algorithms

MDPI

Article

Efficient Matrix Reconstruction via Randomized Linear-Time Singular Value Decomposition and Adaptive Thresholding

Na Zhou [1,*] and Wentao Wang [2]

[1] School of Computer Science and Artificial Intelligence, Zijin College, Nanjing University of Science and Technology, Nanjing 210023, China
[2] ZTE Corporation, Nanjing 210012, China
* Correspondence: 13951827554@163.com

Abstract

Matrix reconstruction is a fundamental problem in high-dimensional data analysis and video background modeling. Robust principal component analysis (RPCA) decomposes an observed data matrix into a low-rank background component and a sparse component representing foreground objects or corruption. However, solvers based on the augmented Lagrangian method (ALM) and its linearized variant (LALM) repeatedly invoke singular value decomposition (SVD), which is computationally expensive for video-scale matrices. This study develops a noise-aware randomized LALM framework that combines randomized linear-time approximate SVD, adaptive regularization based on normalized data dispersion, and dynamic rank estimation. The framework is evaluated using synthetic SVD approximation, singular value thresholding (SVT) preservation, and background reconstruction on 14 Scene Background Initialization (SBI) sequences, with ten random-seed runs per sequence–method pair. Additional statistical, parameter-sensitivity, cross-dataset, and ablation analyses are conducted to assess robustness and the contributions of the adaptive components. The randomized approximate SVD achieves an average 40.52× speedup over full SVD. In the SVT experiments, sampling 80 columns provides a practical efficiency–accuracy balance. On SBI, the proposed method achieves an average peak signal-to-noise ratio (PSNR) of 26.61 dB, signal-to-noise ratio (SNR) of 21.54 dB, and elapsed time of 57.90 s while obtaining the lowest average gray-level error (AGE), percentage of error pixels (pEPs), and percentage of clustered error pixels (pCEPs) among the tested LALM variants. The additional analyses indicate that the performance gains are scene dependent and that dynamic rank estimation and noise-aware regularization provide complementary benefits. Overall, the proposed framework offers a practical quality–efficiency trade-off for RPCA-based video background reconstruction.

Keywords: adaptive regularization; matrix reconstruction; randomized SVD; robust PCA; video background modeling

Check for updates

Academic Editor: Roberto Montemanni

Received: 14 October 2026
Revised: 13 November 2026
Accepted: 13 November 2026
Published: 17 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Matrix reconstruction has become a fundamental technique for image processing, signal recovery, and video background modeling. In video applications, an observed matrix can often be decomposed into a low-rank part that represents the static or slowly changing background and a sparse part that represents moving foreground objects, shadows, or noise. This model naturally leads to robust principal component analysis (RPCA), which replaces a rank-minimization problem with nuclear-norm minimization and an ℓ_1 sparsity penalty [1,2].

Although RPCA is conceptually simple and interpretable, practical solvers based on the augmented Lagrangian method (ALM) and linearized ALM (LALM) repeatedly invoke singular value decomposition (SVD) or partial SVD within singular value thresholding (SVT) [2]. This step often dominates the computational cost when matrices are large or many iterations are required. Consequently, exact SVD can limit deployment in real-time or resource-constrained applications such as surveillance video analysis and edge computing.

This study revisits the LALM framework and focuses on reducing its low-rank update cost without replacing the model with a heavy learning-based architecture. The central idea is to use a randomized column-sampling SVD approximation in the SVT step and to calculate only the singular directions that are relevant to the current threshold. Randomized numerical linear algebra provides the methodological basis for such approximate matrix decompositions [3–6]. Large data matrices often exhibit approximate low-rank structures, which further motivate low-rank reconstruction models [7]. Building on this efficient update, we introduce a noise-aware strategy based on normalized data dispersion to improve the trade-off between background preservation and sparse residual separation.

Related randomized and SVD-based low-rank techniques have been applied to sparse approximation, tensor completion, missing-data reconstruction, nonconvex low-rank learning, robust matrix completion, randomized tensor decomposition, and streaming SVD [8–14]. These studies demonstrate the broad utility of randomized low-rank approximation, whereas the present work focuses on integrating an efficient SVD approximation with noise-aware regularization in an RPCA-based background-reconstruction solver.

The main contributions of this work are summarized as follows.

First, building on the theoretical foundation of randomized and linear-time matrix approximation, we develop an improved linear-time approximate SVD scheme for the repeated low-rank updates required in matrix reconstruction. Instead of performing a full SVD at each iteration, the method constructs a compact matrix through randomized column sampling and recovers the dominant singular components from the resulting low-dimensional eigendecomposition. The formulation is further coupled with the singular value thresholding process so that only the singular components relevant to the current reconstruction step need to be computed.

Second, we employ a threshold-driven rank estimation mechanism in which the requested number of singular components for the subsequent iteration is determined from the number of singular values retained by the current threshold. This mechanism originates from the relationship between the required singular value number and the iterative threshold derived in our original formulation and progressively reduces the computational burden as the effective rank decreases during the reconstruction process.

Third, we introduce a noise-aware regularization strategy based on normalized data dispersion to further improve the balance between background preservation and sparse foreground separation under varying scene conditions.

Finally, we provide a systematic experimental evaluation covering standalone SVD approximation, SVT-threshold preservation, repeated SBI background reconstruction, parameter-sensitivity, cross-dataset validation, and ablation analysis, thereby examining both computational efficiency and reconstruction behavior.

The remainder of this paper is organized as follows. Section 2 presents the RPCA formulation, the randomized SVD approximation, dynamic rank estimation, and the noise-aware regularization strategy. Section 3 reports the experimental results. Section 4 discusses limitations and practical implications, and Section 5 concludes the paper.

2. Materials and Methods

The proposed framework builds on established RPCA, LALM, singular value thresholding, and randomized low-rank approximation theory. These general mathematical and optimization foundations are not claimed as new in this work. Building on them, we develop an improved linear-time approximate SVD formulation for the repeated low-rank updates arising in matrix reconstruction, together with a threshold-driven mechanism for determining the number of singular components required in subsequent iterations. The present work further incorporates a noise-aware adaptive regularization strategy into this reconstruction framework. Thus, the methodological contribution lies in the improved approximate-SVD formulation for iterative matrix reconstruction, its coupling with threshold-driven rank estimation, and its integration with noise-aware regularization.

2.1. Robust Principal Component Analysis

Given an observed matrix $D \in \mathbb{R}^{m\times n}$, RPCA assumes that D can be decomposed into a low-rank matrix L and a sparse matrix S, i.e., $D = L + S$. The convex RPCA formulation is written as [1]

$$\min_{L,S} \|L\|_* + \lambda \|S\|_1, \text{ s.t. } D = L + S \tag{1}$$

where $\|\cdot\|_*$ denotes the nuclear norm, $\|\cdot\|_1$ denotes the element-wise ℓ_1 norm, and $\lambda > 0$ balances the low-rank and sparse terms. In video background modeling, L corresponds to the estimated background and S corresponds to foreground motion or corruption.

2.2. Augmented Lagrangian Method and Singular Value Thresholding

The augmented Lagrangian method solves (1) by alternating among a low-rank update, a sparse update, and a multiplier update [2]. The low-rank update is commonly implemented by singular value thresholding:

$$L_{k+1} = US_\tau(\sum)V^T \tag{2}$$

where $U\sum V^T$ denotes the SVD of the current auxiliary matrix and $S_\tau(\cdot)$ shrinks each singular value by the threshold τ. Consequently, only singular values larger than τ and their corresponding singular vectors contribute to the next low-rank iterate. This observation motivates the use of partial and approximate SVD.

2.3. Computational Bottleneck

For a dense matrix $m \times n$, exact SVD becomes costly when m and n are large. Even partial SVD can remain expensive if the requested rank is overestimated. In LALM-based RPCA, this cost is incurred at every iteration. Reducing the number of columns and singular directions processed in each iteration is therefore a direct way to reduce runtime.

2.4. Randomized Linear-Time Approximate SVD

The formulation below is used for the repeated approximate-SVD computations required in the matrix-reconstruction procedure. Its randomized column-sampling principle follows established randomized numerical linear algebra [3–6], while the present formulation adapts this principle to the threshold-dependent low-rank updates arising in the iterative reconstruction process.

For an input matrix $A \in \mathbb{R}^{m\times n}$, c_s columns are sampled according to the sampling probabilities $\{p_i\}_{i=1}^n$. At the t-th sampling step, an index $i_t \in \{1, \dots, n\}$ is drawn according to $\Pr[i_t = i] = p_i, i = 1, \dots, n$, and the corresponding rescaled sampled column c_t is constructed as follows:

$$c_t = \frac{A :, i_t}{\sqrt{c_s p_{i_t}}}, t = 1, \dots, c_s. \tag{3}$$

The sampled columns are then assembled to form the sketch matrix $C \in \mathbb{R}^{m \times c_s}$, consisting of the rescaled columns c_t. The sampling probabilities satisfy $p_i \geq 0$, $\sum_{i=1}^{n} p_i = 1$. Uniform sampling is used in the reported experiments, i.e., $p_i = \frac{1}{n}, i = 1, \ldots, n$.

The approximation is intended for matrices with an approximate low-rank structure whose dominant subspace information is not strongly concentrated in only a few columns. For the uniform-sampling implementation considered here, we assume $k_s \leq c_s \ll n$ with the sketch size chosen sufficiently larger than the target rank. Under these standard conditions, randomized column-sampling methods can preserve the dominant low-rank subspace with high probability [3–6]. The corresponding probabilistic approximation guarantee is stated after Algorithm 1 in Equation (10).

Algorithm 1. Randomized Linear-Time Approximate SVD.

Input	$A \in \mathbb{R}^{m \times n}, c_s, k_s, \{p_i\}_{i=1}^{n}$.
1	for $t = 1, \ldots, c_s$
2	Sample $i_t \in \{1, \ldots, n\}$ according to $\Pr[i_t = i] = p_i$, and construct $c_t = \frac{A_{:,i_t}}{\sqrt{c_s p_{i_t}}}$
3	**end for**
4	Form $C = [c_1, \ldots, c_{c_s}] \in \mathbb{R}^{m \times c_s}$.
5	Compute the eigendecomposition of $C^T C = \sum_{j=1}^{c_s} \sigma_j^2(C) y_j y_j^T$
6	For the first k_s components, compute $h_j = \frac{C y_j}{\sigma_j(C)}$, $j = 1, \ldots, k_s$.
Output	$H_{k_s} = [h_1, \ldots, h_{k_s}]$ and $\sum_{k_s} = diag(\sigma_1(C), \ldots, \sigma_{k_s}(C))$.

Uniform sampling is adopted because of its low computational overhead and straightforward integration into the iterative reconstruction procedure. However, it may become less reliable when column importance is highly nonuniform because informative columns can be undersampled. Importance-based strategies, such as leverage-score or norm-based sampling, may improve subspace preservation but require additional probability estimation or preprocessing. Adaptive sampling can further respond to an iteration-dependent matrix structure, although repeated probability updates introduce additional computational overhead. Uniform sampling is therefore used here as a computationally lightweight choice, while importance-aware and adaptive sampling remain possible extensions.

From a computational perspective, full SVD of a dense $m \times n$ matrix has a computational cost of approximately $O(mn\min(m, n))$. For the sampled matrix $C \in \mathbb{R}^{m \times c_s}$, forming the compact Gram matrix $C^T C$ requires approximately $O(mc_s^2)$, while the eigendecomposition of the $c_s \times c_s$ Gram matrix requires approximately $O(c_s^3)$. Recovering the first k_s approximate left singular vectors requires approximately $O(mc_s k_s)$, and forming the corresponding right singular factors through $A^T H_{k_s}$ requires approximately $O(mnk_s)$. Therefore, the approximate-SVD factor computation has an overall cost of approximately $O(mc_s^2 + c_s^3 + mc_s k_s + mnk_s)$. If the full rank-$k_s$ low-rank matrix is explicitly reconstructed, an additional $O(mnk_s)$ multiplication is required, which does not change the asymptotic order. When c_s is much smaller than n and k_s is much smaller than $\min(m, n)$, the expensive dense decomposition is replaced by an eigendecomposition of the much smaller $c_s \times c_s$ Gram matrix.

Standard randomized SVD generally requires multiplications involving the full matrix, whereas Lanczos-type partial SVD relies on iterative matrix–vector products whose total cost depends on the requested rank and convergence behavior. In contrast, after column sampling, the principal eigendecomposition in the proposed method is performed on the compact $c_s \times c_s$ Gram matrix $C^T C$. This difference is consistent with the increasing speedup observed with matrix size in Section 3.2.

For the synthetic SVD benchmark, the sampled-column number is selected as $c_s = \max(40, 4k_s)$, whereas the SBI experiments use $c_s = 80$ with an initial target rank of

$k_s = 20$. These settings are evaluated empirically in Section 3 to quantify the corresponding efficiency–accuracy trade-off.

2.5. Dynamic Rank Estimation

In the present implementation, the requested rank is adapted according to the singular values retained after the current singular value thresholding (SVT) operation. Let $k_s^{(k)}$ denote the number of singular components requested at iteration k. The requested rank for the next iteration is determined by the number of thresholded singular values that remain nonzero, subject to a minimum rank of one. The corresponding update rule is given in Equation (12).

As the LALM iterations proceed, the thresholded spectrum may become increasingly compact, and the requested rank may therefore decrease. If only one singular value remains above the current threshold, the subsequent iteration requests one dominant singular component; because the implementation enforces a minimum rank of one, the rank cannot decrease further. Thus, the rank-one behavior observed in the SBI experiments results from the threshold-driven update rule rather than from an independently imposed rank-one model. The implications of this behavior for dynamic-background sequences are discussed in Section 4.

2.6. Noise-Aware Adaptive Regularization

The regularization parameter λ controls the trade-off between the low-rank background and sparse residual components. The classical RPCA choice uses $\lambda_0 = \frac{1}{\sqrt{\max\{m,n\}}}$. However, a fixed value may be suboptimal when input sequences exhibit different levels of intensity variability. We therefore define a noise-aware regularization parameter based on the normalized data dispersion of the input sequence.

Let $D \in \mathbb{R}^{m\times n}$ denote the input video data matrix, where each column represents one vectorized video frame, m is the number of spatial pixels in each frame, and n is the number of frames. The spatial mean intensity of the j-th frame is first calculated as

$$\mu_j = \frac{1}{m}\sum_{i=1}^{m} D_{ij}, j = 1, \ldots, n. \tag{4}$$

Let $\mu_{\min} = \min_j \mu_j$ and $\mu_{\max} = \max_j \mu_j$. The frame-wise mean intensities are normalized using min–max normalization:

$$\tilde{\mu}_j = \begin{cases} \frac{\mu_j - \mu_{\min}}{\mu_{\max} - \mu_{\min}}, \mu_{\max} - \mu_{\min} > \varepsilon, \\ 0, \mu_{\max} - \mu_{\min} \le \varepsilon, \end{cases} \quad \varepsilon = 10^{-15}. \tag{5}$$

The mean of the normalized frame-wise intensities is $\overline{\mu} = \frac{1}{n}\sum_{j=1}^{n} \tilde{\mu}_j$. The normalized variance-based dispersion score is then defined as

$$d(D) = \frac{1}{n}\sum_{j=1}^{n} (\tilde{\mu}_j - \overline{\mu})^2 \tag{6}$$

The corresponding dispersion factor is $c_1(D) = \sqrt{d(D)}$. To prevent extremely small or large values from excessively affecting the regularization parameter, the dispersion factor is bounded to the interval $[0.3, 1.0]$:

$$\hat{c}_1(D) = \min\{1, \max\{0.3, c_1(D)\}\} \tag{7}$$

The proposed noise-aware regularization parameter is then defined as

$$\lambda_{NA}(D) = \frac{\lambda_0}{\sqrt{\hat{c}_1(D)}} = \frac{1}{\sqrt{\hat{c}_1(D)\max(m,n)}}. \tag{8}$$

In this calculation, the spatial dimension is first aggregated through frame-wise averaging, while temporal variability is characterized by the population variance of the normalized frame-wise mean intensities across the sequence. Therefore, $d(D)$ should be interpreted as a sequence-level intensity-dispersion indicator rather than a pixel-wise spatiotemporal variance or a direct estimate of physical noise variance.

The bounds in Equation (7) are implementation-level stabilization settings. The lower bound of 0.30 prevents an excessively small dispersion factor from producing an overly large λ_{NA}. Under the current min–max normalization and variance-based definition, the normalized frame-wise means satisfy $0 \leq \tilde{\mu}_j \leq 1$. Accordingly, the population variance satisfies $d(D) \leq \frac{1}{4}$ and therefore $c_1(D) = \sqrt{d(D)} \leq 0.5$. Consequently, the upper bound of 1.0 is retained as an implementation safeguard but is not expected to be active under the reported min–max-normalized variance setting.

Once the dispersion factor $c_1(D) = \sqrt{d(D)}$ exceeds the lower bound of 0.30, an increase in $d(D)$ decreases the noise-aware regularization parameter $\lambda_{NA}(D)$, thereby reducing the penalty on the sparse component and allowing a larger portion of the observed variation to be represented by the sparse residual. When the unbounded dispersion factor $c_1(D)$ falls below 0.30, the clipping operation sets $\hat{c}_1(D) = 0.3$, thereby preventing λ_{NA} from increasing further. Because $d(D)$ can also respond to structured background motion, foreground activity, and illumination changes, this behavior should not be interpreted as a monotonic response to physical noise alone.

2.7. Algorithmic Workflow

The complete workflow is summarized as follows. The algorithm initializes the low-rank matrix, sparse matrix, Lagrange multiplier, penalty parameter, sampled-column count c_s, and initial target rank k_s. At each iteration, it forms the auxiliary matrix for the low-rank update, computes the randomized approximate SVD, applies SVT, updates the sparse residual by soft thresholding, updates the multiplier and penalty parameter, and adjusts the requested rank for the next iteration. The procedure terminates when the relative residual falls below the prescribed convergence tolerance or when the maximum number of iterations is reached.

Algorithm 1 summarizes the randomized linear-time approximate SVD module used in the low-rank update. The sampled matrix is constructed from c_s rescaled columns, after which the dominant singular components are recovered from the eigendecomposition of the compact Gram matrix $C^T C$.

In Algorithm 1, c_s denotes the total number of sampled columns, while c_t denotes the t-th rescaled sampled column. The index t is used for column sampling, whereas j indexes the eigenpairs of $C^T C$ and the corresponding approximate singular components:

$$A \approx A_{k_s} = H_{k_s} \sum\nolimits_{k_s} \left(A^T H_{k_s} \sum\nolimits_{k_s}^{-1} \right)^T \tag{9}$$

Randomized numerical linear algebra theory provides probabilistic approximation guarantees for sketch-based low-rank approximation [3–6]. Under appropriate randomized column-sampling assumptions, bounds of the following form hold with high probability for spectral and Frobenius norms:

$$\| A - A_{k_s} \|^2 \leq \min_{rank(D) \leq k_s} \| A - D \|_\zeta^2 + poly(k_s, \frac{1}{c_s}) \| A \|_F^2 \tag{10}$$

In the ALM update, only singular values exceeding the threshold τ and their corresponding singular vectors are required:

$$A_\varepsilon := \sum_{\sigma_j > \tau} \sigma_j u_j v_j^T \tag{11}$$

In the proposed framework, k_s denotes the number of singular components requested from Algorithm 1. Overestimating k_s increases computation, whereas underestimating it may omit singular values above τ and degrade the low-rank update. Therefore, k_s is adapted to the current threshold.

As the ALM iterations proceed, the retained rank of the auxiliary matrix often decreases. We therefore update k_s using the number of singular values retained at the previous iteration. This strategy prevents k_s from falling below the number of components needed for the current update while avoiding unnecessary singular-component computations.

At iteration k, the requested rank for the approximate SVD of the current auxiliary matrix is updated as follows:

$$k_s = \left\| \mathcal{S}_{\mu_{k-1}^{-1}}[S] \right\|_0 \tag{12}$$

Here, the thresholding operation retains the singular components that remain above the current SVT threshold and $\|\cdot\|_0$ denotes the number of retained singular components. In the implementation, the requested rank is lower-bounded by one to avoid an empty low-rank update.

3. Results

3.1. Experimental Setup

The core experiments are organized as follows. The first evaluates the standalone SVD approximation on controlled synthetic low-rank matrices. The second assesses whether the approximate singular values and subspaces preserve SVT decisions. The third evaluates video background reconstruction on 14 Scene Background Initialization (SBI) sequences [15]. For the SBI experiment, each sequence–method pair is repeated ten times with different random seeds, yielding 420 runs in total.

All SBI input frames and ground-truth background images were converted to grayscale before matrix construction, reconstruction, and quantitative evaluation.

The synthetic SVD benchmark uses matrix sizes 500 × 300, 1000 × 600, and 2000 × 1000; target ranks 10 and 20; exponential spectra; incoherent singular vectors; condition number 100; and noise levels 0 and 0.001. The compared SVD methods are full SVD, Lanczos svds, standard randomized SVD, and the proposed linear-time approximate SVD. The SVT experiment varies the sampled-column number over $c_s \in \{40, 80, 120\}$ and evaluates relative SVT error, retained rank, false drops, false keeps, and rank-decision accuracy. For the main SBI experiments, $c_s = 80$ and the initial target rank is $k_s^{(0)} = 20$. The SBI experiment compares the fixed-rank LALM, the dynamic-rank LALM, and the proposed method.

Each SBI sequence–method combination is independently repeated ten times using random seeds 1–10, yielding 420 runs for the main SBI comparison. For each method, the performance metrics are first averaged across the 14 SBI sequences within each run, and the resulting ten run-wise averages are used to calculate the mean, standard deviation, and 95% confidence interval. The confidence interval is computed as $\bar{x} \pm t_{0.975,9} \frac{s}{\sqrt{10}}$, $t_{0.975,9} = 2.262$, where $\bar{x}$ and s denote the sample mean and sample standard deviation, respectively.

To assess whether the observed differences are consistent across scenes, the ten-run mean is also calculated separately for each SBI sequence and method. The three methods are compared across the 14 matched sequences using the Friedman test. When the omnibus test is significant, pairwise comparisons between the proposed method and the two baseline methods are performed using the Wilcoxon signed-rank test with Holm correction. A corrected p-value below 0.05 is considered statistically significant.

All experiments are conducted on a Windows 11 Home 64-bit system equipped with an Intel Core i5-10210U CPU at 1.60 GHz (Intel Corporation, Santa Clara, CA, USA) and 16 GB of RAM. The implementation uses Python 3.8.0, NumPy 1.22.0, SciPy 1.10.1, and

Pillow 10.3.0. The maximum number of iterations is 300, and convergence is declared when the relative residual falls below 10^{-6}. The same random-seed sequence is used across the competing randomized methods, and no sequence-specific parameter tuning is applied in the main SBI experiments.

3.2. Synthetic SVD Approximation Benchmark

Table 1 summarizes the average SVD benchmark results. The randomized linear-time SVD used in the framework is substantially faster than the baselines, achieving an average speedup of 40.52× over full SVD under the tested settings. It is also faster than standard randomized SVD and Lanczos svds. Although the reconstruction error remains small, the relative error in the singular values is non-negligible; therefore, the approximation is further evaluated through SVT rather than being assessed solely by factor reconstruction.

Table 1. Average singular value decomposition (SVD) approximation benchmark.

Method	Time (s)	Speedup	RkErr	SVErr
Full SVD	0.3827	1.000	0	0
Lanczos svds	0.0292	12.24	2.73×10^{-15}	7.78×10^{-16}
Standard randomized SVD	0.0200	14.39	1.28×10^{-12}	8.35×10^{-16}
Proposed	7.07×10^{-3}	40.52	2.87×10^{-4}	0.0835

The observed speedup increases with matrix size, supporting the use of column sampling for large video matrices.

Figure 1 shows the scaling trend of the randomized linear-time SVD. The speedup increases with matrix size, supporting the use of column sampling for large video matrices.

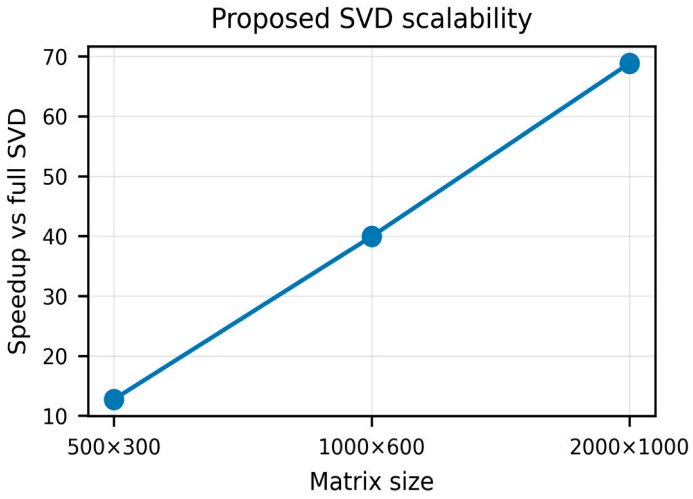


Figure 1. Average speedup of the proposed singular value decomposition (SVD) over full SVD for different matrix sizes.

3.3. SVT Threshold-Preservation Experiment

Because RPCA uses SVT rather than SVD in isolation, Table 2 reports threshold-preservation performance. Increasing c_s improves SVT accuracy and rank-decision accuracy but reduces speedup. The setting $c_s = 80$ provides a practical balance: it retains a speedup above 12× in both tested ranks while keeping rank-decision accuracy above 98% on average. As shown in Figure 2, the relative SVT error decreases as the number of sampled columns increases.

Table 2. SVT threshold-preservation results.

Rank	c_s	Speedup	Error (%)	Accuracy (%)	SVErr (%)
10	40	19.94	5.93	97.67	12.18
10	80	14.26	3.80	98.33	6.86
10	120	10.55	3.21	98.67	5.42
20	40	17.28	10.32	95.83	8.56
20	80	12.08	6.58	98.17	6.84
20	120	10.10	5.29	98.67	6.01

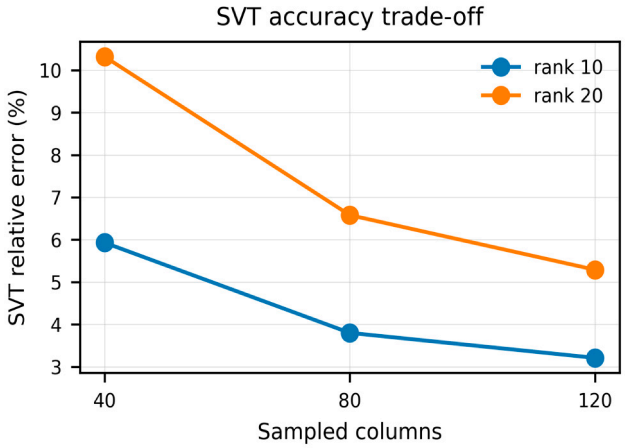


Figure 2. SVT relative error decreases as the number of sampled columns increases.

Although the approximate SVD exhibits non-negligible relative singular value errors, these errors do not translate directly into comparable errors in SVT decisions. With $c_s = 80$, the rank-decision accuracy is 98.33% for target rank 10 and 98.17% for target rank 20, while the corresponding singular value errors are 6.86% and 6.84%, respectively. Approximation errors are most consequential for singular values close to the SVT threshold, where underestimation or overestimation may change whether a component is retained. The implications of such threshold-crossing errors for dynamic-rank updates are discussed in Section 4.

3.4. Ten-Run SBI Background Reconstruction

Table 3 reports the ten-run SBI averages over 14 sequences for elapsed time, frames per second (FPS), peak signal-to-noise ratio (PSNR), signal-to-noise ratio (SNR), average gray-level error (AGE), percentage of error pixels (pEPs), percentage of clustered error pixels (pCEPs), and retained rank. Among the three LALM variants, the proposed method achieves the highest average PSNR and SNR and the lowest average AGE, pEPs, pCEPs, and elapsed time. Relative to the dynamic-rank LALM, the proposed method increases the average PSNR from 26.38 dB to 26.61 dB and decreases the average AGE from 14.30 to 13.63. Relative to the fixed-rank LALM, it reduces mean elapsed time from 81.55 s to 57.90 s while increasing the average PSNR from 26.23 dB to 26.61 dB. The dynamic-rank LALM has a slightly higher average FPS (9.34 versus 9.19). Figure 3 compares the average PSNR of the three LALM variants, where the proposed method achieves the highest value.

Table 3. Ten-run SBI results reported as mean ± standard deviation.

Metric	Fixed-Rank	Dynamic-Rank	Proposed
Time (s)	81.55 ± 65.36	67.95 ± 54.38	57.90 ± 19.63
FPS	7.95 ± 0.34	9.34 ± 0.19	9.19 ± 0.11
PSNR (dB)	26.23 ± 0.04	26.38 ± 0.10	26.61 ± 0.11
SNR (dB)	21.16 ± 0.04	21.31 ± 0.10	21.54 ± 0.11
AGE	14.63 ± 0.04	14.30 ± 0.14	13.63 ± 0.18
pEPs	0.2212 ± 0.0016	0.1968 ± 0.0034	0.1887 ± 0.0031
pCEPs	0.1805 ± 0.0015	0.1599 ± 0.0032	0.1518 ± 0.0028
Retained rank	18.82 ± 0.04	1.00 ± 0.00	1.00 ± 0.00

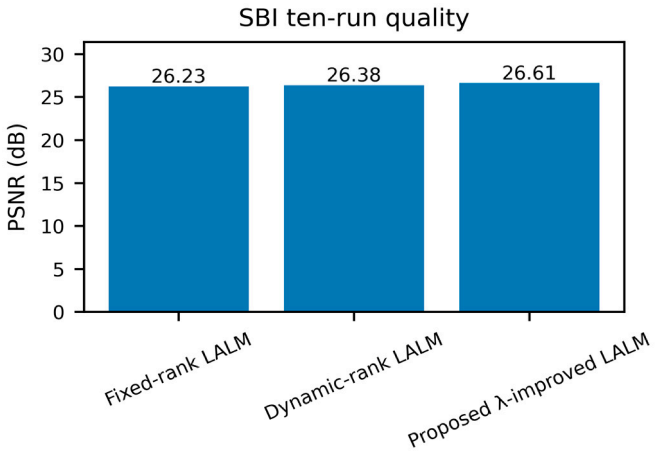


Figure 3. Ten-run SBI average PSNR of the three LALM variants.

Across the ten randomized runs, the proposed method achieves a mean PSNR of 26.61 ± 0.11 dB, with a 95% confidence interval of [26.53, 26.69] dB. The Friedman test across the 14 matched SBI sequences indicates significant overall differences among the three methods for elapsed time, FPS, and AGE, whereas the differences in PSNR, SNR, pEPs, and pCEPs do not reach the 0.05 significance level. Post hoc Wilcoxon signed-rank tests with Holm correction confirm significant improvements of the proposed method over the fixed-rank LALM in elapsed time and FPS, while the corresponding pairwise AGE differences are not significant after correction. Therefore, the reconstruction-quality improvements should be interpreted as average performance gains rather than universal statistically significant superiority across all sequences.

Table 4 shows a per-sequence comparison based on the average PSNR over ten runs. The proposed method achieves the highest PSNR on 5 of the 14 sequences, fixed-rank LALM on 7 sequences, and dynamic-rank LALM on 2 sequences. Thus, no single variant dominates every sequence. Nevertheless, the proposed noise-aware strategy yields the highest overall average PSNR together with the lowest average SBI error measures.

To complement the quantitative evaluation, Figure 4 compares the ground-truth backgrounds with the reconstructions obtained by the fixed-rank LALM, the dynamic-rank LALM, and the proposed method for Board, CaVignal, and Toscana. All images are displayed in grayscale, consistent with the experimental preprocessing. The examples reflect the scene-dependent behavior observed in Table 4: the fixed-rank LALM remains competitive on Board and CaVignal, whereas the proposed method achieves the highest average PSNR on Toscana. The qualitative results are therefore interpreted together with, rather than independently of, the quantitative metrics.

Table 4. Per-sequence ten-run SBI comparison based on PSNR (dB).

Sequence	Proposed	Fixed	Dynamic
Board	15.49	16.79	15.34
CAVIAR1	33.27	29.03	33.28
CAVIAR2	46.92	45.99	46.77
CaVignal	16.93	20.48	16.92
Candela	26.08	27.73	26.06
Foliage	15.89	15.88	15.53
HallMon	28.88	29.66	28.80
HighwayI	38.36	37.99	38.44
HighwayII	34.84	34.91	34.63
HumanBody2	33.84	28.73	33.76
IBMtest2	36.00	34.90	35.54
Ppl&Fol	13.54	13.67	13.32
Snellen	12.78	12.86	12.58
Toscana	19.73	18.60	18.36

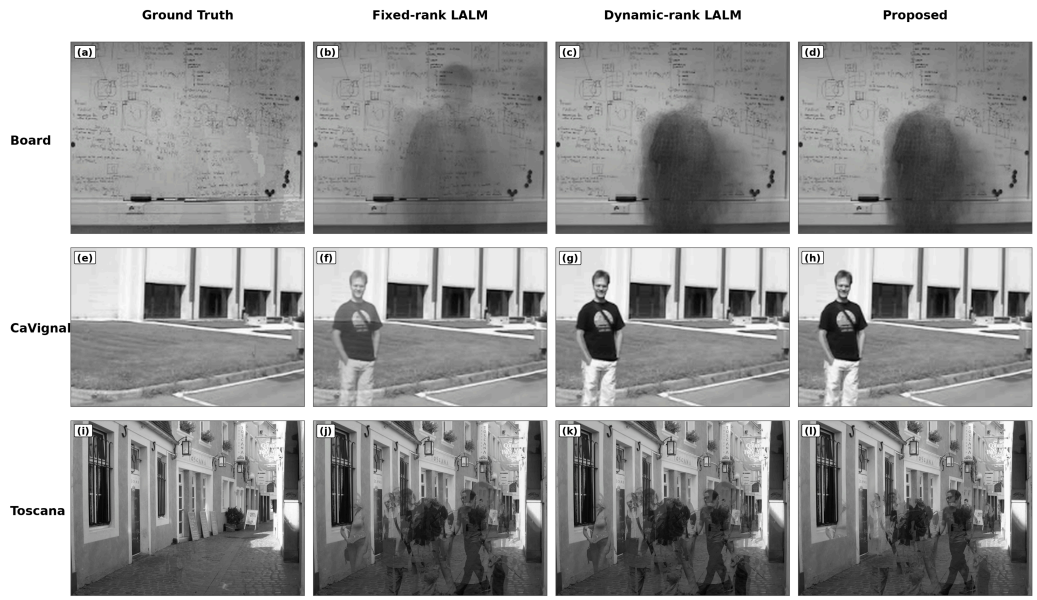


Figure 4. Qualitative background-reconstruction comparison on Board, CaVignal, and Toscana: (**a**) Board ground truth; (**b**) Board reconstructed by fixed-rank LALM; (**c**) Board reconstructed by dynamic-rank LALM; (**d**) Board reconstructed by the proposed method; (**e**) CaVignal ground truth; (**f**) CaVignal reconstructed by fixed-rank LALM; (**g**) CaVignal reconstructed by dynamic-rank LALM; (**h**) CaVignal reconstructed by the proposed method; (**i**) Toscana ground truth; (**j**) Toscana reconstructed by fixed-rank LALM; (**k**) Toscana reconstructed by dynamic-rank LALM; (**l**) Toscana reconstructed by the proposed method. All images are displayed in grayscale, consistent with the experimental preprocessing.

3.5. Parameter-Sensitivity Analysis

Table 5 summarizes the sensitivity of the proposed method to the initial target rank and convergence tolerance on three representative SBI sequences—CaVignal, Foliage, and HighwayI—with three random-seed runs for each configuration. Sensitivity to the sampled-column number c_s has already been evaluated through the SVT-preservation experiment in Section 3.3.

Table 5. Sensitivity of the proposed method to the initial target rank and convergence tolerance on three representative SBI sequences.

Parameter	Setting	PSNR (dB)	AGE	Time (s)	FPS	Iterations
Initial rank $k_s^{(0)}$	10	23.6788	16.0126	35.19	13.70	33.67
	20 (default)	23.6788	16.0126	36.52	13.64	33.67
	30	23.6788	16.0126	36.32	13.79	33.67
Tolerance ε_{tol}	10^{-5}	23.6782	16.0126	30.46	15.75	29.11
	10^{-6} (default)	23.6788	16.0126	34.68	13.90	33.67
	10^{-7}	23.6789	16.0125	39.48	11.93	39.33

Note: Values are averaged over CaVignal, Foliage, and HighwayI with three random-seed runs per sequence. All tested configurations converged successfully and yielded a final retained rank of one.

Varying the initial target rank among 10, 20, and 30 produces essentially unchanged PSNR, AGE, and iteration counts, and all three settings converge to a final retained rank of one. Varying the convergence tolerance mainly affects computational cost: a looser tolerance reduces the number of iterations and runtime with negligible change in reconstruction quality, whereas a tighter tolerance increases computation with only marginal PSNR improvement. These results support the default settings used in the main SBI experiments as practical choices under the tested conditions rather than universally optimal parameters.

3.6. Cross-Dataset Validation on CDnet 2014 [16]

To assess cross-dataset generalization beyond SBI, the fixed-rank LALM, dynamic-rank LALM, and proposed method were additionally evaluated on four representative CDnet 2014 sequences [16]: highway, canoe, peopleInShade, and parking. These sequences cover conventional, dynamic-background, illumination/shadow, and intermittent-foreground conditions. All methods used the same parameter settings without sequence-specific tuning, and each sequence–method configuration was repeated using three random seeds.

The cross-dataset evaluation results for the four selected CDnet 2014 sequences are summarized in Table 6.

Table 6. Cross-dataset evaluation on selected CDnet 2014 sequences. Values are reported as mean ± standard deviation over three random seeds.

Sequence	Method	Precision	Recall	F1	IoU
highway	Fixed	0.6559 ± 0.0009	0.9136 ± 0.0004	0.7636 ± 0.0008	0.6176 ± 0.0010
	Dynamic	0.6013 ± 0.0004	0.9255 ± 0.0001	0.7290 ± 0.0003	0.5736 ± 0.0004
	Proposed	0.6019 ± 0.0003	0.9256 ± 0.0000	0.7295 ± 0.0002	0.5741 ± 0.0003
canoe	Fixed	0.0620 ± 0.0002	0.5794 ± 0.0020	0.1121 ± 0.0004	0.0594 ± 0.0002
	Dynamic	0.0909 ± 0.0001	0.9241 ± 0.0003	0.1655 ± 0.0001	0.0902 ± 0.0001
	Proposed	0.0917 ± 0.0000	0.9248 ± 0.0001	0.1668 ± 0.0000	0.0910 ± 0.0000
peopleInShade	Fixed	0.6930 ± 0.0016	0.6385 ± 0.0011	0.6646 ± 0.0001	0.4977 ± 0.0002
	Dynamic	0.4719 ± 0.0024	0.9700 ± 0.0000	0.6349 ± 0.0022	0.4651 ± 0.0023
	Proposed	0.4743 ± 0.0008	**0.9701 ± 0.0000**	0.6371 ± 0.0007	0.4675 ± 0.0008
parking	Fixed	0.8135 ± 0.0052	0.6055 ± 0.0039	0.6942 ± 0.0010	0.5317 ± 0.0011
	Dynamic	0.5729 ± 0.0010	0.8496 ± 0.0010	0.6844 ± 0.0010	0.5202 ± 0.0012
	Proposed	0.5749 ± 0.0023	0.8508 ± 0.0007	0.6862 ± 0.0019	0.5223 ± 0.0022
Macro average	Fixed	0.5561	0.6842	0.5586	0.4266
	Dynamic	0.4343	0.9173	0.5535	0.4123
	Proposed	0.4357	0.9178	0.5549	0.4137

On the four selected CDnet sequences, the proposed method achieves a slightly higher F1 and IoU than the standard dynamic-rank LALM baseline in all cases. However, the fixed-rank LALM attains a higher F1 and IoU on several sequences, including highway, peopleInShade, and parking. These results indicate that the benefit of dynamic rank re-

duction is scene dependent and that aggressive rank contraction may not be optimal for backgrounds with stronger temporal variability. The CDnet experiment should therefore be interpreted as complementary cross-dataset evidence rather than as demonstrating universal superiority of the proposed method.

3.7. Ablation Study

To quantify the individual and combined contributions of the noise-aware adaptive regularization strategy and dynamic-rank estimation, a 2 × 2 ablation study was conducted on all 14 SBI sequences. The four configurations correspond to classic versus noise-aware regularization and fixed versus dynamic rank control. All configurations used the same sampled-column setting, convergence criterion, and random-seed protocol as the main SBI evaluation.

As shown in Table 7, introducing dynamic-rank estimation into the classic fixed-rank baseline reduces the mean elapsed time from 81.55 s to 67.95 s and the retained rank from 18.82 to 1.00 while increasing the PSNR from 26.23 to 26.38 dB. Noise-aware regularization alone under fixed-rank control produces mixed changes in the reconstruction metrics, whereas combining it with dynamic-rank estimation further increases the PSNR from 26.38 to 26.61 dB and reduces the AGE from 14.30 to 13.63 and elapsed time from 67.95 s to 57.90 s, with the retained rank remaining at 1.00. These results indicate that dynamic-rank estimation primarily contributes to rank reduction and computational efficiency, while the noise-aware regularization provides its clearest reconstruction-quality benefit when combined with adaptive rank control. Although the adaptive fixed-rank and proposed configurations have lower mean elapsed times in the recorded runs, the runtime measurements exhibit substantial run-to-run variability for some configurations; therefore, these timing differences should be interpreted cautiously.

Table 7. Ablation study of noise-aware adaptive regularization and dynamic-rank estimation on the 14 SBI sequences.

Configuration	PSNR (dB) ↑	AGE ↓	pEPs ↓	pCEPs ↓	Time (s) ↓	Retained Rank
Fixed-rank LALM	26.23 ± 0.04	14.63 ± 0.04	0.2212 ± 0.0016	0.1805 ± 0.0015	81.55 ± 65.36	18.82 ± 0.04
Dynamic-rank LALM	26.38 ± 0.10	14.30 ± 0.14	0.1968 ± 0.0034	0.1599 ± 0.0032	67.95 ± 54.38	1.00 ± 0.00
Adaptive λ + fixed rank	26.10 ± 0.04	14.49 ± 0.04	0.2214 ± 0.0008	0.1792 ± 0.0005	66.22 ± 2.26	18.89 ± 0.04
Proposed	26.61 ± 0.11	13.63 ± 0.18	0.1887 ± 0.0031	0.1518 ± 0.0028	57.90 ± 19.63	1.00 ± 0.00

Note: Results are reported as mean ± standard deviation over ten independent runs. Each run includes all 14 SBI sequences. ↑ indicates that a higher value is better, whereas ↓ indicates that a lower value is better.

4. Discussion

The experimental results indicate that the proposed framework improves the quality–efficiency trade-off of LALM-based background reconstruction. The randomized approximate SVD and dynamic-rank strategy reduce the computational cost of the low-rank updates, while the noise-aware regularization improves the average reconstruction metrics relative to the standard dynamic-rank baseline. However, the sequence-level statistical analysis shows that the quality improvements are not uniformly significant across all SBI sequences. The observed gains should therefore be interpreted primarily as improvements in average performance rather than universal superiority across scenes.

The ablation results further show that the two adaptive components play different but complementary roles. Dynamic-rank estimation primarily drives rank contraction and computational savings, whereas noise-aware regularization provides its clearest reconstruction-quality benefit when combined with dynamic rank control. At the same time, the dispersion score should be regarded as a sequence-level variability indicator rather than a direct physical-noise estimate. Dynamic backgrounds, foreground activity, and illumination

changes can also increase the dispersion score, so the resulting regularization adaptation represents a practical low-rank–sparse trade-off rather than a universally optimal noise model.

A further limitation concerns the observed rank-one convergence of the dynamic-rank variants. Because the requested rank is determined by the singular values retained after SVT and is bounded below by one, the algorithm can remain at rank one once only a single component survives the threshold. Such a representation may be adequate for nearly static backgrounds but may be overly restrictive for scenes containing dynamic backgrounds, illumination changes, or multiple persistent background modes. The SBI results indicate that this behavior is scene dependent rather than uniformly detrimental, and the initialization-sensitivity experiment shows that it is not caused solely by the initial target rank. The CDnet results provide additional evidence of this trade-off: the proposed method improves the dynamic-rank baseline and maintains high foreground recall, whereas the fixed-rank LALM achieves a higher F1-score and IoU on three of the four selected sequences because of its higher precision. Thus, aggressive rank reduction may improve efficiency and foreground sensitivity while allowing some complex background variation to enter the sparse component.

The approximate SVD introduces non-negligible errors in individual singular values even when the matrix reconstruction error is small. Nevertheless, the SVT-preservation experiment shows that most threshold decisions remain correct at the adopted sampled-column setting. The main risk arises for singular values close to the SVT threshold, where underestimation or overestimation may change whether a component is retained and may subsequently affect the dynamic-rank update. More conservative rank-control strategies or improved singular value calibration could therefore be considered for applications requiring greater robustness near the threshold.

Finally, the current implementation uses uniform column sampling for computational simplicity. This choice may be less effective when informative columns are highly nonuniformly distributed, in which case importance-aware or adaptive sampling could improve subspace preservation at additional computational cost. Together with the scene-dependent SBI and CDnet results, this limitation indicates that the present framework should be viewed as an efficient practical approach rather than a universally optimal reconstruction strategy. Future work may therefore focus on more robust rank control, improved singular value calibration, and importance-aware sampling for heterogeneous video scenes.

5. Conclusions

This paper presents a noise-aware randomized LALM framework for efficient matrix reconstruction and video background modeling. Building on established randomized low-rank approximation principles, the framework combines randomized approximate SVD, threshold-driven dynamic rank estimation, and dispersion-based adaptive regularization. The synthetic SVD benchmark demonstrates substantial speedups over full SVD, the SVT experiments show that the approximation preserves threshold decisions at a practical sampled-column setting, and the ten-run SBI experiments show improved average reconstruction quality relative to the fixed-rank and dynamic-rank LALM variants. Additional statistical, parameter-sensitivity, cross-dataset, and ablation analyses indicate that the performance gains are scene dependent and that dynamic rank estimation and noise-aware regularization provide complementary benefits.

The proposed framework improves the quality–efficiency balance of a randomized-SVD-based LALM for RPCA-based video background reconstruction. Despite these encouraging results, the proposed framework has several limitations. First, the current dynamic-rank rule may reduce the retained rank too aggressively for some scenes. Second, the

randomized SVD introduces non-negligible relative errors in the estimated singular values, which may affect SVT threshold calibration. Third, the present implementation relies on uniform column sampling, and more informative or adaptive sampling strategies may offer a better accuracy–efficiency trade-off. Future work will focus on more robust rank-control strategies, improved singular value calibration, and importance-aware or adaptive sampling.

Author Contributions: Conceptualization, N.Z.; methodology, N.Z.; software, N.Z.; validation, N.Z. and W.W.; formal analysis, N.Z.; investigation, N.Z.; data curation, N.Z.; writing—original draft preparation, N.Z.; writing—review and editing, W.W.; visualization, N.Z.; supervision, W.W.; project administration, N.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The datasets used in this study are publicly available in the Scene Background Initialization (SBI) benchmark at http://sbmi2015.na.icar.cnr.it/SBIdataset.html (accessed on 15 August 2026) and the CDnet 2014 benchmark at https://changedetection.net/ (accessed on 8 November 2026).

Conflicts of Interest: Wentao Wang is employed by ZTE Corporation. ZTE Corporation had no role in the conception or design of the study; the collection, analysis, or interpretation of data; the development or implementation of the methodology or software; the preparation of the manuscript; or the decision to submit the manuscript for publication. No funding, data, equipment, or other internal resources from ZTE Corporation were used in this research. The authors declare no other conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

AGE	Average Gray-level Error
ALM	Augmented Lagrangian Method
FPS	Frames Per Second
LALM	Linearized Augmented Lagrangian Method
pCEPs	Percentage of Clustered Error Pixels
pEPs	Percentage of Error Pixels
PSNR	Peak Signal-to-Noise Ratio
RPCA	Robust Principal Component Analysis
SBI	Scene Background Initialization
SNR	Signal-to-Noise Ratio
SVD	Singular Value Decomposition
SVT	Singular Value Thresholding

References

1. Candès, E.J.; Li, X.; Ma, Y.; Wright, J. Robust principal component analysis? *J. ACM* **2011**, *58*, 11. [CrossRef]
2. Lin, Z.; Chen, M.; Ma, Y. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv* **2010**, arXiv:1009.5055.
3. Drineas, P.; Mahoney, M.W. RandNLA: Randomized numerical linear algebra. *Commun. ACM* **2016**, *59*, 80–90. [CrossRef] [PubMed]
4. Halko, N.; Martinsson, P.G.; Tropp, J.A. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.* **2011**, *53*, 217–288. [CrossRef]
5. Martinsson, P.G.; Tropp, J.A. Randomized numerical linear algebra: Foundations and algorithms. *Acta Numer.* **2020**, *29*, 403–572. [CrossRef]
6. Tropp, J.A.; Webber, R.J. Randomized algorithms for low-rank matrix approximation: Design, analysis, and applications. *arXiv* **2023**, arXiv:2306.12418.
7. Udell, M.; Townsend, A. Why are big data matrices approximately low rank? *SIAM J. Math. Data Sci.* **2019**, *1*, 144–160. [CrossRef]
8. Fu, T.; Zong, Z.; Yin, X. Generalized orthogonal matching pursuit with singular value decomposition. *IEEE Geosci. Remote Sens. Lett.* **2021**, *19*, 8013405. [CrossRef]

9. Gilman, K.; Tarzanagh, D.A.; Balzano, L. Grassmannian optimization for online tensor completion and tracking with the t-SVD. *IEEE Trans. Signal Process.* **2022**, *70*, 2152–2167. [CrossRef]
10. Intawichai, S.; Chaturantabut, S. A missing data reconstruction method using an accelerated least-squares approximation with randomized SVD. *Algorithms* **2022**, *15*, 190. [CrossRef]
11. Yao, Q.; Kwok, J.T.; Wang, T.; Liu, T.Y. Large-scale low-rank matrix learning with nonconvex regularizers. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2628–2643. [CrossRef] [PubMed]
12. Cherapanamjeri, Y.; Gupta, K.; Jain, P. Nearly optimal robust matrix completion. In Proceedings of the 34th International Conference on Machine Learning. *PMLR* **2017**, *70*, 797–805.
13. Misra, A.; Abdelgawad, M.A.A.; Jing, P.; Cheung, R.C.C.; Yan, H. Randomized tensor decomposition using parallel reconfigurable systems. *J. Supercomput.* **2025**, *81*, 543. [CrossRef]
14. Maulik, R.; Mengaldo, G. PyParSVD: A streaming, distributed and randomized singular-value-decomposition library. In Proceedings of the 7th International Workshop on Data Analysis and Reduction for Big Scientific Data (DRBSD-7), St. Louis, MO, USA, 14 November 2021; pp. 19–25. [CrossRef]
15. Maddalena, L.; Petrosino, A. Towards benchmarking scene background initialization. In *New Trends in Image Analysis and Processing—ICIAP 2015 Workshops*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2015; Volume 9281, pp. 469–476. [CrossRef]
16. Wang, Y.; Jodoin, P.-M.; Porikli, F.; Konrad, J.; Benezeth, Y.; Ishwar, P. CDnet 2014: An expanded change detection benchmark dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Columbus, OH, USA, 23–28 June 2014; pp. 387–394.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.