{k+1}, S^k)\rangle + \frac{1}{2\gamma_2}\|S - S^k\|_F^2 + \mathcal{H}(S). \end{cases} \tag{3.6}$$

The calculation process is as follows:

I. Compute L^{k+1} via (3.3);

II. Compute S^{k+1} via (3.5).

Step 2. Let $\Lambda^{k+1} := (L^{k+1}, S^{k+1})$, if $\frac{\|\Lambda^{k+1}-\Lambda^k\|_F}{\max\{1,\|\Lambda^{k+1}\|_F\}} \le xtol$, terminate. Otherwise, let $k := k + 1$, return to **Step 1.**

Output: (L^{k+1}, S^{k+1}).

3.2. Convergence analysis

In this section, we analyze the convergence of the APGP for solving problem (1.1). The set of critical points of problem (1.1) is denoted by crit$\mathcal{M}$. We first introduce the following basic assumptions.

Assumption 3.1. (i) $\inf\{\mathcal{M}(L, S) = \mathcal{L}(L, S) + \mathcal{F}(L) + \mathcal{H}(S)\} > -\infty$.

(ii) $\mathcal{M}(L, S) \to \infty$ *as* $\|(L, S)\| \to \infty$.

(iii) *The parameters satisfy* $0 < \gamma_1 < \frac{1}{L_{\mathcal{F}}}$ *and* $0 < \gamma_2 < \frac{1}{L_{\mathcal{H}}}$.

Under these assumptions, the following auxiliary lemma is proved, showing that the objective function is decreasing along the sequence generated by Algorithm 1.

Lemma 3.2. *Let* $\{(L^k, S^k)\}$ *be a sequence generated by Algorithm 1. Then the sequence* $\{\mathcal{M}(L^k, S^k)\}$ *is decreasing and satisfies*

$$\begin{aligned} &\mathcal{M}(L^{k+1}, S^{k+1}) - \mathcal{M}(L^k, S^k) \\ &\le -\left(\frac{1}{2\gamma_1} - \frac{L_{\mathcal{F}}}{2}\right)\|L^{k+1} - L^k\|_F^2 - \left(\frac{1}{2\gamma_2} - \frac{L_{\mathcal{H}}}{2}\right)\|S^{k+1} - S^k\|_F^2. \end{aligned} \tag{3.7}$$

Proof. It follows from the first inequality in (3.6) that

$$\langle L^{k+1} - L^k, \nabla_L \mathcal{L}(L^k, S^k)\rangle + \frac{1}{2\gamma_1}\|L^{k+1} - L^k\|_F^2 + \mathcal{F}(L^{k+1}) \le \mathcal{F}(L^k). \tag{3.8}$$

Similarly, the second inequality in (3.6) implies that

$$\langle S^{k+1} - S^k, \nabla_S \mathcal{L}(L^{k+1}, S^k)\rangle + \frac{1}{2\gamma_2}\|S^{k+1} - S^k\|_F^2 + \mathcal{H}(S^{k+1}) \le \mathcal{H}(S^k). \tag{3.9}$$

Combining (3.8) and (3.9), we obtain

$$\begin{aligned} &\mathcal{F}(L^{k+1}) + \mathcal{H}(S^{k+1}) \\ &+\langle L^{k+1} - L^k, \nabla_L \mathcal{L}(L^k, S^k)\rangle + \frac{1}{2\gamma_1}\|L^{k+1} - L^k\|_F^2 \\ &+\langle S^{k+1} - S^k, \nabla_S \mathcal{L}(L^{k+1}, S^k)\rangle + \frac{1}{2\gamma_2}\|S^{k+1} - S^k\|_F^2 \le \mathcal{F}(L^k) + \mathcal{H}(S^k). \end{aligned} \tag{3.10}$$

Moreover, using the Lipschitz continuity of $\nabla_L \mathcal{L}(\cdot, S)$ and $\nabla_S \mathcal{L}(L, \cdot)$, we further obtain that

$$\begin{aligned} \mathcal{L}(L^{k+1}, S^k) &\le \mathcal{L}(L^k, S^k) + \langle \nabla_L \mathcal{L}(L^k, S^k), L^{k+1} - L^k\rangle + \frac{L_{\mathcal{F}}}{2}\|L^{k+1} - L^k\|_F^2 \\ \mathcal{L}(L^{k+1}, S^{k+1}) &\le \mathcal{L}(L^{k+1}, S^k) + \langle \nabla_S \mathcal{L}(L^{k+1}, S^k), S^{k+1} - S^k\rangle + \frac{L_{\mathcal{H}}}{2}\|S^{k+1} - S^k\|_F^2. \end{aligned}$$

Rearranging the above two inequalities yields

$$\begin{aligned} \mathcal{L}(L^{k+1}, S^{k+1}) \le \mathcal{L}(L^k, S^k) \quad &+ \quad \langle \nabla_L \mathcal{L}(L^k, S^k), L^{k+1} - L^k\rangle + \frac{L_{\mathcal{F}}}{2}\|L^{k+1} - L^k\|_F^2 \\ &+ \quad \langle \nabla_S \mathcal{L}(L^{k+1}, S^k), S^{k+1} - S^k\rangle + \frac{L_{\mathcal{H}}}{2}\|S^{k+1} - S^k\|_F^2. \end{aligned} \tag{3.11}$$

By (3.10) and (3.11), we conclude that

$$\mathcal{M}(L^{k+1}, S^{k+1}) - \mathcal{M}(L^k, S^k) \le -\left(\frac{1}{2\gamma_1} - \frac{L_{\mathcal{F}}}{2}\right)\|L^{k+1} - L^k\|_F^2 - \left(\frac{1}{2\gamma_2} - \frac{L_{\mathcal{H}}}{2}\right)\|S^{k+1} - S^k\|_F^2.$$

Since $0 < \gamma_1 < \frac{1}{L_{\mathcal{F}}}$ and $0 < \gamma_2 < \frac{1}{L_{\mathcal{H}}}$, it follows that $\frac{1}{2\gamma_1} - \frac{L_{\mathcal{F}}}{2} > 0$, $\frac{1}{2\gamma_2} - \frac{L_{\mathcal{H}}}{2} > 0$. Therefore,

$$\mathcal{M}(L^{k+1}, S^{k+1}) - \mathcal{M}(L^k, S^k) \le 0,$$

which shows that $\{\mathcal{M}(L^k, S^k)\}$ is decreasing and satisfies (3.7). This completes the proof. □

The global subsequential convergence result for Algorithm 1 now follows, together with a characterization of the cluster points of the generated sequence.

Theorem 3.3. *Let $\{(L^k, S^k)\}$ be a sequence generated by Algorithm 1. Then the following statements hold.*

(i) *$\{(L^k, S^k)\}$ is bounded;*

(ii) $\lim_{k\to\infty} \|L^{k+1} - L^k\|_F = 0$, $\lim_{k\to\infty} \|S^{k+1} - S^k\|_F = 0$;

(iii) *Any accumulation point of $\{(L^k, S^k)\}$ is a critical point of problem (1.1).*

Proof. (i) It follows from Lemma 3.2 that the sequence $\{\mathcal{M}(L^k, S^k)\}$ is decreasing. By Assumption 3.1(i), $\mathcal{M}$ is bounded from below, and hence the sequence $\{\mathcal{M}(L^k, S^k)\}$ is convergent. Moreover, $\{(L^k, S^k)\} \subseteq \{(L, S) : \mathcal{M}(L, S) \leq \mathcal{M}(L^0, S^0)\}$ which is bounded due to Assumption 3.1(ii). Therefore, the sequence $\{(L^k, S^k)\}$ is bounded.

(ii) Since $\{(L^k, S^k)\}$ is bounded, there exists at least one accumulation point. Let (L^*, S^*) be an accumulation point of the sequence $\{(L^k, S^k)\}$. Then there exists a convergence subsequence, still denoted by $\{k_j\}$, such that

$$(L^{k_j}, S^{k_j}) \to (L^*, S^*), \quad j \to \infty. \tag{3.12}$$

Base on this and (3.7), it holds that

$$\delta \left(\|L^{k+1} - L^k\|_F^2 + \|S^{k+1} - S^k\|_F^2 \right) \leq \mathcal{M}(L^k, S^k) - \mathcal{M}(L^{k+1}, S^{k+1}),$$

where $\delta = \frac{1}{2} \min\{1/\gamma_1 - L_{\mathcal{F}}, 1/\gamma_2 - L_{\mathcal{H}}\} > 0$. Summing the above inequality from $k = 0$ to ∞, we obtain

$$\begin{aligned} \sum_{k=0}^{\infty} \delta \left(\|L^{k+1} - L^k\|_F^2 + \|S^{k+1} - S^k\|_F^2 \right) &\leq \mathcal{M}(L^0, S^0) - \liminf_{k\to\infty} \mathcal{M}(L^{k+1}, S^{k+1}) \\ &\leq \mathcal{M}(L^0, S^0) - \mathcal{M}(L^*, S^*) < \infty, \end{aligned}$$

then

$$\lim_{k\to\infty} \|L^{k+1} - L^k\|_F = 0, \quad \lim_{k\to\infty} \|S^{k+1} - S^k\|_F = 0. \tag{3.13}$$

(iii) To simplify the notation, we write

$$\begin{aligned} D_L^{k+1} &:= \nabla_L \mathcal{L}(L^{k+1}, S^{k+1}) - \nabla_L \mathcal{L}(L^k, S^k) - \frac{1}{\gamma_1}(L^{k+1} - L^k), \\ D_S^{k+1} &:= \nabla_S \mathcal{L}(L^{k+1}, S^{k+1}) - \nabla_S \mathcal{L}(L^{k+1}, S^k) - \frac{1}{\gamma_2}(S^{k+1} - S^k). \end{aligned}$$

By the first-order optimality conditions for the L-subproblem (3.2), we have

$$0 \in \frac{1}{\gamma_1}(L^{k+1} - L^k) + \nabla_L \mathcal{L}(L^k, S^k) + \partial \mathcal{F}(L^{k+1}). \tag{3.14}$$

Adding $\nabla_L \mathcal{L}(L^{k+1}, S^{k+1})$ to both sides of (3.14) and then rearranging the terms yields

$$\begin{aligned} D_L^{k+1} &= \nabla_L \mathcal{L}(L^{k+1}, S^{k+1}) - \nabla_L \mathcal{L}(L^k, S^k) - \frac{1}{\gamma_1}(L^{k+1} - L^k) \\ &\in \nabla_L \mathcal{L}(L^{k+1}, S^{k+1}) + \partial \mathcal{F}(L^{k+1}). \end{aligned} \tag{3.15}$$

Similarly, the first-order optimality condition associated with the S-subproblem (3.4) is

$$0 \in \frac{1}{\gamma_2}(S^{k+1} - S^k) + \nabla_S \mathcal{L}(L^{k+1}, S^k) + \partial \mathcal{H}(S^{k+1}). \tag{3.16}$$

Adding $\nabla_S \mathcal{L}(L^{k+1}, S^{k+1})$ to both sides of (3.16) and rearranging the terms, we obtain

$$\begin{aligned} D_S^{k+1} &= \nabla_S \mathcal{L}(L^{k+1}, S^{k+1}) - \nabla_S \mathcal{L}(L^{k+1}, S^k) - \frac{1}{\gamma_2}(S^{k+1} - S^k) \\ &\in \nabla_S \mathcal{L}(L^{k+1}, S^{k+1}) + \partial \mathcal{H}(S^{k+1}). \end{aligned} \tag{3.17}$$

Therefore, (3.15) and (3.17) imply that

$$(D_L^{k+1}, D_S^{k+1}) \in \partial \mathcal{M}(L^{k+1}, S^{k+1}), \tag{3.18}$$

where $\partial \mathcal{M}(L^{k+1}, S^{k+1}) := (\nabla_L \mathcal{L}(L^{k+1}, S^{k+1}) + \partial \mathcal{F}(L^{k+1}), \nabla_S \mathcal{L}(L^{k+1}, S^{k+1}) + \partial \mathcal{H}(S^{k+1}))$. Since $\nabla_L \mathcal{L}(\cdot, S)$ is Lipschitz continuous with modulus $L_{\mathcal{F}}$, we have $\|\nabla_L \mathcal{L}(L_1, S) - \nabla_L \mathcal{L}(L_2, S)\| \leq L_{\mathcal{F}} \|L_1 - L_2\|_F,\ \forall L_1,\ L_2 \in \mathbb{R}^{m \times n}$. Likewise, $\nabla_S \mathcal{L}(L, \cdot)$ is Lipschitz continuous with modulus $L_{\mathcal{H}}$. Combining these properties with (3.15), (3.17), we obtain

$$\begin{aligned} \|D_L^{k+1}\|_F &= \|\nabla_L \mathcal{L}(L^{k+1}, S^{k+1}) - \nabla_L \mathcal{L}(L^k, S^k) - \frac{1}{\gamma_1}(L^{k+1} - L^k)\|_F \\ &\leq \|\nabla_L \mathcal{L}(L^{k+1}, S^{k+1}) - \nabla_L \mathcal{L}(L^k, S^k)\|_F + \frac{1}{\gamma_1}\|L^{k+1} - L^k\|_F \\ &\leq L_{\mathcal{F}} \|S^{k+1} - S^k\|_F + (L_{\mathcal{F}} + \frac{1}{\gamma_1})\|L^{k+1} - L^k\|_F, \end{aligned}$$

and

$$\begin{aligned} \|D_S^{k+1}\|_F &= \|\nabla_S \mathcal{L}(L^{k+1}, S^{k+1}) - \nabla_S \mathcal{L}(L^{k+1}, S^k) - \frac{1}{\gamma_2}(S^{k+1} - S^k)\|_F \\ &\leq \|\nabla_S \mathcal{L}(L^{k+1}, S^{k+1}) - \nabla_S \mathcal{L}(L^{k+1}, S^k)\|_F + \frac{1}{\gamma_2}\|S^{k+1} - S^k\|_F \\ &\leq (L_{\mathcal{H}} + \frac{1}{\gamma_2})\|S^{k+1} - S^k\|_F, \end{aligned}$$

Then, we have

$$\begin{aligned} \|(D_L^{k+1}, D_S^{k+1})\|_F^2 &= \|D_L^{k+1}\|_F^2 + \|D_S^{k+1}\|_F^2. \\ &\leq 2L_{\mathcal{F}}^2 \|S^{k+1} - S^k\|_F^2 + 2(L_{\mathcal{F}} + \frac{1}{\gamma_1})^2 \|L^{k+1} - L^k\|_F^2 + (L_{\mathcal{H}} + \frac{1}{\gamma_2})^2 \|S^{k+1} - S^k\|_F^2 \\ &\leq 2(L_{\mathcal{F}} + \frac{1}{\gamma_1})^2 \|L^{k+1} - L^k\|_F^2 + \left(2L_{\mathcal{F}}^2 + (L_{\mathcal{H}} + \frac{1}{\gamma_2})^2\right) \|S^{k+1} - S^k\|_F^2 \\ &\leq \rho^2 (\|L^{k+1} - L^k\|_F^2 + \|S^{k+1} - S^k\|_F^2), \end{aligned} \tag{3.19}$$

where $\rho = \max\{(2(L_{\mathcal{F}} + 1/\gamma_1)^2, 2L_{\mathcal{F}}^2 + (L_{\mathcal{H}} + 1/\gamma_2)^2\}$. (3.19) implies that

$$\|(D_L^{k+1}, D_S^{k+1})\|_F \leq \rho \|(L^{k+1} - L^k, S^{k+1} - S^k)\|_F. \tag{3.20}$$

By (3.12), (3.13), and (3.20), let $j \to \infty$, thus we have

$$\lim_{j \to \infty} \|(D_L^{k_j+1}, D_S^{k_j+1})\|_F = 0,$$

and hence

$$(D_L^{k_j+1}, D_S^{k_j+1}) \to (0,0),\ j \to \infty. \tag{3.21}$$

From the second inequality in (3.6), taking $S = S^*$ as a comparison point yields

$$\begin{aligned}&\left\langle S^{k_j+1} - S^{k_j}, \nabla_S \mathcal{L}(L^{k_j+1}, S^{k_j})\right\rangle + \frac{1}{2\gamma_2}\|S^{k_j+1} - S^{k_j}\|_F^2 + \mathcal{H}(S^{k_j+1})\\ &\le \left\langle S^* - S^{k_j}, \nabla_S \mathcal{L}(L^{k_j+1}, S^{k_j})\right\rangle + \frac{1}{2\gamma_2}\|S^* - S^{k_j}\|_F^2 + \mathcal{H}(S^*).\end{aligned}$$

Since $\{(L^k, S^k)\}$ is bounded and $\nabla_S \mathcal{L}$ is continuous, $\nabla_S \mathcal{L}(L^{k_j+1}, S^{k_j})$ is bounded. Together with $S^{k_j} \to S^*$ and $\|S^{k_j+1} - S^{k_j}\|_F \to 0$, taking the limit superior on both sides gives

$$\limsup_{j\to\infty} \mathcal{H}(S^{k_j+1}) \le \mathcal{H}(S^*).$$

Together with the lower semicontinuity of $\mathcal{H}$, we have $\lim_{j\to\infty} \mathcal{H}(S^{k_j+1}) = \mathcal{H}(S^*)$. Since $\mathcal{L}$ and $\mathcal{F}$ are continuous, it follows that

$$\begin{aligned}\lim_{j\to\infty} \mathcal{M}(L^{k_j+1}, S^{k_j+1}) &= \lim_{j\to\infty} \mathcal{L}(L^{k_j+1}, S^{k_j+1}) + \mathcal{F}(L^{k_j+1}) + \mathcal{H}(S^{k_j+1})\\ &= \mathcal{M}(L^*, S^*).\end{aligned} \tag{3.22}$$

This means that

$$\lim_{j\to\infty} \mathcal{F}(L^{k_j+1}) = \mathcal{F}(L^*),\ \lim_{j\to\infty} \mathcal{H}(S^{k_j+1}) = \mathcal{H}(S^*). \tag{3.23}$$

Therefore, by (3.12), (3.15), (3.17), (3.18), (3.21), (3.23), and the outer semi-continuity of the subdifferential mapping $\partial\mathcal{F}(\cdot)$ and $\partial\mathcal{H}(\cdot)$, letting $j \to \infty$ yields

$$(0,0) \in \partial\mathcal{M}(L^*, S^*). \tag{3.24}$$

Hence, (L^*, S^*) is a critical point of problem (1.1). □

Let $\Gamma(\{(L^k, S^k)\})$ denote the set of all limit points of the sequence $\{(L^k, S^k)\}$, namely,

$$\Gamma(\{(L^k, S^k)\}) = \left\{(L^*, S^*) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} : \exists\{k_j\} \in \mathbb{N},\ s.t.\ (L^{k_j}, S^{k_j}) \to (L^*, S^*),\ j \to \infty\right\}.$$

Then, within the framework of the KŁ property, we establish the global convergence of the whole sequence $\{(L^k, S^k)\}_{k\ge1}$ generated by the proposed APGP algorithm.

Theorem 3.4. *Assume that $\mathcal{M}$ satisfies the KŁ property. Let $\{(L^k, S^k)\}_{k\ge1}$ be a sequence generated by Algorithm 1. Then the sequence $\{(L^k, S^k)\}_{k\ge1}$ converges to a critical point of problem (1.1).*

Proof. By the boundedness of $\{(L^k, S^k)\}$, the set $\Gamma(\{(L^k, S^k)\})$ is nonempty. Note that $\Gamma(\{(L^k, S^k)\})$ can be represented as an intersection of compact sets $\Gamma(\{(L^k, S^k)\}) = \bigcap_{t\in\mathbb{N}} \overline{\bigcup_{k\ge t}\{(L^k, S^k)\}}$. Since the intersection of bounded closed sets is still bounded and closed, it is also compact. Moreover,

since $\{\mathcal{M}(L^k, S^k)\}_{k\geq 1}$ is bounded from below and decreasing by Lemma 3.2, we conclude that $\mathcal{M}^* := \lim_{k\to\infty} \mathcal{M}(L^k, S^k)$ exists. $\mathcal{M}(L^k, S^k)\}$ is decreasing to $\mathcal{M}^*$, then $\mathcal{M}(L^k, S^k) \geq \mathcal{M}^*$.

We first consider the case where there exists a $\bar{k} \geq 0$ such that $\mathcal{M}(L^k, S^k) = \mathcal{M}^*$ for all $k \geq \bar{k}$. Since $\{\mathcal{M}(L^k, S^k)\}$ is decreasing and converges to $\mathcal{M}^*$, then $\mathcal{M}(L^k, S^k) \equiv \mathcal{M}^*$ for all $k \geq \bar{k}$. It then follows from Lemma 3.2 that $L^{k+\bar{k}} = L^{\bar{k}}$ and $S^{k+\bar{k}} = S^{\bar{k}}$ for all $k \geq 1$. Hence, $\{(L^k, S^k)\}$ is convergent.

We next consider the case where $\mathcal{M}(L^k, S^k) > \mathcal{M}^*$. Define $\Omega := \Gamma(\{(L^k, S^k)\})$, then we have $\lim_{k\to\infty} \operatorname{dist}((L^k, S^k), \Omega) = 0$. Hence, for any $\epsilon > 0$, there exists $k_1 > 0$ such that

$$\operatorname{dist}((L^k, S^k), \Omega) < \epsilon$$

for any $k \geq k_1$. Moreover, by (3.22) and the definition of Ω, we obtain that $\mathcal{M}$ is a constant on Ω. Therefore, for any $\eta > 0$, there exists a nonnegative integer k_2 such that $\mathcal{M}^* < \mathcal{M}(L^k, S^k) < \mathcal{M}^* + \eta$ for all $k \geq k_2$. Since $\mathcal{M}$ is a KŁ function, by the property of the KŁ function, there exist an $\epsilon > 0$, $\eta > 0$, a desingularization function φ such that for all $\{(L^k, S^k)\}$ satisfying

$$\begin{aligned}&\left\{(L, S) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} : \operatorname{dist}((L, S), \Omega) < \epsilon\right\} \bigcap \\ &\left\{(L, S) \in \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} : \mathcal{M}^* < \mathcal{M}(L, S) < \mathcal{M}^* + \eta\right\},\end{aligned}$$

it holds that

$$\varphi'(\mathcal{M}(L^k, S^k) - \mathcal{M}^*) \cdot \operatorname{dist}((0, 0), \partial\mathcal{M}(L^k, S^k)) \geq 1, \tag{3.25}$$

for all $k \geq k_3 := \max\{k_1, k_2\}$. From (3.20) and (3.25), we obtain

$$\varphi'(\mathcal{M}(L^k, S^k) - \mathcal{M}^*) \geq \frac{1}{\rho\|(L^k - L^{k-1}, S^k - S^{k-1})\|_F}. \tag{3.26}$$

By use of the concavity of φ, it follows that

$$\begin{aligned}\varphi(\mathcal{M}(L^{k+1}, S^{k+1}) - \mathcal{M}^*) \quad \leq \quad & \varphi(\mathcal{M}(L^k, S^k) - \mathcal{M}^*) + \\ & \varphi'(\mathcal{M}(L^k, S^k) - \mathcal{M}^*)(\mathcal{M}(L^{k+1}, S^{k+1}) - \mathcal{M}(L^k, S^k)).\end{aligned} \tag{3.27}$$

Combining Lemma 3.2 with (3.26) and (3.27), we deduce that for any $k \geq k_3$,

$$\Lambda^k - \Lambda^{k+1} \geq \frac{\|(L^{k+1} - L^k, S^{k+1} - S^k)\|_F^2}{T\|(L^k - L^{k-1}, S^k - S^{k-1})\|_F},$$

where $\Lambda^k = \varphi(\mathcal{M}(L^k, S^k) - \mathcal{M}^*)$, $T = \rho/\alpha$, and $\alpha = \frac{1}{2}\min\left\{\gamma_1^{-1} - L_{\mathcal{F}}, \gamma_2^{-1} - L_{\mathcal{H}}\right\}$. Hence

$$\|(L^{k+1} - L^k, S^{k+1} - S^k)\|_F^2 \leq T(\Lambda^k - \Lambda^{k+1})\|(L^k - L^{k-1}, S^k - S^{k-1})\|_F.$$

Using the inequality $\sqrt{ab} \leq \frac{a+b}{2}$ and rearranging terms, we obtain

$$\begin{aligned}&\|(L^{k+1} - L^k, S^{k+1} - S^k)\|_F \\ \leq \quad & \|(L^k - L^{k-1}, S^k - S^{k-1})\|_F - \|(L^{k+1} - L^k, S^{k+1} - S^k)\|_F + T(\Lambda^k - \Lambda^{k+1}).\end{aligned} \tag{3.28}$$

Summing both sides of (3.28) from $k = k_3$ to ∞, we get

$$\begin{aligned} & \sum_{k=k_3+1}^{\infty} \|(L^{k+1} - L^k, S^{k+1} - S^k)\|_F \\ \leq\ & \|(L^{k_3+1} - L^{k_3}, S^{k_3+1} - S^{k_3})\|_F + 2T\varphi(\mathcal{M}(L^{k_3+1}, S^{k_3+1}) - \mathcal{M}^*) \\ <\ & \infty. \end{aligned} \tag{3.29}$$

This implies that

$$\sum_{k=0}^{\infty} \|L^{k+1} - L^k\|_F < \infty, \sum_{k=0}^{\infty} \|S^{k+1} - S^k\|_F < \infty.$$

The above inequality implies that $\{(L^k, S^k)\}_{k\geq 1}$ is a Cauchy sequence and hence converges to an element (L^*, S^*), which is a critical point of problem (1.1). □

Theorem 3.5 (Convergence rate)**.** *Let $\{(L^k, S^k)\}$ be a sequence generated by Algorithm 1. Suppose that $\mathcal{M}$ satisfies the KŁ property at (L^*, S^*) with $\varphi(s) = cs^{1-\theta}$, $\theta \in [0, 1)$, $c > 0$. Then the following statements hold.*

(i) *If $\theta = 0$, then the sequence $\{(L^k, S^k)\}$ converges in a finite number of steps.*

(ii) *If $\theta \in (0, \frac{1}{2}]$, then there exist $c > 0$ and $\tau \in [0, 1)$ such that*

$$\|(L^k, S^k) - (L^*, S^*)\| \leq c\tau^k.$$

(iii) *If $\theta \in (\frac{1}{2}, 1)$, then there exist $c > 0$ such that*

$$\|(L^k, S^k) - (L^*, S^*)\| \leq ck^{-\frac{1-\theta}{2\theta-1}}.$$

Proof. From (3.22) and the definition of Ω, we obtain that $\mathcal{M}$ is a constant on Ω. Without loss of generality, we may assume that $\mathcal{M}(L^*, S^*) = 0$ (replace if necessary $\mathcal{M}$ by $\mathcal{M} - \mathcal{M}(L^*, S^*)$).

(i) Consider the case $\theta = 0$. If $\{\mathcal{M}(L^k, S^k)\}$ is stationary, then Lemma 3.2 implies that $\{(L^k, S^k)\}$ is stationary as well. If $\{\mathcal{M}(L^k, S^k)\}$ is not stationary, the KŁ inequality yields, for all sufficiently large k, $c\text{dist}((0, 0), \partial\mathcal{M}(L^k, S^k)) \geq 1$. This contradicts (3.18), (3.21), and (3.24), which imply $\text{dist}((0, 0), \partial\mathcal{M}(L^k, S^k)) \to 0$.

(ii) Consider the case $0 < \theta < 1$. For any $k \geq 0$, define

$$\triangle_k := \sum_{j=k}^{\infty} \|(L^{j+1} - L^j, S^{j+1} - S^j)\|_F.$$

By Theorems 3.3 and 3.4, $\triangle_k$ is finite. Moreover, since $\triangle_k \geq \|(L^k - L^*, S^k - S^*)\|_F$, it remains only to estimate $\triangle_k$. With this notation, inequality (3.28) can be rewritten as

$$\triangle_k \leq 2M\varphi(\mathcal{M}(L^k, S^k)) + (\triangle_{k-1} - \triangle_k). \tag{3.30}$$

The KŁ inequality further gives

$$\varphi'(\mathcal{M}(L^k, S^k)\text{dist}((0, 0), \partial\mathcal{M}(\mathrm{L^k}, \mathrm{S^k})) = c(1 - \theta)\mathcal{M}(L^k, S^k)^{-\theta}\text{dist}((0, 0), \partial\mathcal{M}(\mathrm{L^k}, \mathrm{S^k})) \geq 1,$$

then

$$\mathcal{M}(L^k, S^k)^\theta \le c(1-\theta)\mathrm{dist}((0,0), \partial\mathcal{M}(\mathrm{L^k}, \mathrm{S^k})).$$

From (3.18) and (3.20), we obtain

$$\mathrm{dist}((0,0), \partial\mathcal{M}(\mathrm{L^k}, \mathrm{S^k})) \le \rho(\triangle_{k-1} - \triangle_k).$$

Combining the preceding two estimates, there exists a constant $K/2 > 0$ such that

$$\varphi(\mathcal{M}(L^k, S^k)) = c\mathcal{M}(L^k, S^k)^{1-\theta} \le \frac{K}{2}(\triangle_{k-1} - \triangle_k)^{\frac{1-\theta}{\theta}}.$$

Substituting this estimate into (3.30) yields

$$\triangle_k \le MK(\triangle_{k-1} - \triangle_k)^{\frac{1-\theta}{\theta}} + (\triangle_{k-1} - \triangle_k). \tag{3.31}$$

By Theorem 3.3-(ii), we obtain

$$\triangle_{k-1} - \triangle_k = \|(L^k - L^{k-1}, S^k - S^{k-1})\|_F \to 0. \tag{3.32}$$

If $0 < \theta \le \frac{1}{2}$, then $\frac{1-\theta}{\theta} \ge 1$. By (3.31) and (3.32), we have

$$\begin{aligned} \triangle_k &\le MK(\triangle_{k-1} - \triangle_k)^{\frac{1-\theta}{\theta}} + (\triangle_{k-1} - \triangle_k) \\ &\le MK(\triangle_{k-1} - \triangle_k) + (\triangle_{k-1} - \triangle_k) = (MK+1)(\triangle_{k-1} - \triangle_k) \end{aligned}$$

for all sufficiently large k. Therefore, $\triangle_k \le \frac{MK+1}{MK+2}\triangle_{k-1}$. Let $\tau := \frac{MK+1}{MK+2} \in (0,1)$, then

$$\begin{aligned} \|(L^k - L^*, S^k - S^*)\|_F &\le \sum_{j=k}^{\infty} \|(L^{j+1} - L^j, S^{j+1} - S^j)\|_F \\ &\le \triangle_k \le \frac{MK+1}{MK+2}\triangle_{k-1} \le \left(\frac{MK+1}{MK+2}\right)^2 \triangle_{k-2} \le \cdots \le \left(\frac{MK+1}{MK+2}\right)^k \triangle_0 = c\tau^k, \end{aligned}$$

where $c := \triangle_0 > 0$.

(iii) If $\frac{1}{2} < \theta < 1$, then $\frac{1-\theta}{\theta} \in (0,1)$. By (3.31) and (3.32), we obtain

$$\begin{aligned} \triangle_k &\le MK(\triangle_{k-1} - \triangle_k)^{\frac{1-\theta}{\theta}} + (\triangle_{k-1} - \triangle_k) \\ &\le MK(\triangle_{k-1} - \triangle_k)^{\frac{1-\theta}{\theta}} + (\triangle_{k-1} - \triangle_k)^{\frac{1-\theta}{\theta}} = (MK+1)(\triangle_{k-1} - \triangle_k)^{\frac{1-\theta}{\theta}} \end{aligned}$$

for all sufficiently large k. Since $\frac{\theta}{1-\theta} > 1$, the above inequality implies

$$\triangle_k^{\frac{\theta}{1-\theta}} \le M_1(\triangle_{k-1} - \triangle_k),$$

for all sufficiently large k, where $M_1 := (MK+1)^{\frac{\theta}{1-\theta}} > 0$. Therefore, by [42, Theorem 2], $\triangle_k \le M_2 k^{-\frac{1-\theta}{2\theta-1}} = M_2 k^{-\varsigma}$ for all sufficiently large k, where $M_2 > 0$, $\varsigma := \frac{1-\theta}{2\theta-1} > 0$. Consequently, $\|(L^k - L^*, S^k - S^*)\|_F \le \triangle_k \le M_2 k^{-\varsigma}$ for all sufficiently large k. Thus, the proof is complete. □

3.3. Partial-SVD strategy

In Algorithm 1, the singular value decomposition involved in the (L)-update constitutes the major computational cost. Inspired by existing partial-SVD techniques, instead of computing a full SVD at each iteration, we compute only a rank-(r) approximation of the matrix involved in the (L)-update, where (r) is an estimate of the rank of the low-rank component. Specifically, we employ the linear-time approximate SVD algorithm in [43] to efficiently perform the partial SVD. The code used in our experiments was developed by Shiqian Ma (see [43]) and is available at http://www.columbia.edu/~sm2756/FPCA.htm.

4. Numerical experiments

In this section, we present numerical experiments on both simulated and real datasets to show the performances of our proposed model and algorithm. The parameters in APGP are set as follows: $\lambda_1 = \sqrt{\max(m,n)}/2$, $\lambda_2 = 1/\sqrt{\max(m,n)}$, $\lambda_3 = 1/\sqrt{\max(m,n)}$, $\gamma_1 = 1/\|D\|_2$, $\gamma_2 = \gamma_1$, $xtol = 10^{-4}$, and $MaxIter = 1000$. The principal component pursuit (PCP) [15] and the weighted nuclear norm minimization (WNNM) [44] are used for comparison with the default parameter settings recommended in their original references. For a fair comparison, all methods adopt the same stopping rule: The iteration is terminated when the relative-change criterion with tolerance $xtol$ is satisfied or when the maximum number of iterations is reached. To determine $h_{\tau,p}(t)$ in (2.3), the Newton iteration is performed with $InnerMaxIter = 10$ and $Innerxtol = 10^{-4}$, where the inner iteration is terminated when $abs(x^{k+1} - x^k) < Innerxtol$ or when the maximum number of inner iterations is reached. All computational experiments are run in MATLAB R2018b under Windows 11 on a desktop of a 3.00GHz CPU and 32.00GB memory.

4.1. Experimental results on simulated data

In the simulated tests, each instance consists of a low-rank component and a mix-sparse component. The low-rank component is generated by $L = M_L M_R^\top$, where $M_L \in \mathbb{R}^{m\times r}$ and $M_R \in \mathbb{R}^{n\times r}$ are independently sampled from the standard Gaussian distribution $\mathcal{N}(0,1)$. The rank is determined by $r = p_r \cdot \min\{m,n\}$. To generate the mix-sparse matrix S, we randomly choose $p_e \cdot \min\{m,n\}$ groups as nonzero groups (NZG). Within each selected group, $p_{re} \cdot \min\{m,n\}$ entries are randomly selected to be nonzero entries (NZE), whose values are independently sampled from the standard Gaussian distribution $\mathcal{N}(0,1)$. The observed matrix is then constructed as $D = L + S$.

For the numerical experiments, we set $m = n = 200$, $(p_e, p_{re}) = (10\%, 10\%)$. The true rank was varied from 4 to 40 with an increment of 2. To select a suitable value of p, we tested five different choices, namely $p = 0.1, 0.3, 0.5, 0.7, 0.9$. For each parameter setting, 10 random instances were generated. We recorded the average CPU time, the average number of iterations, the average relative error (Rel.err), and the frequency of success. Here $Rel.err = \|L - L_{opt}\|_F/\|L\|_F$, L_{opt} denotes the numerical solution to problem (1.1). A test instance is counted as successful if the computed solution L_{opt} satisfies $\mathrm{rank}(L) = \mathrm{rank}(L_{opt})$ and $Rel.err < 10^{-4}$. The computational results are displayed in Figure 1. As shown in Figure 1, the APGP algorithm with $p = 0.1$ and $p = 0.3$ generally performs better than the other choices of p. Both settings are among the fastest and achieve almost the same frequency of success, which is higher than those obtained with $p = 0.5$, $p = 0.7$, and $p = 0.9$. Moreover, the average number of iterations and the average CPU time for

$p = 0.1$ are close to those for $p = 0.3$. However, compared with $p = 0.1$, $p = 0.3$ yields a smaller average relative error and a higher frequency of success. Thus, in the remaining tests, we always used $p = 0.3$.

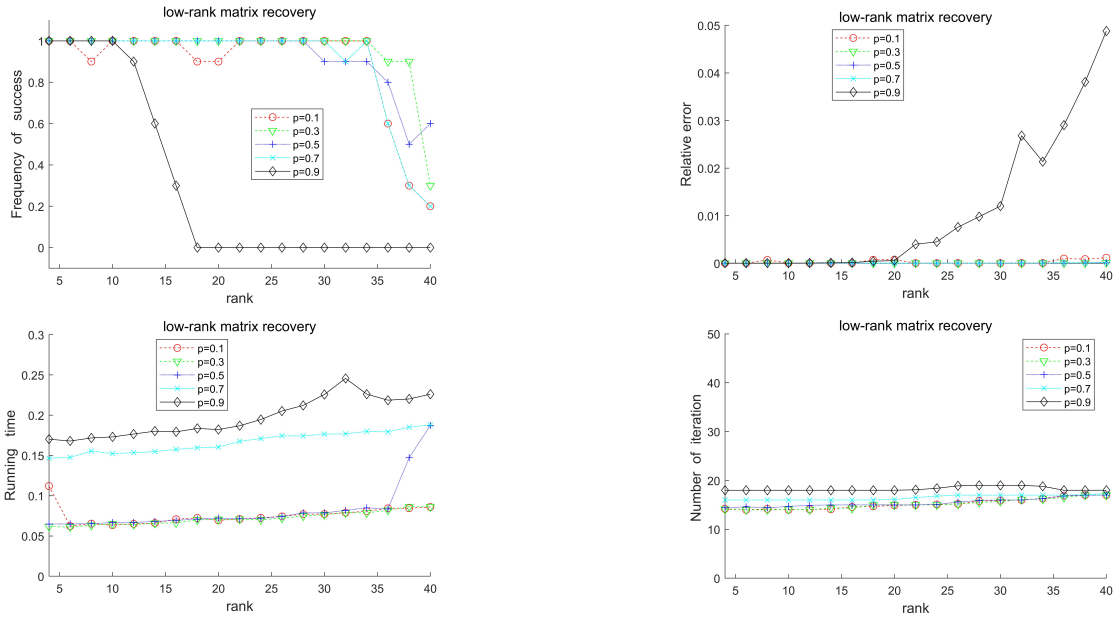
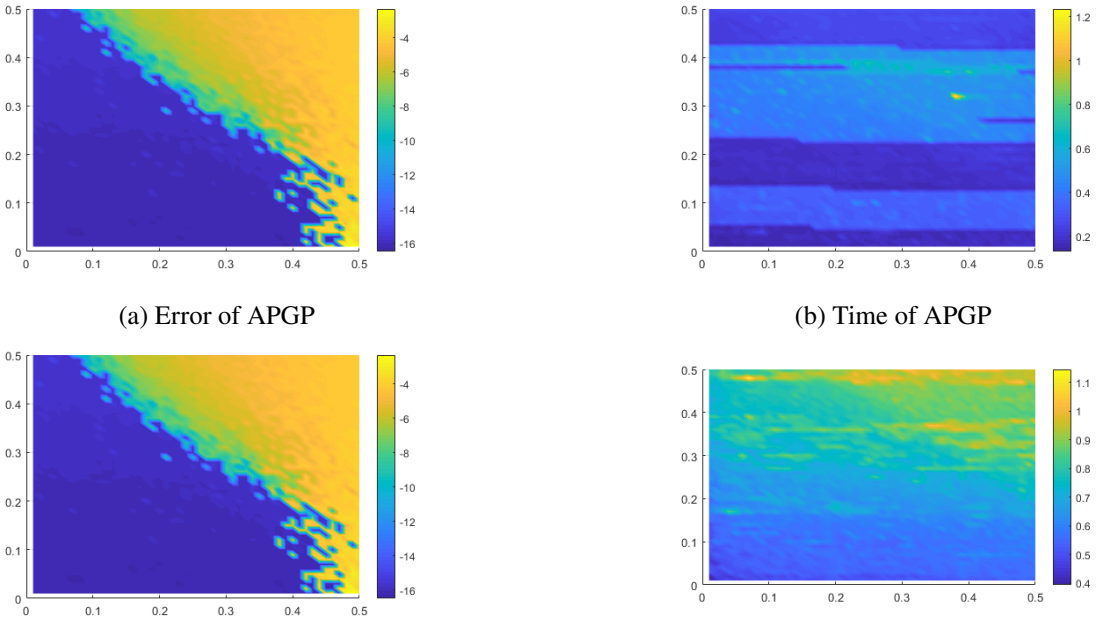


Figure 1. Numerical performance of APGP under different choices of p.



(a) Error of APGP

(b) Time of APGP

(c) Error of APGP_full

(d) Time of APGP_full

Figure 2. Numerical performance of APGP and APGP_full under different values of (p_r, p_{re}).

Similarly, we also set $m = n = 200$, $p_e = 5\%$, and let p_r and p_{re} vary from 0.01 to 0.5 with step length 0.01, respectively. For each parameter setting, we run 10 independent instances and report the

average results. Meanwhile, we plot the log-scale relative error maps with respect to the different values of (p_r, p_{re}), as presented in Figure 2. In these maps, a larger blue area indicates a higher success rate of matrix decomposition. As shown in Figure 2, APGP and APGP_full have nearly identical successful regions, while APGP exhibits generally lower computational times, as indicated by the larger blue regions in Figure 2. Here, APGP_full uses MATLAB's built-in SVD for full SVD. Specifically, we set $m = n = 200$, $p_e = 5\%$, $p_r = p_{re} = 10\%$, $NZG = 10$, $NZE = 200$, and $r = 20$. We recorded the CPU time, NZG and NZE, and the relative errors of the recovered low-rank and sparse components, defined as $Rel.err(L) = \frac{\|L-L_{opt}\|_F}{\|L_{opt}\|_F}$ and $Rel.err(S) = \frac{\|S-S_{opt}\|_F}{\|S_{opt}\|_F}$, where L_{opt} and S_{opt} denote the ground-truth low-rank and sparse components, respectively. The results are presented in Table 1. It can be seen that the ℓ_0-only model is effective by exploiting element-wise sparsity, while the $\ell_{2,0}$-only model shows inferior recovery performance. In contrast, the proposed mixed regularization effectively exploits both element-wise and group-wise sparsity, achieving the best recovery accurate and NZG/NZE identification.

Table 1. Ablation study of different sparsity regularizations.

Method	Regularizations	Rel.err(L)	Rel.err(S)	Time	NZG	NZE	r
APGP-ℓ_0	$\lambda_2\ell_0$	1.41e-5	1.31e-3	0.18	10	196	20
APGP-$\ell_{2,0}$	$\lambda_3\ell_{2,0}$	6.63e-3	4.19e-1	0.22	200	26400	20
APGP-Mixed	$\lambda_2\ell_0 + \lambda_3\ell_{2,0}$	1.38e-5	1.20e-3	0.18	10	200	20

In the following numerical experiments, the proposed APGP algorithm is compared with PCP [15], WNNM [44]. To ensure a comprehensive comparison, the selected instances cover both small-scale ($m \leq 1000$) and large-scale ($m > 1000$) problems. The noise parameter pair is set to $(p_e, p_{re}) = (10\%, 10\%)$. For each parameter setting, we generate 10 independent random instances and report the averaged results. The parameter settings and corresponding results are summarized in Table 2 and 3.

For small-scale problems ($m \leq 1000$), the results in Table 2 show that APGP provides consistently high recovery accuracy across all tested instances. WNNM maintains competitive accuracy but generally requires more CPU time. PCP exhibits relatively lower accuracy, especially for high-rank cases. In terms of computational efficiency, APGP requires considerably less CPU time than WNNM, while maintaining stable and accurate recovery results.

For large-scale problems ($m > 1000$), as shown in Table 3, the performance differences among the methods become increasingly pronounced with the growth of problem size. WNNM exhibits similar scalability limitations, with increasing runtime and occasional "out-of-memory" failures as the matrix size increases. PCP remains applicable across all tested settings, but its computational burden becomes prohibitively high for large-scale problems. In contrast, APGP achieves competitive recovery accuracy and exhibits better scalability for large-scale problems. For challenging large-scale cases, APGP requires substantially less computational time than PCP while avoiding the memory limitations encountered by WNNM. For example, when $m = n = 7000$ and $r = 1500$, APGP achieves a relative error of 1.71×10^{-4} in 2557.02 seconds, whereas PCP requires 23395.81 seconds and WNNM fails due to memory limitations. These results demonstrate the superior scalability and overall efficiency of APGP for large-scale low-rank matrix recovery.

Table 2. Low-rank matrix recovery on small-scale problems.

Problems		Rel.err			Time		
$m = n$	r	PCP	WNNM	APGP	PCP	WNNM	APGP
100	6	2.46e-4	4.99e-5	**3.21e-5**	**0.02**	0.05	0.03
	12	3.84e-4	6.32e-5	**3.81e-5**	**0.02**	0.07	0.04
	18	6.20e-4	8.10e-5	**4.39e-5**	**0.04**	0.07	**0.04**
	24	1.82e-3	1.01e-4	**5.39e-5**	**0.05**	0.07	**0.05**
	30	6.60e-3	1.22e-4	**6.75e-5**	0.08	0.08	**0.05**
200	12	1.75e-4	4.22e-5	**2.56e-5**	**0.09**	0.25	0.15
	24	2.68e-4	6.31e-5	**3.42e-5**	**0.11**	0.29	0.16
	36	3.58e-4	8.10e-5	**4.29e-5**	**0.14**	0.29	0.17
	48	7.52e-4	9.95e-5	**5.26e-5**	0.19	0.31	**0.17**
	60	2.85e-3	1.19e-4	**6.41e-5**	0.27	0.32	**0.18**
500	30	1.36e-4	4.09e-5	**2.24e-5**	**0.77**	1.73	1.70
	60	2.31e-4	6.13e-5	**3.22e-5**	**1.69**	4.04	2.13
	90	2.86e-4	8.02e-5	**4.10e-5**	2.56	5.87	**2.42**
	120	4.41e-4	9.78e-5	**5.21e-5**	2.87	9.25	**2.36**
	150	1.36e-3	1.17e-4	**6.02e-5**	3.60	11.94	**2.53**
1000	60	1.19e-4	4.13e-5	**2.15e-5**	**3.89**	6.38	5.75
	120	1.91e-4	6.17e-5	**3.25e-5**	**5.13**	13.28	5.87
	180	2.50e-4	7.88e-5	**4.20e-5**	**6.17**	29.51	6.98
	240	3.70e-4	9.76e-5	**5.54e-5**	43.54	98.69	**25.81**
	300	8.50e-4	1.18e-4	**6.73e-5**	23.48	120.74	27.72

Overall, APGP exhibits high consistent performance on both small-scale and large-scale problems. Its performance advantage demonstrates an excellent balance between solution quality and computational efficiency.

4.2. Experimental results on background subtraction

In this subsection, we apply the proposed APGP to background subtraction and evaluate its practical performance on the CDnet dataset [45], I2R dataset [46], and BMC dataset [47]. For all datasets, grayscale video sequences are used for matrix decomposition. Given a sequence of size $h \times w \times T$, each $h \times w$ grayscale frame is vectorized into a column of $D \in \mathbb{R}^{hw \times T}$. The RGB images shown in the figures are used only for visualization and are not involved in the computation. The performance of APGP is compared with those of PCP and WNNM. The 45th frame is selected, and a binary foreground mask is obtained by applying a fixed threshold of 0.1 to the absolute values of the recovered sparse component. The resulting mask is then compared with the corresponding ground-truth mask, and the F-measure reported in Table 3 is calculated for the 45th frame of each sequence. F-measure [48] and CPU time are

used as evaluation criteria, where a higher F-measure indicates more accurate foreground-background separation and a shorter CPU time indicates higher computational efficiency.

Table 3. Low-rank matrix recovery via PCP, WNNM, and APGP for large-scale problems.

Problems		Rel.err			Time		
$m = n$	r	PCP	WNNM	APGP	PCP	WNNM	APGP
2000	120	9.49e-5	4.19e-5	**2.29e-5**	**24.36**	41.92	37.56
	240	1.68e-4	6.10e-5	**2.86e-5**	**32.02**	79.86	32.47
	360	2.23e-4	7.71e-5	**4.23e-5**	**48.38**	173.98	54.62
	480	3.10e-4	9.41e-5	**4.83e-5**	79.14	258.88	**55.79**
	600	6.72e-4	1.19e-4	**5.92e-5**	104.80	371.60	**55.36**
3000	180	9.38e-5	4.07e-5	**1.91e-5**	164.23	**158.80**	197.76
	360	1.60e-4	5.93e-5	**3.19e-5**	193.29	273.90	**142.65**
	540	2.18e-4	7.80e-5	**4.09e-5**	182.07	430.94	**146.60**
	720	2.98e-4	9.71e-5	**4.69e-5**	232.22	663.48	**154.31**
	900	6.05e-4	1.13e-4	**5.83e-5**	936.75	1810.41	**262.46**
5000	300	5.40e-5	3.99e-5	**1.99e-5**	1185.53	**744.20**	929.64
	600	1.45e-4	5.83e-5	**2.96e-5**	1008.49	1108.76	**858.38**
	900	1.87e-4	7.71e-5	**4.29e-5**	1242.03	1976.31	**868.23**
	1200	2.52e-4	9.55e-5	**4.90e-5**	1591.20	3179.34	**897.17**
	1500	5.47e-4	1.17e-4	**5.98e-5**	3369.22	5867.55	**1383.83**
7000	300	4.59e-5	4.07e-5	**3.61e-5**	9477.37	4287.37	**2055.47**
	600	1.39e-4	5.90e-5	**2.95e-5**	10829.59	11070.57	**2233.13**
	900	1.86e-4	7.78e-5	**6.67e-5**	13165.39	21438.77	**2223.74**
	1200	2.47e-4	-	**7.87e-5**	16956.87	-	**2416.15**
	1500	5.03e-4	-	**1.71e-4**	23395.81	-	**2557.02**

The sign "−" denotes "out of memory".

Six short-video sequences from the CDnet and I2R datasets are used. Among them, the fountain01 and fountain02 from CDnet have a size of $288 \times 432 \times 80$, while Campus, Escalator, Lobby, and ShoppingMall from I2R have a size of $128 \times 160 \times 80$. Table 4 and Figure 3 list the selected video instances, where "Frame" refers to a representative frame selected from the corresponding video for result comparison. From Table 4, we observe that APGP attains the highest or tied highest F-measure among all compared algorithms for background subtraction in short videos. It also maintains the lowest or near-lowest CPU time. The visual results in Figure 3 further show that APGP introduces less background noise and preserves more complete foreground objects than PCP and WNNM. These findings confirm that APGP can effectively balance foreground-background separation accuracy and computational efficiency in short-video background subtraction.

Table 4. Numerical results of background subtraction on the CDnet and I2R short-video dataset.

Frame	F-measure			Time		
	PCP	WNNM	APGP	PCP	WNNM	APGP
Campus	0.77	0.78	**0.80**	10.69	8.72	**6.30**
Escalator	0.82	0.83	**0.84**	22.23	7.41	**7.26**
fountain01	**0.76**	**0.76**	**0.76**	74.57	30.56	**28.91**
fountain02	0.60	0.64	**0.66**	86.84	**28.92**	31.95
Lobby	0.89	**0.90**	**0.90**	**7.48**	9.89	8.22
ShoppingMall	0.60	0.63	**0.67**	9.75	9.67	**9.66**

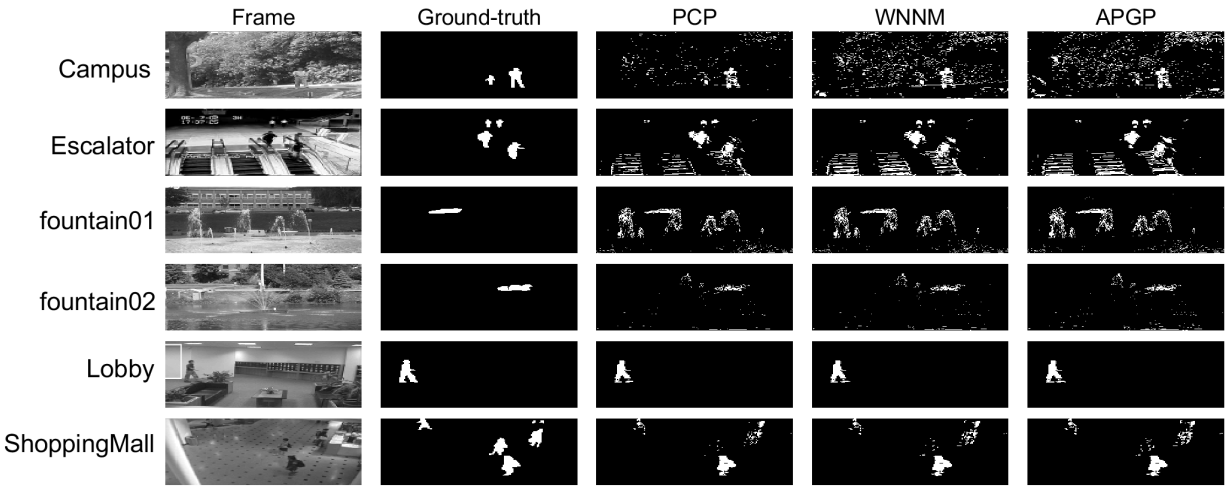


Figure 3. Visual results of APGP on the CDnet and I2R short-video dataset.

Building on these results, we further evaluate the robustness of the proposed algorithm in complex dynamic scenes using the BMC long-video dataset. This dataset contains nine real surveillance videos, denoted as Video_001 to Video_009, where noticeable background disturbances caused by abrupt illumination changes are present, as shown in Figure 4. APGP achieves the highest average F-measure of 0.79, outperforming WNNM and PCP, whose average F-measures are 0.76 and 0.70, respectively. It also requires the shortest average CPU time of 13.28s, compared with 16.85s for WNNM and 22.39s for PCP. The visual results in Figure 4 further show that APGP can effectively suppress background misclassification under abrupt illumination changes while preserving more complete foreground structures. These results indicate that the algorithm exhibits stability and robustness in long-video background subtraction under complex dynamic scenes.

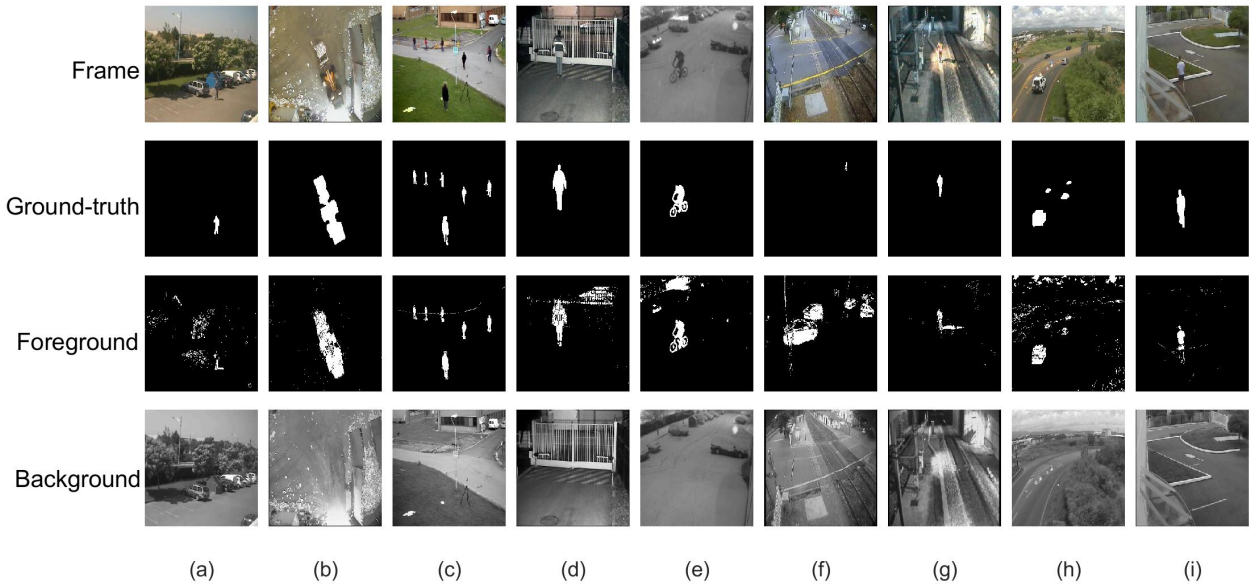


Figure 4. Visual results of APGP on the BMC long-video dataset. (a) Video_001. (b) Video_002. (c) Video_003. (d) Video_004. (e) Video_005. (f) Video_006. (g) Video_007. (h) Video_008. (i) Video_009.

Table 5. Numerical results of background subtraction on the BMC long-video dataset.

Video(a-i) \ ALGORITHM	F-measure			Time		
	PCP	WNNM	APGP	PCP	WNNM	APGP
Video_001	0.58	0.59	**0.60**	19.32	14.78	**12.52**
Video_002	0.79	0.76	**0.89**	26.61	23.40	**17.67**
Video_003	0.52	0.94	**0.95**	3.02	2.32	**2.30**
Video_004	0.75	0.79	**0.86**	18.48	13.06	**11.08**
Video_005	0.82	0.82	**0.83**	20.54	17.51	**13.83**
Video_006	0.54	0.56	**0.58**	27.25	19.81	**15.68**
Video_007	0.80	0.81	**0.82**	58.09	30.15	**22.28**
Video_008	0.79	**0.84**	**0.84**	12.94	13.89	**10.40**
Video_009	0.70	**0.73**	**0.73**	15.22	16.73	**13.76**
Average	0.70	0.76	**0.79**	22.39	16.85	**13.28**

In summary, the experimental results on the CDnet and I2R short-video dataset and the BMC long-video dataset show that APGP provides good separation accuracy, computational efficiency, and robustness in background subtraction, and can adapt well to different video scenarios and complex illumination disturbances.

5. Conclusion

In this paper, we proposed an APGP for the low-rank and mix-sparse matrix optimization problem involving both element-wise and group-wise sparsity. By incorporating partial SVD into the low-rank update, the proposed algorithm reduces the computational cost of large-scale SVD, while the mix-sparse component admits a closed-form update with low memory requirements. Moreover, the global convergence of the iterative sequence generated by APGP was established under mild conditions. Numerical experiments on synthetic data and background subtraction demonstrate that APGP shows competitive performance compared with existing algorithms.

Author contributions

Xian Zhang: Writing-original draft, methodology, data curation, visualization, conceptualization; Xin Wang: Writing-review & editing, methodology, supervision, resources; Lihua Wang: Writing-review & editing, methodology, formal analysis.

Use of Generative-AI tools declaration

The authors declare that they did not utilize any artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

This work is supported by the Scientific Research Project for Introduced Talents of Guiyang University (GYU-KY-[2025]-07).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. J. Bolte, S. Sabach, M. Teboulle, Proximal alternating linearized minimization for nonconvex and nonsmooth problems, *Math. Program.*, **146** (2014), 459–494. https://doi.org/10.1007/s10107-013-0701-9
2. D. Peng, N. Xiu, J. Yu, Global optimality conditions and fixed point continuation algorithm for non-Lipschitz ℓ_p regularized matrix minimization, *Sci. China Math.*, **61** (2018), 1139–1152. https://doi.org/10.1007/s11425-016-9107-y
3. Y. Hu, J. Lu, X. Yang, K. Zhang, Iterative mix thresholding algorithm with continuation technique for mix sparse optimization and application, *J. Glob. Optim.*, **91** (2025), 511–534. https://doi.org/10.1007/s10898-024-01441-w
4. Y. Jiao, B. Jin, X. Lv, Group sparse recovery via the $\ell_0(\ell_2)$ penalty: Theory and algorithm, *IEEE Trans. Signal Process.*, **65** (2017), 998–1012. https://doi.org/10.1109/TSP.2016.2630028

5. R. Fan, M. Jing, J. Shi, TVRPCA+: Low-rank and sparse decomposition based on spectral norm and structural sparsity-inducing norm. *Signal Process.*, **217** (2024), 109319. https://doi.org/10.1016/j.sigpro.2023.109319

6. D. Bertsimas, R. Cory-Wright, N. Johnson, Sparse plus low rank matrix decomposition: a discrete optimization approach, *J. Mach. Learn. Res.*, **24** (2023), 1–51. Available from: http://jmlr.org/papers/v24/21-1130.html.

7. E. Candès, Y. Plan, Matrix completion with noise, *Proc. IEEE*, **98** (2010), 925–936. https://doi.org/10.1109/JPROC.2009.2035722

8. S. Liao, S. Fu, Y. Li, H. Han, Image inpainting using non-convex low rank decomposition and multidirectional search, *Appl. Math. Comput.*, **452** (2023), 128048. https://doi.org/10.1016/j.amc.2023.128048

9. H. Yang, J. Lu, H. Zhang, Field of experts regularized nonlocal low rank matrix approximation for image denoising, *J. Comput. Appl. Math.*, **412** (2022), 114244. https://doi.org/10.1016/j.cam.2022.114244

10. L. Zhuang, J. Bioucas-Dias, Fast hyperspectral image denoising and inpainting based on low-rank and sparse representations, *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.*, **11** (2018), 730–742. https://doi.org/10.1109/JSTARS.2018.2796570

11. S. Ji, K. Sze, Z. Zhou, A. So, Y. Ye, Beyond convex relaxation: A polynomial-time non-convex optimization approach to network localization, *IEEE International Conference on Computer Communications (INFOCOM)*, 2013, 2499–2507. https://doi.org/10.1109/INFCOM.2013.6567056

12. Y. Li, X. Sun, Sensor network localization via riemannian conjugate gradient and rank reduction, *IEEE Trans. Signal Process.*, **72** (2024), 1910–1927. https://doi.org/10.1109/TSP.2024.3378378

13. H. Cai, C. Kundu, J. Liu, W. Yin, Deeply learned robust matrix completion for large-scale low-rank data recovery, *IEEE Trans. Pattern Anal. Mach. Intell.*, **48** (2026), 6541–6556. https://doi.org/10.1109/TPAMI.2026.3659041

14. Z. Lin, M. Chen, Y. Ma, The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices, arXiv:1009.5055v3.

15. E. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? *J. ACM.*, **58** (2011), 1–37. https://doi.org/10.1145/1970392.1970395

16. L. Yang, T. Pong, X. Chen, Alternating direction method of multipliers for a class of nonconvex and nonsmooth problems with applications to background/foreground extraction, *SIAM J. Imaging Sci.*, **10** (2017), 74–110. https://doi.org/10.1137/15M1027528

17. A. Ganesh, Z. Lin, J. Wright, L. Wu, M. Chen, Y. Ma, Fast algorithms for recovering a corrupted low-rank matrix, *IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2009, 213–216. https://doi.org/10.1109/CAMSAP.2009.5413299

18. X. Long, J. Nie, G. Li, Z. Peng, An accelerated preconditioned primal-dual gradient algorithm for nonconvex composite optimization problems with applications, *J. Comput. Math.*, **44** (2026), 1867–1889. https://doi.org/10.4208/jcm.2512-m2024-0107

19. X. Long, J. Nie, Z. Gou, X. Sun, G. Li, An accelerated preconditioned primal-dual gradient algorithm for structured nonconvex optimization problems, *Commun. Nonlinear Sci. Numer. Simulat.*, **153** (2026), 109480. https://doi.org/10.1016/j.cnsns.2025.109480
20. Y. Chen, J. Fan, C. Ma, Y. Yan, Bridging convex and nonconvex optimization in robust PCA: Noise, outliers, and missing data, *Ann. Statist.*, **49** (2021), 2948–2971. https://doi.org/10.1214/21-AOS2066
21. P. Netrapalli, U. Niranjan, S. Sanghavi, Non-convex robust PCA, arXiv:1410.7660v1.
22. Z. Wang, X. Li, H. So, Z. Liu, Robust PCA via non-convex half-quadratic regularization, *Signal Process.*, **204** (2023), 108816. https://doi.org/10.1016/j.sigpro.2022.108816
23. Y. Xie, S. Gu, Y. Liu, W. Zuo, Weighted Schatten p-norm minimization for image denoising and background subtraction, *IEEE Trans. Image Process.*, **25** (2016), 4842–4857. https://doi.org/10.1109/TIP.2016.2599290
24. K. Quach, C. Duong, K. Luu, T. Bui, Non-convex online robust PCA: enhance sparsity via ℓ_p-norm minimization, *Comput. Vis. Image. Underst.*, **158** (2017), 126–140. https://doi.org/10.1016/j.cviu.2017.03.002
25. Z. Xu, Y. Lu, J. Wu, Adaptive weighted robust principal component analysis, *IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2020, 19–24. https://doi.org/10.1109/ICIEA48937.2020.9248290
26. Z. Kang, C. Peng, Q. Cheng, Robust PCA via nonconvex rank approximation, *IEEE International Conference on Data Mining (ICDM)*, 2015, 211–220. https://doi.org/10.1109/ICDM.2015.15
27. S. Fang, Z. Xu, S. Wu, S. Xie, Efficient robust principal component analysis via block Krylov iteration and cur decomposition, *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, 1348–1357. https://doi.org/10.1109/CVPR52729.2023.00136
28. F. Wen, R. Ying, P. Liu, R. Qiu, Robust PCA using generalized nonconvex regularization, *IEEE Trans. Circuits Syst. Video. Technol.*, **30** (2019), 1497–1510. https://doi.org/10.1109/TCSVT.2019.2908833
29. Z. Xu, H. Xing, S. Fang, Double-weighted low-rank matrix recovery based on rank estimation, *IEEE/CVF International Conference on Computer Vision (ICCVW)*, 2021, 172–180. https://doi.org/10.1109/ICCVW54120.2021.00024
30. M. Indibi, S. Aviyente, Higher-order grouped outlier robust PCA for spatio-temporal anomaly detection, *IEEE Statistical Signal Processing Workshop (SSP)*, 2025, 176–180. https://doi.org/10.1109/SSP64130.2025.11073198
31. H. Likassa, D. Chen, K. Chen, Robust PCA with $L_{w,*}$ and $L_{2,1}$ norms: a novel method for low-quality retinal image enhancement, *J. Imaging*, **10** (2024), 151. https://doi.org/10.3390/jimaging10070151
32. H. Likassa, W. Fang, J. Leu, Robust image recovery via affine transformation and $L_{2,1}$ norm, *IEEE Access*, **7** (2019), 125011–125021. https://doi.org/10.1109/ACCESS.2019.2932470
33. H. Yang, Y. Zhao, J. Liu, Sparse regularization tensor robust PCA based on t-product and its application in cancer genomic data, *IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2020, 2131–2138. https://doi.org/10.1109/BIBM49941.2020.9313596

34. S. Ebadi, E. Izquierdo, Foreground segmentation with tree-structured sparse RPCA, *IEEE Trans. Pattern Anal. Mach. Intell.*, **40** (2017), 2273–2280. https://doi.org/10.1109/TPAMI.2017.2745573
35. L. Zhao, X. Hou, H. Yang, GRPCA21 for recovering a corrupted low-rank matrix, *Int. J. Mach. Learn. Cybern.*, **11** (2020), 1293–1305. https://doi.org/10.1007/s13042-019-01039-9
36. Y. Zhang, N. Zhang, D. Sun, K. Toh, An efficient Hessian based algorithm for solving large-scale sparse group Lasso problems, *Math. Program.*, **179** (2020), 223–263. https://doi.org/10.1007/s10107-018-1329-6
37. T. Thang, Bregman proximal point algorithms for quasi-equilibrium problems beyond prox-convexity, *J. Optim. Theory. Appl.*, **210** (2026), 19. https://doi.org/10.1007/s10957-026-03039-6
38. R. Rockafellar, R. Wets, *Variational Analysis*, Berlin: Springer, 1998. https://doi.org/10.1007/978-3-642-02431-3
39. Y. Chen, N. Xiu, D. Peng, Global solutions of non-Lipschitz S_2-S_p minimization over the positive semidefinite cone, *Optim. Lett.*, **8** (2014), 2053–2064. https://doi.org/10.1007/s11590-013-0701-y
40. Z. Xu, X. Chang, F. Xu, H. Zhang, $L_{1/2}$ regularization: a thresholding representation theory and a fast solver, *IEEE Trans. Neural Netw. Learning Sys.*, **23** (2012), 1013–1027. https://doi.org/10.1109/TNNLS.2012.2197412
41. C. Zeng, Proximal linearization methods for Schatten p-quasi-norm minimization, *Numer. Math.*, **153** (2023), 213–248. https://doi.org/10.1007/s00211-022-01335-7
42. H. Attouch, J. Bolte, On the convergence of the proximal algorithm for nonsmooth functions involving analytic features, *Math. Program. Ser. B*, **116** (2009), 5–16. https://doi.org/10.1007/s10107-007-0133-5
43. S. Ma, D. Goldfarb, L. Chen, Fixed point and Bregman iterative methods for matrix rank minimization, *Math. Program*, **128** (2011), 321–353. https://doi.org/10.1007/s10107-009-0306-5
44. S. Gu, Q. Xie, D. Meng, W. Zuo, Weighted nuclear norm minimization and its applications to low level vision, *Int. J. Comput. Vis.*, **121** (2017), 183–208. https://doi.org/10.1007/s11263-016-0930-5
45. N. Goyette, P. Jodoin, F. Porikli, J. Konrad, P. Ishwar, Changedetection.net: A new change detection benchmark dataset, *In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012, 1–8. https://doi.org/10.1109/CVPRW.2012.6238919
46. L. Li, W. Huang, I. Gu, Q. Tian, Statistical modeling of complex backgrounds for foreground object detection, *IEEE Trans. Image Process*, **13** (2004), 1459–1472. https://doi.org/10.1109/TIP.2004.836169
47. A. Vacavant, T. Chateau, A. Wilhelm, L. Lequièvre, A benchmark dataset for outdoor foreground/background extraction, *Computer Vision - ACCV 2012 Workshops*, 2013, 291–300. https://doi.org/10.1007/978-3-642-37410-4_25
48. B. Alawode, S. Javed, Unsupervised background subtraction using generator-discriminator learning, *IEEE T. Circ. Syst. Vid.*, **36** (2026), 986–1002. https://doi.org/10.1109/tcsvt.2025.3598078



© 2026 the Author(s), licensee AIMS Press. This is an open access article distributed under the terms of the Creative Commons Attribution License (https://creativecommons.org/licenses/by/4.0)