A NEW ROBUST SCALABLE SINGULAR VALUE DECOMPOSITION ALGORITHM FOR VIDEO SURVEILLANCE BACKGROUND MODELLING

BY SUBHRAJYOTI ROY*, AYANENDRANATH BASU† AND ABHIK GHOSH‡

Indian Statistical Institute, Kolkata, India,
*roysubhra98@gmail.com; †ayanbasu@isical.ac.in; ‡abhik.ghosh@isical.ac.in

A basic algorithmic task in automated video surveillance is to separate background and foreground objects. Camera tampering, noisy videos, low frame rate, etc., pose difficulties in solving the problem. A general approach which classifies the tampered frames, and performs subsequent analysis on the remaining frames after discarding the tampered ones, results in loss of information. We propose a robust singular value decomposition (SVD) approach based on the density power divergence to perform background separation robustly even in the presence of tampered frames. We also provide theoretical results and perform simulations to validate the superiority of the proposed method over the few existing robust SVD methods. Finally, we indicate several other use-cases of the proposed method to show its general applicability to a large range of problems.

1. Introduction. Automated surveillance from noisy videos is an extremely important problem which has applications in areas such as defence, security, research and monitoring, etc. In such video surveillance, the basic algorithmic task is to separate the background of a video from the foreground or moving objects, based on the input image frames from a surveillance video. The modelled background and foreground are then widely used in different image processing and computer vision applications. The most well-known and oldest applications were monitoring human activities in traffic surveillance systems. However, recently many other applications have been developed based on the same principle of background modelling such as detection of moving objects for visual observation systems of animals and insects to study their behaviours, vision-based hand gesture recognition, autonomous vehicle pilot systems, content based video coding, etc. Even though considerable effort has been given to solve this problem (see Garcia-Garcia, Bouwmans and Silva (2020) for details), various challenges such as presence of noisy frames, low frame rate of camera, change in illumination, multimodal backgrounds, presence of small moving objects, camera tampering still present major hindrances. Mantini and Shah (2019a) define tampering as a change in the view of a surveillance camera which may occur naturally or manually through unexpected or unauthorized actions. Reflection or glare of sunlight onto the camera lens, change in the camera view from daylight mode to night vision mode, defocusing of the camera lens, are some of the natural occurrences of tampering without any human intervention. On the other hand, intentionally covering or obstructing the view of the camera, rotating the camera to point in different directions are examples of manual tampering to accomplish malicious activities. The goal associated with any video surveillance data in presence of camera tampering is primarily to detect the presence of such tampering using classification algorithms, and the rest of the data are used for solving the computer vision problems. Several techniques have been developed to solve this problem (Mantini and Shah, 2019b; Sitara and Mehtre, 2019) based on sophisticated image processing and deep learning techniques intended to classify each frame as tampered or not. However, these techniques achieve good results only at the

Keywords and phrases: Robust SVD, Density Power Divergence, Asymptotic Consistency, Video Processing.
cost of an extensive amount of training data and computing power. Moreover, if the ultimate goal is to perform background modelling, discarding the tampered frames can result in loss of information, since if a frame is partially covered, the uncovered regions may provide useful information about the unauthorized perpetrator. A robust methodology to extract the background and foreground should be able to make use of that information and provide valid inference under contaminated and noisy data.

One approach to background modelling proposed by Candès et al. (2011) is to consider a large matrix $M$ in which each column is constructed by the pixel values of that particular
frame. Since the subsequent frames in a video are correlated, a suitable low rank approximation $L$ should be able to extract the non-moving part of the video, i.e., the background objects, while the difference $(M - L)$ should be able to extract the sparse and noisy part, namely the moving foreground objects. The singular value decomposition (SVD) method is one of the most viable methods for the above low rank approximation. As an illustration, we consider the “freeway” video sequence from UCSD Background Subtraction Dataset (Mahadevan and Vasconcelos, 2010) consisting of 44 frames of grayscale images of dimension $316 \times 236$. It depicts a highway in the background with moving cars as foreground objects having different shades of gray. Since the usual SVD can be seriously affected by these outliers, adding salt-and-pepper noise to only some of the consecutive frames as tampering and applying the usual SVD for background modelling affects the non-tampered frames as well; see Figures 1b and 1d for illustration of this phenomenon. The (few) existing robust SVD methods (see Section 1.1) cannot handle such large $M$ matrices (of dimension $74356 \times 44$) obtained in real-life video surveillance due to their computational inefficiencies, hence, the substitute robust PCA is generally used to solve such problems (Candès et al., 2011; Bouwmans and Zahzah, 2014). Also, as the video surveillance processing needs to be performed in real-time with a very minimal lag, longer execution time of the existing robust SVD algorithms would defeat the purpose and hence would have limited utility for this particular problem. An alternative computationally efficient robust SVD approach would certainly be of extreme practical value in such real-life video surveillance problems. In this paper, we develop a robust SVD algorithm rSVDdpd, which would serve this purpose extremely well; see Figures 1c and 1e. Moreover, even without the presence of any tampering, the proposed rSVDdpd algorithm outputs better foreground estimates than the usual SVD by removing noisy artifacts such as moving shadows, illumination change, etc., which is detailed with an example in Section S4.1 of the Supplementary material.

1.1. Existing Literature. SVD, as a matrix factorization technique from linear algebra, is generally thought of a low rank (equal to the number of positive singular values) approximation of any given linear transformation, which serves the purpose of explaining the transformation as a composition of a rotation followed by a dilation followed by another rotation. Computation of a pseudoinverse or Moore-Penrose inverse of a matrix, efficient computation of the solution to a system of homogeneous linear equations, calculation of range space, null space and rank of a linear transformation, finding the ordinary least square solutions of a system using the Golub and Reinsch algorithm (Golub and Reinsch, 1971) are a few among many well known mathematical applications of singular value decomposition. Throughout the course of its existence, SVD has also been abundantly used in various statistical and machine learning methods for analyzing and modelling data in different real-life applications. In particular, SVD is found in the guise of principal component analysis (PCA), which uses SVD to decompose a data matrix into a low dimensional representation such that these transformed variables can explain the largest possible variability present in the original data with a few of these variables. Various other popular dimension reduction techniques such as correspondence analysis (Greenacre, 2017; Fithian and Josse, 2017; Kamalja and Khangar, 2017), latent semantic indexing (Hofmann, 1999; Mirzal, 2016; Anandarajan, Hill and Nolan, 2019) and clustering techniques (Drineas et al., 2004; Maheswari and Duraiswamy, 2008; Zekri, Mokhtari and Cohen, 2016; Cheng et al., 2019; Mardia et al., 2005) also rely on singular value decomposition at their core. The clustering methods based on SVD generally rely on two major approaches. In case of graph-based clustering, the graph Laplacian matrix is decomposed using SVD to find a lower rank approximation which can reveal the clusters present in the graph. The other approach views the problem of clustering as a least square minimization problem and uses SVD to solve it efficiently. Also, different generalizations of
SVD such as K-SVD based on k-means clustering algorithm (Aharon, Elad and Bruckstein, 2006) and tensor-SVD based on multi-view clustering approach (Zhang and Xia, 2018; Gao et al., 2020), are found to be extremely useful in tackling different types of clustering problems. SVD is also widely used in the domain of pattern recognition, including the signal, image or video processing. A nice review of the SVD-based image processing techniques is presented by Sadek (2012). Other important specific applications include image watermarking schemes (Dappuri, Rao and Sikha, 2020; Meenakshi, Swaraja and Kora, 2020; Thakkar and Srivastava, 2017), signal denoising and feature enhancements (Zhao and Jia, 2017), audio watermarking (Özer, Sankur and Memon, 2005; Rezaei and Khalili, 2019), sound source localization (Grondin and Glass, 2019a,b), sound recovery techniques (Zhang et al., 2016), etc. Moreover, SVD is recently being used extensively to solve different problems in bioinformatics, which include the analysis of protein functional associations (Franceschini et al., 2016), clustering for gene expression analysis (Horn and Axel, 2003; Liang, 2007; Bustamam, Formalidin and Siswantining, 2018), protein coding region prediction (Das, Das and Nanda, 2017), etc. In geographical science as well, Kumar, Nasser and Sarker (2011) used SVD based techniques (in fact, its robust version) in order to find proper graphical representations of climate data, mitigating the effect of outlying thunderstorms and heavy rains. Such a wide range of applications clearly underscores the relevance of SVD as an extremely integral component of data analysis across a multitude of disciplines.

However, as already indicated by several authors including Kumar, Nasser and Sarker (2011); Hawkins, Liu and Young (2001); Liu et al. (2003), the usual method of computing SVD is highly susceptible to outliers in the data matrix. As the data are becoming increasingly vast and complex in the present era, it is also being susceptible to the inclusion of different forms of noises, corruptions and contamination by outlying observations. Thus our proposal rSVDpd as a robust SVD algorithm could be useful for a plethora of applications, other than our aforementioned primary target of the video surveillance problem.

Among the existing robust SVD algorithms, one of the earliest attempts was made by Ammann (1993) who considered it as a special case of the projection pursuit problem to be solved via using the transposed QR algorithm. Several authors (Hawkins, Liu and Young, 2001; Liu et al., 2003; Ke and Kanade, 2005) viewed the problem of computing the usual SVD as a least squares problem (see Section 2), and accordingly developed a robust extension using an alternating $L_1$ regression algorithm with the least absolute deviation (LAD) loss function. Jung (2010) pointed out that, although such a simple LAD approach of robustly computing the SVD always converges, it is still heavily sensitive to high leverage points. Jung circumvented this problem by using a weighted LAD approach with a suitable choice of weights. Rey (2007) proposed a method to robustly tackle the alternating regression problem in the computation of SVD using the Huber weight function (Huber, 2004). Rey proposed the use of “Total” least squares (Markovsky and Van Huffel, 2007) since both of the data matrix and the left and right singular vectors are subject to errors in the measurements, instead of using the usual least squares where the predictor covariates are assumed to be fixed without any error and the only source of error is associated with the response variable. With the combination of this “Total” least squares objective function and Huber’s weight function as a penalty term for any outlying observations, Rey developed the “Total” SVD method which is more robust than the usual SVD. However, as mentioned in the original paper itself (Rey, 2007), this method has severe convergence issues. Alternatively, Zhang, Shen and Huang (2013) used the same Huber weight function in defining the loss and combined it with a squared error based penalty function for regularization to construct the minimization criterion for obtaining a robust SVD estimator. Wang (2017) used a myriad estimator derived from an $\alpha$-stable distribution with the cost function $\rho(x) = \log(x^2 + K^2)$, where $K$ is a tuning parameter which provides a balance between a mean type estimator and a mode type estimator, thus creating a bridge
between robustness and efficiency. However, the suitable value of such tuning parameter $K$ cannot be easily obtained unless a large amount of data are present, and hence a heuristic process has been suggested.

Except for the simple case with alternating $L_1$ regression approach by Gabriel and Zamir (1979), no theoretical guarantee of the convergence of the computational algorithms or any kind of theoretical properties of the resulting SVD estimates are established in the literature. In most cases, the resulting singular vectors are also not necessarily orthogonal to each other. In addition, these algorithms are computationally challenging and become infeasible for the modern-day large data matrices, for instance in the video surveillance application discussed earlier.

1.2. Our Contribution. In this paper, we develop a new, efficient approach of computing robust SVD, that can easily be employed for extremely large data matrices as well. As noted above, most of the previous works considered the problem of computing the SVD as an alternating regression problem and used suitable robust loss functions (instead of the usual squared-error loss). We will also take the same route but propose to use the novel density power divergence (DPD) (Basu et al., 1998) based loss function for estimating the robust SVD. The DPD and the associated minimum DPD estimator (MDPDE) have been observed to produce robust as well as highly efficient estimators under several important situations in statistics and information theory (see, e.g., (Basu et al., 1998); a brief description is also provided in the Section S1 of the Supplementary Material accompanying this paper). As a result, the proposed use of the DPD-loss function automatically brings all its nice statistical properties in estimating the robust SVD. It also makes it possible to provide a balance between robustness and efficiency based on the amount of expected contamination present in the data via the specification of a hyperparameter $\alpha$. Our proposed DPD-based method of computing robust SVD, which we will refer to as the rSVDdpd, is also particularly suited to the estimation of singular vectors that are orthogonal to each other, unlike most existing approaches. Additionally, in this paper, we also derive various mathematical properties of our proposed algorithm including the theoretical guarantees of its convergence and the asymptotic consistency for the resulting SVD estimators under some suitable regularity conditions, which further strengthen the base of our algorithm. Extensive simulation studies are conducted to illustrate the performance of the proposed rSVDdpd method under different types of possible data contamination. The proposal is also compared (empirically) with two common existing approaches of computing robust SVD, namely the ones proposed by Zhang, Shen and Huang (2013) and Hawkins, Liu and Young (2001); our proposed algorithm rSVDdpd is found to clearly outperform these two existing methods in all simulation setups with contamination. For simulation setups with uncontaminated error distributions, the regularized variant of RobRSVD is found to be the best, but its high computational complexity poses a challenge to apply this technique for large matrices. In comparison, the proposed rSVDdpd method achieves near-optimal performance with a much lower computational complexity, making it suitable for general purpose usages. Among various possible real-data applications, our method is successfully applied to obtain the desired inference for the video surveillance problem, for which existing methods of robust SVD are found to be computationally demanding as compared to rSVDdpd. On the other hand, to demonstrate the general applicability of the proposed algorithm beyond the video surveillance problem, we use rSVDdpd to obtain a robust estimate of the latent financial market index which is described in Section S4.3 of the Supplementary Material.

The rest of the paper is organized as follows. In Section 2, we formulate the necessary mathematical setup and develop the proposed robust SVD algorithm, the rSVDdpd. In Section 3, we establish some desirable theoretical properties of the rSVDdpd. In particular, we
prove the convergence and asymptotic consistency of rSVDdpd under some suitably chosen regularity assumptions. Section 4 presents simulation studies under different setups and compares the performance of rSVDdpd with existing algorithms. Finally, interesting real life applications are presented in Section 5, followed by conclusions and possible limitations of the work in Section 6. For brevity in presentation, the proofs of all of the theoretical results are presented in Section S3 of the Supplementary Material.

2. The proposed rSVDdpd: Theoretical formulation and the Algorithm. The problem of singular value decomposition starts with a data matrix \( X \) of dimension \( n \times p \) (\( n \) and \( p \) may be different). The matrix \( X \) is assumed to have rank \( r \), where \( r \) is much lower than \( n \) or \( p \). However, in real-life applications, such low rank representation of the observed data matrix may be approximate but not exact, in the sense that, the matrix \( X \) can be written as

\[
X = \sum_{k=1}^{r} \lambda_k u_k v_k^T + E,
\]

where \( u_k \) is a vector of length \( n \) and \( v_k \) is a vector of length \( p \) for \( k = 1, 2, \ldots, r \), and \( E \) is a \( n \times p \) matrix. The entries of the error matrix \( e_{ij} \)s are generally expected to be smaller in magnitude than the corresponding entry of the data matrix \( X \), when the data are not contaminated. We also need the vectors \( \{u\} = \{1, 2, \ldots, r\} \)’s and \( \{v\} = \{1, 2, \ldots, r\} \)’s to be orthonormal, i.e.,

\[
 u_l^T u_l = \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{if } k \neq l, \end{cases} \quad v_l^T v_l = \begin{cases} 1 & \text{if } k = l, \\ 0 & \text{if } k \neq l, \end{cases}
\]

where \( k, l = 1, 2, \ldots, r \).

The vectors \( u_k \)s and \( v_k \)s are called the left and the right singular vectors, respectively, whereas \( \lambda_k \)’s are called the singular values of the matrix \( X \). As in Rey (2007), the above description of the SVD can also be reformalized into

\[
X = \sum_{k=1}^{r} a_k b_k^T + E,
\]

where \( a_k \)s and \( b_k \)s are still orthogonal sets of vectors for \( k = 1, 2, \ldots, r \), but not necessarily normalized. Once the estimates of \( a_k \)s and \( b_k \)s are known, they can be normalized to obtain the \( u_k \)s and \( v_k \)s and the singular values are then given by \( \lambda_k = \|a_k\| \|b_k\| \) for each \( k = 1, \ldots, r \), where \( \| \cdot \| \) denotes the usual Euclidean (\( L_2 \)) norm.

For the computation of the left and right singular vectors from Eq. (2.1), one can proceed sequentially focusing only on a rank one approximation as follows. First, consider the approximation \( X \approx a_1 b_1^T \). Once we have estimates for \( \hat{a}_1 \) and \( \hat{b}_1 \), we can obtain \( \hat{\lambda}_1 = \|\hat{a}_1\| \|\hat{b}_1\| \), and normalize \( \hat{a}_1 \) and \( \hat{b}_1 \) as required. Next, we compute the residual matrix \( X - \hat{\lambda}_1 \hat{a}_1 \hat{b}_1^T \), and the rank one approximation algorithm can be used again on this residual matrix, to obtain the second singular value and the corresponding vectors. Proceeding similarly, one can compute all \( r \) singular values and singular vectors of the given data matrix \( X \). So, throughout the rest of the paper unless otherwise specified, we will consider the problem of obtaining the vectors \( a \) and \( b \) yielding the best rank-one approximation of the data matrix \( X \) as

\[
X \approx a b^T.
\]

Accordingly, in the following, we shall use \( a_i \) for \( i = 1, 2, \ldots, n \), to denote the elements of the vector \( a \) and similarly, \( b_j \) for \( j = 1, 2, \ldots, p \), to denote the elements of the vector \( b \). The elements of the \( X \) matrix are denoted by \( x_{ij} \).
2.1. The Regression based approach. We consider the idea of transforming the SVD estimation problem into a regression problem, following Rey (2007). For this purpose, let us first fix the index $j$ (i.e., the column index), and consider the setup

$$X_{ij} = a_i b_j + e_{ij}, \quad i = 1, 2, \ldots, n,$$

which is a simple linear regression problem without any intercept. Here, $x_{ij}$s are observed response variables and $e_{ij}$s are random error components. Therefore, for any given value of $a_i$, we can treat $a_i$ as covariate values and estimate the $b_j$ as regression slopes, and as we vary the column index $j = 1, 2, \ldots, p$, we are posed with $p$ such linear regression problems, jointly yielding an estimate of $b = (b_1, \ldots, b_p)$. Next, given these estimated values of $b_j$'s, we treat them as covariates in Eq. (2.3) and estimate the $a_i$'s as the unknown regression parameters for each $i = 1, \ldots, n$. Repeating these two steps sequentially until convergence, we get the final desired estimates of $a$ and $b$ from which the desired singular value and vectors can be obtained as discussed previously.

Now, for estimating the regression coefficients ($b_j$ or $a_i$) in each iteration of the above-mentioned alternating regression approach, we propose to use the robust and efficient minimum DPD estimation procedure. The minimum DPD estimator (MDPDE) was initially developed for independent and identically distributed data in Basu et al. (1998), and later extended to the more general independent non-homogeneous set-up in Ghosh and Basu (2013); see Section S1 of the Supplementary Material for more details. It considers a general form of DPD-based loss function involving

$$V_i(y_i; \theta) = \left[ \int f_i(y; \theta)^{1+\alpha} dy - \left( 1 + \frac{1}{\alpha} \right) f_i(y_i; \theta)^{\alpha} \right],$$

where $\alpha > 0$ is the robustness controlling parameter, and the distribution of the $i$-th sample observation $y_i$ is modeled using a family of distributions $F_i = \{ f_i(y; \theta) : \theta \in \Theta \}$ for each $i$. As a particular case, the MDPDE under the linear regression model with normally distributed errors are also discussed in Ghosh and Basu (2013), which can be used in the present context. For our purpose, assuming the error components $e_{ij}$s in Eq. (2.3) to be independent and normally distributed with mean zero and variance $\sigma^2$, the MDPDE objective function for estimating the unknown parameters $a_i$ or $b_j$ and $\sigma^2$ from Eq. (2.3), turns out to be

$$H_{n,p}(a, b, \sigma^2) = \frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} V(x_{ij}; a_i, b_j, \sigma^2),$$

where

$$V(y; c, d, \sigma^2) = \frac{1}{(2\pi)^{\alpha/2}\sigma^\alpha \sqrt{1+\alpha}} - \frac{(1+\alpha)}{\alpha} \frac{1}{(2\pi)^{\alpha/2}\sigma^\alpha} \exp \left\{ -\frac{\alpha(y-cd)^2}{2\sigma^2} \right\},$$

is a simplification of Eq. (2.4) with $f_i(y, \theta)$ as the density of a normal distribution with mean $cd$ and variance $\sigma^2$. The MDPDE of the parameter $\theta = (a, b, \sigma^2)$ is then obtained by minimizing this objective function in Eq. (2.5) with respect to each of the components of the triplet. So, instead of estimating the elements of the scaled singular vectors $a$ and $b$ i.e., $a_i$'s and $b_j$'s in the alternating regression models, we can indeed minimize this MDPDE objective function iteratively over either of the parameters ($a_i$, $b_j$ or $\sigma^2$) given the most recent estimates of the other parameters. By standard differentiation, as in Ghosh and Basu (2013), the corresponding estimating equations turn out to be

$$\sum_{j=1}^{p} b_j (x_{ij} - a_i b_j) e^{-\alpha \frac{(x_{ij} - a_i b_j)^2}{2\sigma^2}} = 0, \quad i = 1, \ldots, n,$$
usual SVD estimators for SVD. Hawkins, Liu and Young (2001) have shown that such alternating estimating equations lead to the usual least square SVD estimates. Thus, the proposal is a generalization of the usual SVD estimator to gain robustness with increasing $\alpha > 0$, with $\alpha = 0$ recovering the usual SVD.

From another viewpoint, the mentioned proposal yields a robust generalization of the usual SVD estimators for $\alpha > 0$, in contrast to Remark 1. To see this, denoting $w_{ij} = e^{-\alpha (x_{ij} - a_i b_j)^2 / 2\sigma^2}$, it follows that the estimating equations in Eq. (2.7)-(2.9) can be rearranged into the form

\begin{equation}
\sum_{i=1}^{n} a_i (x_{ij} - a_i b_j) e^{-\alpha (x_{ij} - a_i b_j)^2 / 2\sigma^2} = 0, \quad j = 1, \ldots, p, \tag{2.8}
\end{equation}

\begin{equation}
\sum_{i=1}^{n} \sum_{j=1}^{p} \left[ 1 - \frac{(x_{ij} - a_i b_j)^2}{\sigma^2} \right] e^{-\alpha (x_{ij} - a_i b_j)^2 / 2\sigma^2} = \frac{\alpha}{(1 + \alpha)^{3/2}}. \tag{2.9}
\end{equation}

**Remark 1.** As the robustness parameter $\alpha$ tends towards zero, the quantity $V(y; c, d, \sigma^2)$ in Eq. (2.6) tends towards the normalized squared error $(y - cd)^2 / 2\sigma^2$. Thus, as $\alpha \to 0$, the MDPDE objective function in Eq. (2.5) will converge to a scaled version of the usual squared error loss function between the elements $x_{ij}$ of the data matrix and the rank one estimates $a_i b_j$, i.e.,

$$
\lim_{\alpha \to 0} H_{n, p}(a, b, \sigma^2) = \frac{1}{2\sigma^2} \sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij} - a_i b_j)^2.
$$

The alternating estimating equations Eq. (2.7)-(2.9) then turn out to be explicitly solvable, with

$$
a_i = \frac{\sum_{j} x_{ij} b_j}{\sum_{j} b_j^2}, \quad i = 1, \ldots, n; \quad b_j = \frac{\sum_{i} x_{ij} a_i}{\sum_{i} a_i^2}, \quad j = 1, \ldots, p; \quad \sigma^2 = \frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} (x_{ij} - a_i b_j)^2.
$$

Hawkins, Liu and Young (2001) have shown that such alternating estimating equations lead to the usual least square SVD estimates. Thus, the proposal is a generalization of the usual SVD estimator to gain robustness with increasing $\alpha > 0$, with $\alpha = 0$ recovering the usual SVD.

In the first estimating equation of Eq. (2.10), as $x_{ij} \approx a_i b_j$, i.e., $x_{ij} / b_j \approx a_i$, the equation considers the weighted mean of $x_{ij} / b_j$ with weights $b_j^2 w_{ij}$. Similar normalization is also present in the second equation. In the third equation, we see that $\sigma^2$ is approximately a normalized version of the squared residuals $(x_{ij} - a_i b_j)^2$ except for a term subtracted from the denominator to make the corresponding estimating equation unbiased. Thus, each of the parameters are estimated as a weighted average of different crude estimates based on the elements of the data matrix $X$ and the most recent values of the other parameters. Also, for any $\alpha > 0$, $w_{ij} = e^{-\alpha (x_{ij} - a_i b_j)^2 / 2\sigma^2}$ is a decreasing function of the magnitude of the residuals, i.e., of $|x_{ij} - a_i b_j|$. This decreasing nature is crucial to make the overall estimates robust since a large deviation from the model having large $|x_{ij} - a_i b_j|$ would yield smaller weights, and hence those outlying values would have little effect on the estimating equations given in Eq. (2.10).

The particular form of these weights $w_{ij}$ in Eq. (2.10) appears due to the assumption of normality of the errors $e_{ij}$. In a more general version with appropriately modified weight
function, one can consider the estimating equations

\[
a_i = \frac{\sum_j b_j x_{ij} \psi(|x_{ij} - a_i b_j|)}{\sum_j b_j^2 \psi(|x_{ij} - a_i b_j|)}, \quad i = 1, \ldots, n,
\]

\[
b_j = \frac{\sum_i a_i x_{ij} \psi(|x_{ij} - a_i b_j|)}{\sum_i a_i^2 \psi(|x_{ij} - a_i b_j|)}, \quad j = 1, \ldots, p,
\]

\[
\sigma^2 = \frac{\sum_i \sum_j (x_{ij} - a_i b_j)^2 \psi(|x_{ij} - a_i b_j|)}{\sum_i \sum_j \psi(|x_{ij} - a_i b_j|) - S_\psi}.
\]

where \( \psi(\cdot) \) is a suitably smooth and decreasing function of its argument, and \( S_\psi \) is a suitably chosen quantity to make the estimating equation corresponding to \( \sigma^2 \) unbiased.

**Remark 2.** Cichocki, Cruces and Amari (2011) uses a generalized density power divergence, namely the alpha-beta divergence (Cichocki and Amari, 2010) in order to perform robust non-negative matrix factorization (NMF). However, this method solely relies on the fact that the entries of the data matrix \( X \) and the entries of the factorization remain positive throughout the course of the algorithm. Since, in case of estimation of singular values, such a restriction cannot be imposed on the left and right singular vectors \( a, b \), the same procedure of NMF cannot be extended to the problem of SVD.

### 2.2. Orthogonalization: Regression with a Penalty Parameter.

In the mathematical framework developed in Section 2.1, there is no component in the regression formulation that ensures orthogonality of the singular vectors. It is natural to extend the objective function \( H_{n,p}(a, b, \sigma^2) \) of Eq. (2.5) using a penalty function to achieve the orthogonality requirements. One such penalty function could be the sum of the inner products (dot-products) of the current singular vector with all preceding singular vectors.

Thus, we can again use the alternating regression approach to estimate the parameters \( a, b, \sigma^2 \) iteratively and consider the penalized MDPDE objective function as

\[
\tilde{H}_{k,n,p}(a, b, \sigma^2) = \frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} V(x_{ij}^t; a_i, b_j, \sigma^2) + \frac{\kappa}{k} \left( \sum_{r=1}^{k} \frac{(a^T a_r)^2}{\|a_r\|^2} + \sum_{r=1}^{k} \frac{(b^T b_r)^2}{\|b_r\|^2} \right),
\]

\[k = 1, 2, \ldots \text{rank}(X - 1),\]
where $x'_{ij}$ is the $(i, j)$-th element of $X'$ and $\xi$ is a penalty parameter balancing between the ability of singular vectors being able to express the data matrix, and the ability of being orthogonal with the previously obtained scaled singular values. Notice that the normalization factors $\|a_r\|^2$ and $\|b_r\|^2$ have been used in Eq. (2.14) to make sure that each penalty term is of similar magnitude.

Again, differentiating the objective function in Eq. (2.14) with respect to the individual parameters and setting them equal to 0, yields the estimating equations for the parameters $a, b, \sigma^2$. A reorganization of the estimating equations can be done to obtain a fixed point iteration formula similar to Eq. (2.10) as given by

\begin{equation}
\begin{aligned}
a_i &= \left( \sum_{j=1}^{p} b_j^2 w_{ij} \right)^{-1} \left[ \sum_{j=1}^{p} b_j x'_{ij} w_{ij} + \frac{np(2\pi)^{(\alpha/2)}\sigma^2 \xi}{k(1 + \alpha)} \sum_{r=1}^{k} (a_r^T a_r) a_r \right], \quad i = 1, \ldots n, \\
b_j &= \left( \sum_{i=1}^{n} a_i^2 w_{ij} \right)^{-1} \left[ \sum_{i=1}^{n} a_i x'_{ij} w_{ij} + \frac{np(2\pi)^{(\alpha/2)}\sigma^2 \xi}{k(1 + \alpha)} \sum_{r=1}^{k} (b_r^T b_r) b_r \right], \quad j = 1, \ldots p, \\
&\quad k = 1, 2, \ldots \text{rank}(X) - 1,
\end{aligned}
\end{equation}

where $w_{ij} = e^{-\alpha \left( x'_{ij} - a_i b_j \right)^2}$. Since, the penalty term is not a function of $\sigma^2$, the estimating equation corresponding to $\sigma^2$ remains same as in Eq. (2.10).

**Remark 3.** It remains a natural question about the choice of the penalty parameter $\xi$, which is often obtained through cross validation procedures. Also, it was found that the convergence of the algorithm with estimating equations Eq. (2.15) is highly sensitive to the choice of $\xi$ and a bad choice of $\xi$ can lead to numerical underflow or overflow, thus compromising the convergence of the algorithm.

2.3. The Final Algorithm: rSVDdpd. As we have noted in Remark 3, there are practical issues of convergence and numerical instability of the estimation process using Eq. (2.15). In order to circumvent this problem, the similarity between the iterative equations in Eq. (2.10) and in Eq. (2.15) can be exploited. The penalized MDPDE based iterative equations have an extra term appearing in the numerator consisting of the inner products between the current estimate of the singular vector and the former singular vectors. This term appears due to the penalty cost in the objective function (see Eq. (2.14)) and is very similar to the orthogonalization trick of Gram Schimdt orthogonalization process (Giraud, Langou and Rozloznik, 2005). Thus a possible trick to ensure orthogonalization between the estimated singular vectors is to use the iterative equations in Eq. (2.10) along with an additional step in between the alteration of the parameters similar to the Gram Schimdt orthogonalization process. In particular, between alternatively using Eq. (2.7)-(2.9), the estimates $a$ and $b$ are updated as

\begin{equation}
\begin{aligned}
a &\leftarrow a - \sum_{r=1}^{k} a_r^T a_r, \quad b &\leftarrow b - \sum_{r=1}^{k} b_r^T b_r, \quad k = 1, \ldots, \text{rank}(X) - 1,
\end{aligned}
\end{equation}

where the symbol $\leftarrow$ denotes the assignment operator, i.e., the value of left hand side of Eq. (2.16) is updated with the value of the right hand side. Note that, such an orthogonalization step need not be performed for the first singular value, but is performed only from the estimation of second and subsequent singular values. Including the singular values $\lambda$ and restricting the vectors $a$ and $b$ to have unit $L_2$-norm, we ultimately consider the estimation of the parameter $\theta = (\lambda, a, b, \sigma^2)$. The final algorithm has been outlined below in Algorithm 1.
Algorithm 1: rSVDdpd: Algorithm for computing Robust SVD using DPD

Input: Data matrix $X$, A tolerance $tol$, Robustness control parameter $\alpha$

$number\_of\_singular\_vectors \leftarrow 0$;

while $\|X\|_2 > tol$, i.e., there is significant singular value do

$k \leftarrow number\_of\_singular\_vectors$;

Let $a_1, a_2, \ldots, a_k$ and $b_1, b_2, \ldots, b_k$ are $k$ estimated singular vectors already obtained;

Initialize $a_{k+1}^{(0)}$ and $b_{k+1}^{(0)}$ which are unit vectors of length $n$ and $p$ respectively;

Initialize singular value $\lambda_{k+1}^{(0)} \leftarrow 0$ and $(\sigma^2)^{(0)}$ as something positive;

while $\|a_{k+1}^{(t+1)} - a_{k+1}^{(t)}\|$ and $\|b_{k+1}^{(t+1)} - b_{k+1}^{(t)}\|$, both are not small do

$c \leftarrow a_{k+1}^{(t+1)}$;

Repeat until convergence

\[ w_{ij} \leftarrow \exp\left(\frac{(x_{ij} - c_{b_{k+1}^{(t+1)}})^2}{2(\sigma^2)^{(t)}}\right); \]

forall $i = 1, 2, \ldots, n$

\[ c_i \leftarrow \frac{\sum_j w_{ij}^{(t)} x_{ij} (b_{k+1}^{(t+1)})_j}{\sum_j w_{ij}^{(t)} (b_{k+1}^{(t+1)})_j^2}; \]

end

$c \leftarrow c - \sum_{r=1}^k c^\top a_r$ (This is the step corresponding to Gram Schimdt process);

$\lambda^{(t+1)} \leftarrow \|c\|$;

$a_{k+1}^{(t+1)} \leftarrow c/\lambda^{(t+1)}$;

$d \leftarrow b_{k+1}^{(0)}$;

Repeat until convergence

\[ w_{ij} \leftarrow \exp\left(\frac{(x_{ij} - a_{b_{k+1}^{(t+1)}})^2}{2(\sigma^2)^{(t)}}\right); \]

forall $j = 1, 2, \ldots, p$

\[ d_j \leftarrow \frac{\sum_i w_{ij}^{(t)} x_{ij} (a_{b_{k+1}^{(t+1)}})_i}{\sum_i w_{ij}^{(t)} (a_{b_{k+1}^{(t+1)}})_i^2}; \]

end

$d \leftarrow d - \sum_{r=1}^k d^\top b_r$ (This is the step corresponding to Gram Schimdt process);

$\lambda^{(t+1)} \leftarrow \|d\|$;

$b_{k+1}^{(t+1)} \leftarrow d/\lambda^{(t+1)}$;

$e_{ij} \leftarrow x_{ij} - \lambda^{(t+1)} a^{(t+1)}_{(k+1)i} b^{(t+1)}_{(k+1)j}$;

$w_{ij} \leftarrow \exp(e_{ij})$;

$\sigma^2^{(t+1)} \leftarrow \frac{\sum_i \sum_j w_{ij}^{(t)} e_{ij}^2}{\sum_i \sum_j w_{ij}^{(t)} - (\alpha/(1 + \alpha)^{3/2})}$;

end

$number\_of\_singular\_vectors \leftarrow number\_of\_singular\_vectors + 1$;

$X \leftarrow X - \lambda^{(t+1)} a^{(t+1)}_{(k+1)} b^{(t+1)}_{(k+1)}^\top$;

end
3. Mathematical Properties. In order to study the properties of the proposed robust SVD algorithm rSVDdpd and the obtained robust estimates, we shall restrict our study to the properties of only the first singular value and corresponding vectors obtained from a rank one approximation of the data matrix $X$. As mentioned in Section 2, the estimation process of the subsequent singular values and vectors is based on the rank one approximations of the residual matrix $X' = X - \tilde{\lambda}_1 \tilde{a}_1 \tilde{b}_1^T$. In order to make sure all the results developed in this section are also true for the subsequent singular values and vectors, the assumptions on the distribution of the data matrix $X$ must hold for the residual matrix $X'$ as well. We assume that this is the case and concern ourselves with only the study of rank-one approximation. This assumption is indeed true for the situations where the true distributions of the random variable $X_{ij}$ denoting the elements of the data matrix $X$ belong to a location family of distributions with location parameters $\sum_{r=1}^{k} \lambda_r a_{ir} b_{jr}$, where the vectors $a_1, \ldots, a_r$ and $b_1, \ldots, b_r$ are of unit $L_2$ norm and $\lambda_r$ denotes the true singular values of the matrix.

As mentioned in Section 2.1, the proposed robust SVD algorithm, the rSVDdpd, estimates the parameter by an alternating iterative method, in each of which the MDPDE objective function given in Eq. (2.5) is minimized with respect to a single parameter (a $\lambda$ or $b_j$ or $\sigma^2$), fixing the values of the other parameters at their most recent estimates during the iterative process. However, at the end of the iteration, the singular value is estimated as $\hat{\lambda}_1 = ||\tilde{a}_1|| ||\tilde{b}_1||$, and the estimated scaled singular vectors $a$ and $b$ are normalized. Hence, from here onwards, we shall denote $\theta = (\lambda, \{a_i\}_{i=1}^{n}, \{b_j\}_{j=1}^{p}, \sigma^2)$ as the parameters to be estimated, where $a_i$ and $b_j$ denotes the $i$-th and $j$-th coordinate of the vectors $a$ and $b$, respectively. Since, the vectors $a$ and $b$ are normalized to have unit $L_2$ norm, i.e., $\sum_i a_i^2 = \sum_j b_j^2 = 1$, the parameter space $\Theta$ is an $(n+p)$ dimensional manifold in $\mathbb{R}^{(n+p+2)}$, namely $[0, \infty) \times S_n \times S_p \times [0, \infty)$, where $S_n$ and $S_p$ denote the $n$ and $p$-dimensional hyperspheres of unit radius centered at the origin respectively.

For any non-negative integer $t$, let us denote $\theta^{(t)} = (\lambda^{(t)}, \{a_i^{(t)}\}, \{b_j^{(t)}\}, (\sigma^2)^{(t)})$ as the estimate obtained at $t$-th iteration. Then, these estimates are related as

\begin{align}
\lambda^{(t+1/2)} a_i^{(t+1)} &= \arg \min_{a} \frac{1}{p} \sum_{j=1}^{p} V(x_{ij}; a, b_j^{(t)}, (\sigma^2)^{(t)}), \quad i = 1, \ldots, n, \quad t = 0, 1, \ldots \\
\lambda^{(t+1)} b_j^{(t+1)} &= \arg \min_{b} \frac{1}{n} \sum_{i=1}^{n} V(x_{ij}; a_i^{(t+1)}, b, (\sigma^2)^{(t)}), \quad j = 1, \ldots, p, \quad t = 0, 1, \ldots \\
(\sigma^2)^{(t+1)} &= \arg \min_{\sigma^2} \frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} V(x_{ij}; a_i^{(t+1)}, b_j^{(t+1)}, \sigma^2), \quad t = 0, 1, \ldots
\end{align}

where $x_{ij}$ is the $(i, j)$-th entry of the data matrix $X$ and $V$ is the density power divergence loss function given in Eq. (2.6). On the other hand, the ultimate “true” value of the parameter that is being estimated, i.e., $\theta^0 = (\lambda^0, \{a_i^0\}_{i=1}^{n}, \{b_j^0\}_{j=1}^{p}, (\sigma^0)^2)$ is also expected to satisfy such minimization criteria, in the sense of overall population based measures rather than its empirical counterparts, and, maintain a relationship similar to Eq. (3.1)-(3.3) (again, a population version of these). With these in mind, we start by defining these “best” fitting parameters for the particular setup of SVD under consideration.

**Definition 1.** Let, $X$ be a data matrix of order $n \times p$ such that the random variable $X_{ij}$, corresponding to its $(i, j)$-th entry $x_{ij}$, follows a distribution with density function $g_{ij}$, for all $i = 1, \ldots, n$ and $j = 1, \ldots, p$ and $X_{ij}$s are independent to each other. Then, $\theta^0 = (\lambda^0, \{a_i^0\}_{i=1}^{n}, \{b_j^0\}_{j=1}^{p}, (\sigma^0)^2)$ is called a “best” fitting parameter if the following conditions hold.
1. The vectors \( a^g = (a^g_1, \ldots, a^g_n) \) and \( b^g = (b^g_1, \ldots, b^g_p) \) are unit vectors, i.e.,

\[
\sum_{i=1}^{n} (a^g_i)^2 = 1 \quad \text{and} \quad \sum_{j=1}^{p} (b^g_j)^2 = 1.
\]

2. For any \( i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, p, \)

\[
\lambda^g a_i^g = \arg \min_a \int V(x; a, b^g_j, (\sigma^g)^2)g_{ij}(x)dx.
\]

3. For any \( i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, p, \)

\[
\lambda^g b_j^g = \arg \min_b \int V(x; a^g_i, b, (\sigma^g)^2)g_{ij}(x)dx.
\]

4. For any \( i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, p, \)

\[
(\sigma^g)^2 = \arg \min_{\sigma^2} \int V(x; \lambda^g a_i^g, b_j^g, \sigma^2)g_{ij}(x)dx
\]

\[
= \arg \min_{\sigma^2} \int V(x; a_i^g, \lambda^g b_j^g, \sigma^2)g_{ij}(x)dx.
\]

Here, Eq. (3.5) shows that the minimizer of the quantity on the right-hand side of the equation, is always \( \lambda^g a_i^g \), independent of the choice of column index \( j \). This assumption holds if the true densities \( g_{ij} \) are normal densities with the location parameters being elements from the best rank one approximation of \( X \), i.e., the entries of the data matrix \( X_{ij} \) are normally distributed with mean \( \mu_{ij} \) and constant variance \( \sigma^2 \), and the matrix \( \mu = (\mu_{ij})_{i=1,j=1}^{np} \) is of rank 1.

3.1. Properties of a best fitting parameter. Since the aim of the rSVDdpd algorithm is to robustly estimate the singular values and the singular vectors of a given data matrix, it is required to show that the “best” fitting parameters as introduced by Definition 1 resemble the behaviour of the usual singular values and vectors. In this connection, the following theorem establishes that if the elements of the data matrix \( X \) are normally distributed, the “best” fitting parameter given by Definition 1 matches with the usual notion of singular values and vectors.

**Theorem 1.** Let the data matrix \( X \) be such that \( X_{ij} \sim \mathcal{N} \left( \lambda^* a_i^* b_j^*, (\sigma^*)^2 \right) \) i.e., \( X_{ij} \) follows a normal distribution with mean \( \lambda^* a_i^* b_j^* \) and variance \( (\sigma^*)^2 \), for \( i = 1, 2, \ldots, n, \ \ j = 1, 2, \ldots, p \), with \( \sum_i (a_i^*)^2 = \sum_j (b_j^*)^2 = 1, \) and the elements \( X_{ij} \) are independently distributed. Then there exists a best fitting parameter given by \( \theta^g = (\lambda^*, \{a^*_i\}_{i=1}^{n}, \{b^*_j\}_{j=1}^{p}, (\sigma^*)^2) \). Moreover, this parameter is unique up to the magnitude of each of its coordinate.

The uniqueness of the parameter provided in Theorem 8 is only up to the magnitude of each elements of the parameter vector \( \theta \), hence if both \( \theta_1 = (\lambda_1, \{a_{1i}\}_{i=1}^{n}, \{b_{1j}\}_{j=1}^{p}, \sigma_1^2) \) and \( \theta_2 = (\lambda_2, \{a_{2i}\}_{i=1}^{n}, \{b_{2j}\}_{j=1}^{p}, \sigma_2^2) \) are best fitting parameters for the normal setup given in Theorem 8, then we must have \( |\lambda_1| = |\lambda_2|, |a_{1i}| = |a_{2i}|, |b_{1j}| = |b_{2j}| \) and \( \sigma_1^2 = \sigma_2^2 \). Hence, we shall restrict our attention to only the subset of the parameter space \( \Theta \) where each of these coordinates are positive, so that the uniqueness of the “best” fitting parameter can be asserted. Thus, we redefine our parameter space as \( \Theta = [0, \infty) \times S_n^+ \times S_p^+ \times [0, \infty) \), where \( S_n^+ \) and
\[ S^+_n = \{(x_1, \ldots, x_n) : \sum_{i=1}^{n} x_i^2 = 1, x_i \geq 0, i = 1, \ldots, n\} \]
\[ S^+_p = \{(x_1, \ldots, x_p) : \sum_{i=1}^{p} x_i^2 = 1, x_i \geq 0, i = 1, \ldots, p\} \]

An immediate corollary of Theorem 8 is a similar result for a deterministic data matrix \( X \).

**Corollary 1.** Suppose that the data matrix \( X \) is of rank 1 such that \( X_{ij} = \lambda^* a^*_i b^*_j \) for all \( i = 1, 2, \ldots, n \), \( j = 1, 2, \ldots, p \), with \( \sum_i (a^*_i)^2 = \sum_j (b^*_j)^2 = 1 \). Then there exists a best fitting parameter given by \( \theta^g = (\lambda^*, \{a^*_i\}_{i=1}^{n}, \{b^*_j\}_{j=1}^{p}, 0) \). Moreover, this parameter is unique up to the magnitude of each of its coordinate.

In the deterministic setup given in Corollary 1, since the true distribution is a degenerate distribution at \( x_{ij} = \lambda^* a^*_i b^*_j \), it follows that
\[ \int V(x; c, d, \sigma^2) g_{ij}(x) dx = V(\lambda^* a^*_i b^*_j; c, d, \sigma^2). \]
Therefore, the population version of the alternating equations given in Eq. (3.5)-(3.7) becomes the same as the iteration formulas given in Eq. (3.1)-(3.3). In other words, \( \theta^g = (\lambda^*, \{a^*_i\}_{i=1}^{n}, \{b^*_j\}_{j=1}^{p}, 0) \) becomes the unique fixed point of the alternating iteration rules, in the restricted parameter space \( \Theta = [0, \infty) \times S^+_n \times S^+_p \times [0, \infty) \). For fixed \( n \) and \( p \), if we assume the convergence of the iteration rules (which can be assured later using Theorem 13), then it follows that the converged estimator will be a fixed point of the iteration rules (Eq. (3.1)-(3.3)). Since the fixed point is unique, hence the converged estimator will also be unique in this particular case.

An equivariance property satisfied by the usual singular value decomposition is that whenever the data matrix is multiplied by some scalar quantity, the singular values are also multiplied by the same scalar. However, the singular vectors remain unchanged. The following theorem presents this equivariance property for a "best" fitting parameter.

**Theorem 2.** If \( \theta^g = (\lambda^g, \{a^g_i\}_{i=1}^{n}, \{b^g_j\}_{j=1}^{p}, (\sigma^g)^2) \) is a "best" fitting parameter for the matrix \( X \), then for any real constant \( c \), \( \tilde{\theta}^g = (c\lambda^g, \{c a^g_i\}_{i=1}^{n}, \{b^g_j\}_{j=1}^{p}, (c\sigma^g)^2) \) is a "best" fitting parameter for the matrix \( cX \).

Another equivariance property of the usual singular value decomposition is that under any row (or column) permutation of the data matrix, the entries of the left (or right) singular vectors permute accordingly, while the singular values remain unaffected. Such a property also holds for a "best" fitting parameter. Let \( \pi_R \) and \( \pi_C \) denote some such permutation on the row and column indices of the matrix respectively.

**Theorem 3.** If \( \theta^g = (\lambda^g, \{a^g_i\}_{i=1}^{n}, \{b^g_j\}_{j=1}^{p}, (\sigma^g)^2) \) is a "best" fitting parameter for the matrix \( X \), then a "best" fitting parameter for the matrix \( PXQ^T \) where \( P, Q \) are permutation matrices corresponding to the permutations \( \pi_R \) and \( \pi_C \), is the transformed parameter \( \tilde{\theta}^g = (\lambda^g, \{a^g_{\pi_R(i)}\}_{i=1}^{n}, \{b^g_{\pi_C(j)}\}_{j=1}^{p}, (\sigma^g)^2) \).
Theorem 9 and Theorem 10 provide indications of similarities between the usual singular value and singular vectors, and a “best” fitting parameter defined in Definition 1. In other words, a “best” fitting parameter can also be regarded as a proxy of singular value, as both share the same desirable equivariance properties. Also, in view of Theorem 8, the “best” fitting parameter turns out to be exactly the same as the singular value for the normal distribution.

3.2. Convergence of the rSVDdpd algorithm. The iteration formulas given in Eq. (3.1)-(3.3) are essentially obtained from the particular need to solve the empirical versions of the minimization problems shown in Eq. (3.5), Eq. (3.6) and Eq. (3.7) subject to the most current estimate of those parameters in an alternating fashion. In connection to these iteration rules of the proposed rSVDdpd method in Eq. (3.1)-(3.3), and a “best” fitting parameter \( \theta^* \) as in Definition 1, it is necessary to know if the iteration converges to some estimate, and whether that estimate is close to a “best” fitting parameter. These two questions are answered through two main results of this paper. The first of the pair claims that under some reasonable assumption about the parameter space, the estimate given by iterations in Eq. (3.1)-(3.3) converges as \( t \to \infty \). Before stating the necessary assumptions and the theorem, we rewrite the objective function \( H_{n,p} \) as a function of \( \theta \), denoted by \( H_{n,p,\theta} \), as follows

\[
H_{n,p}(\theta) = H_{n,p}(\lambda, \{a_i\}_{i=1}^n, \{b_j\}_{j=1}^p, \sigma^2) = \frac{1}{np} \sum_{i=1}^n \sum_{j=1}^p V(x_{ij}; \lambda a_i, b_j, \sigma^2)
\]

\[
= \frac{1}{np} \sum_{i=1}^n \sum_{j=1}^p V(x_{ij}; a_i, \lambda b_j, \sigma^2).
\]

The assumptions necessary for proving the convergence of the algorithm can be expressed as

(A1) The initial values are chosen in such a way that the estimates \( \lambda^{(t)} \) and \( \sigma^{(t)} \) for any iteration \( t \) are uniformly bounded in the interval \([0, M]\).

(A2) There is a unique minimizer \( \theta^* \) of \( H_{n,p} \) in a compact set \( \Theta \subseteq [0, M] \times S_n^+ \times S_p^+ \times [0, M] \), where \( S_n^+ \) and \( S_p^+ \) are as defined in Eq. (3.8)-(3.9). That means, there is a unique \( \theta^* \in \Theta \) such that \( H_{n,p}(\theta^*) = \min_{\theta \in \Theta} H_{n,p}(\theta) \).

**THEOREM 4.** Consider fixed \( n \) and \( p \). Under Assumptions (A1)-(A2), the sequence of estimates \( \theta^{(t)} = (\lambda^{(t)}, \{a_i^{(t)}\}_{i=1}^n, \{b_j^{(t)}\}_{j=1}^p, (\sigma^2)^{(t)}) \) obtained through Eq. (3.1)-(3.3) converge to the minimizer \( \theta^* \) of \( H_{n,p} \) provided that the initial starting value of the parameter \( \theta \) in the algorithm, i.e., \( \theta^{(0)} \), belongs to the compact set \( \Theta \).

Assumptions (A1)-(A2) are not restrictions on the probability distribution of the elements of the data matrix \( X \), but rather constraints on the elements of the matrix \( X \) as fixed quantities. Both of these assumptions can be enforced in the rSVDdpd algorithm itself, by carefully choosing the starting value \( a^{(0)} \) and \( b^{(0)} \). In other words, during any stage of the iteration, if the estimates of singular value \( \lambda^{(t)} \) show a monotonic pattern with increasing magnitude of slope over a period of many iterations, that particular iteration could be abandoned and a new starting point may be chosen. One automatic choice of this initial estimate is the singular values and vectors estimated by usual SVD on the transformed matrix \( \tilde{X} \) where the elements are given by

\[
\tilde{x}_{ij} = \begin{cases} 
Q_{\beta} & \text{if } x_{ij} < Q_{\beta} \\
x_{ij} & \text{if } Q_{\beta} < x_{ij} > Q_{(1-\beta)} \\
Q_{(1-\beta)} & \text{if } x_{ij} > Q_{(1-\beta)} 
\end{cases},
\]
where \( x_{ij} \) is the \((i, j)\)-th element of the data matrix \( X \) for which robust SVD is to be computed, and, \( Q_\beta \) and \( Q_{(1-\beta)} \) are respectively the \( \beta \)-th and \((1 - \beta)\)-th sample quantiles of all elements \( x_{ij} \), with typical values of \( \beta \) ranging from 0.05 to 0.25, based on the expected level of contamination.

3.3. Properties of the converged rSVDdpd estimate. As in the case of the equivariance properties of a “best” fitting parameter mentioned in Theorems 9 and 10, the converged rSVDdpd estimate also satisfies such equivariance properties. Theorems 11 and 12 establish these facts, based on the assumption that the initial values of the iteration procedures also obey the equivariance properties.

**Theorem 5.** Let \( \theta^* = (\lambda^*, \{a_i^*\}_{i=1}^n, \{b_j^*\}_{j=1}^p, (\sigma^*)^2) \) be the converged rSVDdpd estimator of \( X \), starting from \( \theta^{(0)} = (\lambda^{(0)}, \{a_i^{(0)}\}_{i=1}^n, \{b_j^{(0)}\}_{j=1}^p, (\sigma^{(0)})^2) \). Then, starting with the new initial estimate \( (c\lambda^{(0)}, \{a_i^{(0)}\}_{i=1}^n, \{b_j^{(0)}\}_{j=1}^p, (c\sigma^{(0)})^2) \), the rSVDdpd estimator for the data matrix \( cX \) converges to \((c\lambda^*, \{a_i^*\}_{i=1}^n, \{b_j^*\}_{j=1}^p, (c\sigma^*)^2)\).

**Theorem 6.** Let \( \theta^* = (\lambda^*, \{a_i^*\}_{i=1}^n, \{b_j^*\}_{j=1}^p, (\sigma^*)^2) \) be the converged estimator of singular value of \( X \) obtained by rSVDDpd, starting from the initial estimate \( \theta^{(0)} = (\lambda^{(0)}, \{a_i^{(0)}\}_{i=1}^n, \{b_j^{(0)}\}_{j=1}^p, (\sigma^{(0)})^2) \). Also, let \( P \) and \( Q \) be the permutation matrices corresponding to the permutations \( \pi_R \) and \( \pi_C \) respectively. Then, starting with the new initial estimate \( (\lambda^{(0)}, \{a_{\pi_R(i)}^{(0)}\}_{i=1}^n, \{b_{\pi_C(j)}^{(0)}\}_{j=1}^p, (\sigma^{(0)})^2) \), the rSVDDpd estimator for the data matrix \( PXQ^T \) converges to \((\lambda^*, \{a_{\pi_R(i)}^*\}_{i=1}^n, \{b_{\pi_C(j)}^*\}_{j=1}^p, (\sigma^*)^2)\).

Consistency is one of the basic desirable large sample properties of an estimate. Before proving the consistency of the converged rSVDDpd estimate \( \theta^* \) (whose existence is given by Theorem 13), one should note that the converged estimate is constrained to satisfy \( \sum_i (a_i^*)^2 = \sum_j (b_j^*)^2 = 1 \). Thus, there are two primary difficulties. First, the dimension of the parameter space \( \Theta \) is \((n + p + 2)\), which increases to infinity as the dimension of the matrix (i.e., \( n \) or \( p \)) tends to infinity in the notion of consistency of the estimators. Secondly, the parameter space \( \Theta \) is not necessarily convex due to the presence of the coordinates related to singular vectors, as the linear combination of two unit length vectors is not necessarily of unit length. Hence, the inverse stereographic projection is employed in order to transform this non-convex parameter space \( \Theta \) into a convex parameter space \( \Xi \) which is a subset of \( \mathbb{R}^{(n+p)} \). We call this parameter space \( \Xi \) as the natural parameter space in the given setup. The one-one transformation \( \mathcal{T} \) between these two parameter spaces \( \Theta \) and \( \Xi \) are governed by the following two equations:

\[
\mathcal{T}(\lambda, \{a_i\}_{i=1}^n, \{b_j\}_{j=1}^p, \sigma^2) = \left( \lambda, \left\{ \frac{a_i}{1 - a_n} \right\}_{i=1}^{n-1}, \left\{ \frac{b_j}{1 - b_n} \right\}_{j=1}^{p-1}, \sigma^2 \right),
\]

and,

\[
\mathcal{T}^{-1} \left( \lambda, \{a_i\}_{i=1}^{n-1}, \{b_j\}_{j=1}^{p-1}, \sigma^2 \right) = \left( \lambda, \left\{ \frac{2\alpha_i}{U^2 + 1} \right\}_{i=1}^{(n-1)} \left( \frac{U^2 - 1}{U^2 + 1} \right)^{(n-1)} \left( \frac{2\beta_j}{V^2 + 1} \right)_{j=1}^{(p-1)} \left( \frac{V^2 - 1}{V^2 + 1} \right)^{(p-1)} \sigma^2 \right),
\]

where \( U^2 = \sum_{i=1}^{n-1} \alpha_i^2 \) and \( V^2 = \sum_{j=1}^{p-1} \beta_j^2 \). Accordingly, we denote \( \eta \) as an element of this natural parameter space \( \Xi \), where the corresponding transformed parameter \( \theta = \mathcal{T}^{-1}(\eta) \)
denotes an element of $\Theta$. Several useful smoothness properties of this stereographic projection has been discussed in Section S2 of the Supplementary Material.

Now that the necessary foundations are laid out, we can present the second main theorem of the paper. This theorem claims that with some reasonable assumptions, the minimizer $\theta^*$ of $\mathcal{H}_{n,p}$ as given in Theorem 13, is a consistent estimator of the best fitting parameter $\theta^0$. The uniqueness of the best fitting parameter in this setup is a consequence of Theorem 8 following from the assumption of normality of the true data generating distribution. However, note that the description of “best” fitting parameter indicated in Definition 1 is applicable for fixed $n$ and $p$. In contrast, the statistical consistency is an asymptotic property, which requires the dimensions of the matrix $n$ and $p$ to tend towards infinity. To resolve this conflict in an unified setup, we assume that a sequence of “best” fitting parameter exists for each fixed $n$ and $p$. We denote the $(i,j)$-th entry of the data matrix $X_{n,p}$ of order $n \times p$ by the random variable $(X_{ij})_{n,p}$, and make several assumptions about it as follows

(B1) There exists a sequence of best fitting parameters $\theta^n_{n,p} = (\lambda^n, \{a^n_{i,n}\}_{i=1}^n, \{b^n_{j,p}\}_{j=1}^p, (\sigma^n_{n,p})^2)$ for each set of positive integers $n$ and $p$. Moreover, assume that for any fixed positive integers $n$ and $p$, the random variables $(X_{ij})_{n,p}$ are independent for $i = 1, \ldots, n; j = 1, \ldots, p$, and they follow a normal distribution with mean $\lambda^n a^n_{i,n} b^n_{j,p}$ and variance $(\sigma^n_{n,p})^2$.

(B2) For each set of positive integers $n$ and $p$, there exists a set $S_{n,p}$ inside $\Theta_{n,p} \subset \mathbb{R}^{(n+p+2)}$ such that it does not contain any point $\theta$ of the form $(\lambda, a, b, \sigma^2)$ with either $a = (0, \ldots, 0, 1)$ or $b = (0, \ldots, 0, 1)$ and its image under the stereographic transformation $\tau$ is an open rectangle in $\Xi_{n,p} \subset \mathbb{R}^{(n+p)}$.

(B3) The converged rSVDdpd estimate for the data matrix $X_{n,p}$, i.e., $\theta^*_{n,p}$ (the minimizer of $\mathcal{H}_{n,p}$ as indicated by Theorem 13) satisfy $(a^n_1)^T \neq (0, \ldots, 0, 1)$ and $(b^n_p)^T \neq (0, \ldots, 0, 1)$.

(B4) The error variances in best fitting parameters satisfy $(\sigma^n_{n,p})^2 = \Theta((np)^{-2/(\alpha+2)})$ for some $0 \leq \alpha \leq 1$. This means, there exists positive finite constants $\gamma_1, \gamma_2$ such that

$$\frac{\gamma_1}{(np)^{1/(\alpha+2)}} < \sigma^n_{n,p} < \frac{\gamma_2}{(np)^{1/(\alpha+2)}}$$

for sufficiently large $n$ and $p$.

(B5) There exists a positive finite constant $M$ independent of $n$ and $p$ such that

$$\mathbb{P}\left(\max_{1 \leq i \leq n, 1 \leq j \leq p} |(X_{ij})_{n,p}| < M\right) \rightarrow 1$$

as $n$ and $p$ both tends to infinity.

The first assumption (B1) is simply a description of the setup. This assumption along with normality assumption allows one to perform differentiation under the integral sign and allows the interchange of differentiation and expectation operators, which is quite common. The assumptions (B2) and (B3) ensure that the natural parametrization based on the stereographic projection described above remains valid. Assumption (B4) is a very particular assumption for this setup, which is not connected to the usual assumptions of MDPD estimation mentioned in Ghosh and Basu (2013). It is well known from random matrix theory that Gaussian ensemble with each entries following a standard normal distribution has the singular values asymptotically at the order of $(\sqrt{n} + \sqrt{p})$ (Mehta, 2004; Tracy and Widom, 1993). However, since the assumption (B1) indicates that the same $\lambda^n$ acts as the singular value for any $n$ and $p$, the variance of the entries of the data matrix has to go down asymptotically to ensure that this singular value does not grow asymptotically with increase in $n$ and $p$. Finally, assumption (B5) is, in spirit, similar to the assumption (A2), which can be enforced in the algorithm itself by scaling the entries of the data matrix within a compact set and then scaling back the estimated singular values accordingly.
THEOREM 7. Under Assumptions (B1)-(B5), the rSVDdpd estimator \( \theta_{n,p}^* \) corresponding to the same \( \alpha \) from assumption (B4), is a consistent estimator of the sequence of the “best” fitting parameter values \( \theta_{n,p}^{\eta} \) as both dimensions of the data matrix \( X_{n,p} \) (i.e., \( n \) and \( p \)) tend to infinity. In other words, \( (\theta_{n,p}^* - \theta_{n,p}^{\eta}) \to 0 \) in probability as both \( n \to \infty \) and \( p \to \infty \) subject to a constant ratio in limit, i.e., \( \lim_{n \to \infty} \frac{n}{p} = c \) with \( 0 < c < \infty \).

REMARK 4. Since we have derived rSVDdpd as an extension of MDPDE in linear regression setup based on the work of Ghosh and Basu (2013), it is natural to think that the result on the consistency of the MDPDE given in Theorem 3.1 of Ghosh and Basu (2013) can be imitated to deliver a proof of the consistency of the proposed rSVDdpd estimators. However, there are several major complications involved.

1. The basic assumption required for consistency of MDPDE in INH (independent and non-homogeneous) setup is existence of an open set in the parameter space \( \Theta \). However, this parameter space \( \Theta \) by itself does not contain any open neighbourhood. Therefore, all the necessary formulations are required to be applied on the natural parametrization \( \eta \in \Xi \) instead, which converts the setup into a nonlinear regression problem.

2. In case of SVD, the length of the singular vectors is not fixed and grows with the dimension of the matrix. Thus, as \( n, p \to \infty \), the dimension of the parameter space also increases to infinity. This problem had considerable interest from many authors under a general M-estimation setup (Huber, 1973; Portnoy et al., 1984). Almost all of these results assumes convexity of the objective function (He and Shao, 2000), but that cannot be employed in our case, because the objective function is convex in any of the parameters individually when the other parameters are kept fixed, but becomes non-convex if all the parameters are taken together.

3. In each of the alternating iterations, the consistency ensures that the resulting estimator based on the minimization of that particular iteration is probabilistically close to the minimizer of the population version of the iterative equation. However, the population version of the iterative equation depends on the current estimates of the other parameters. Hence, in each of the iterations, the empirical estimates are allowed to deviate from the population estimates and these errors sum up as the number of iterations increase. Hence, a bound on the tail of the distribution of such sum of small errors are needed to ensure that consistency holds throughout.

Thus, an extensive non-trivial modification of the existing proof technique is required. Due to its length and complications, the proof of this theorem is given in Section S3 of the Supplementary Material.

REMARK 5. A particularly interesting observation in view of assumption (B4) is that when \( \alpha = 0 \), the usual MLE of singular values are consistent if the error variance decreases at the order of \( \frac{1}{np} \). However, for higher values of \( \alpha \), the error variance needs to decrease at a slower rate, only at the order of \( \frac{1}{(np)^{\frac{2}{2(\alpha+2)}}} \).

An immediate corollary of Theorem 13 and Theorem 14 establishes the correctness of our algorithm rSVDdpd, at least in a probabilistic setup.

COROLLARY 2. Under the Assumptions (A1)-(A2) and (B1)-(B5), when the true data generating distribution is a normal distribution as mentioned in Theorem 14, the rSVDdpd algorithm outlined in Algorithm 1 converges as the number of iterations goes to infinity, and the converged estimator is a consistent estimator of the best fitting parameter (uniqueness follows from Theorem 8) as both the dimensions of the matrix increases to infinity subject to the restriction of constant ratio in limit i.e., \( \lim_{p \to \infty} \frac{n}{p} = c \) with \( 0 < c < \infty \).
Remark 6. Theorems 11 and 12 are, respectively, the empirical counterparts of Theorems 9 and 10. In view of Theorems 11 and 12, the equivariance properties hold given that the initial values of the iterations of rSVDdpd also satisfy such equivariant properties. However, under the Assumptions (A1)-(A2) and (B1)-(B5), the convergence of the rSVDdpd algorithm is immediate and by consistency of this converged estimator, it follows that for large \( n \) and \( p \), the converged estimator can be made arbitrarily close to the true “best” fitting parameters. Since these “best” fitting parameters obey equivariance properties as assured in Theorems 9 and 10, it follows that for large \( n \) and \( p \), the converged estimator will also approximately satisfy these equivariance properties, irrespective of the equivariance of starting values.

4. Numerical Illustrations: Simulation Studies. In this section, we compare the performance of our method rSVDdpd with two existing robust SVD methods, namely the ones proposed by Hawkins, Liu and Young (2001) and Zhang, Shen and Huang (2013). Implementation of the robust SVD algorithm proposed by Hawkins, Liu and Young (2001) (to be referred to here as pcaSVD) is available as an R package pcaMethods (Stacklies et al., 2007), which outputs all singular values and vectors of the input data matrix. The second algorithm by Zhang, Shen and Huang (2013) obtains the first pair of singular vectors based on the minimization procedure

\[
(\hat{u}, \hat{v}) = \arg \min_{(u, v)} \left[ \rho \left( \frac{X - uv^T}{\sigma} \right) + P_\lambda(u, v) \right]
\]

where \( \rho(\cdot) \) is a robust loss function (namely Huber’s loss function) and \( P_\lambda \) is a regularization penalty term to motivate smoothing in the entries of the singular vectors. For an extensive comparison, we consider two variants of this algorithm. In one variation, we perform the minimization with only the Huber’s loss function without any penalty term, which we shall refer to as RobSVD, while in the other variation, we follow the recommended procedure of minimization with penalty term, which we shall call RobRSVD. Implementation of both of these variants is available in the R package RobRSVD (Zhang and Pan, 2013) which is programmed only to output the first singular value and its corresponding singular vectors. Thus, in order to compare performances of the algorithms on an equal footing, we add a wrapper outside this function to apply the same algorithm on the residual matrix to output the subsequent singular values (see Section 2 and (Hawkins, Liu and Young, 2001) for details). Along with these, we also consider an implementation of the usual \( L_2 \) norm minimization based SVD procedure available in R base package (R Core Team, 2021) written using LAPACK FORTRAN library. To aid the simulation procedure of performance comparison, we have made available an R package rsvddpd in github\(^1\), and all of the simulation studies have been done with the help of this package.

4.1. Performance Measures. For comparative illustration of the performances of our proposed rSVDdpd approach, we use a Monte Carlo method, where for each resample, we add randomly generated errors to the true matrix, and then perform the desired robust SVD algorithm, and track each of the estimates obtained from each resample. Let, \( \hat{\lambda}_{ib} \) be the estimated \( i \)-th singular value at the \( b \)-th resample. Then, as a measure of accuracy based on \( B \) Monte Carlo resamples, we use the MSE and the bias of the estimate given as

\[
\text{MSE}_i = \frac{1}{B} \sum_{b=1}^{B} \left( \hat{\lambda}_{ib} - \lambda_i \right)^2, \quad \text{Bias}_i = \frac{1}{B} \sum_{b=1}^{B} \hat{\lambda}_{ib} - \lambda_i, \quad i = 1, 2, \ldots \min\{n, p\},
\]

\(^1\)https://github.com/subroy13/rsvddpd
where $\lambda_i$ is the true $i$-th singular value which was used to construct the data matrix and $n$ and $p$ are, respectively, the number of rows and columns of the data matrix. Also, we compute a total score for both of these measures considering the sum of MSEs and the sum of squared biases for the estimates of all singular values.

To compare the left and right singular vectors, we define a dissimilarity score between two normalized vectors $u$ and $v$ as

$$\text{dissimilarity score}(u, v) = 1 - |\langle u, v \rangle|,$$

where $|x|$ denotes the absolute value of $x$, and $\langle u, v \rangle$ is the usual euclidean inner product between two vectors $u$ and $v$. Since $u$ and $v$ are both normalized, this score is equal to 0 if either $u = v$ or $u = (-v)$, and is equal to 1 if $u$ and $v$ are orthogonal to each other. As a performance measure, we consider the average dissimilarity score between the estimated singular vectors and the true singular vectors, averaged over all Monte Carlo resamples as

$$\text{(Average dissimilarity score)}_i = \frac{1}{B} \sum_{b=1}^{B} \text{dissimilarity score}(u_i, \hat{u}_{ib}).$$

Here, $\hat{u}_{ib}$ denotes the estimated $i$-th left (or right) singular vector for the $b$-th resample and $u_i$ denotes the true $i$-th singular vector. As before, a total dissimilarity score is also calculated as the sum of these average dissimilarity scores given by Eq. (4.2) for all estimated singular vectors.

However, it should be noted that all of the robust SVD algorithms under consideration, i.e., the two existing algorithms of robust SVD, namely, pcaSVD (Stacklies et al., 2007), RobRSVD (Zhang and Pan, 2013), and the proposed algorithm rSVDDpd are iterative algorithms; hence they might not converge within a reasonable amount of time for some of the Monte Carlo resamples due to poor choice of the initial estimates. In particular, Assumption (A1) in Theorem 13 is found to be violated in a few of the Monte Carlo resamples where the sequence of singular value estimates in the iterations seem to diverge. Schur (1911) provided a simple upper bound on the largest singular value of a matrix as

$$\lambda_{\max}(X) \leq \sqrt{\|X\|_1 \|X\|_\infty},$$

where $\lambda_{\max}(X)$, $\|X\|_1$, $\|X\|_\infty$ are the largest singular value, $L_1$-matrix norm and $L_\infty$-matrix norm of the matrix $X$. We verify this bound for each resample to assess the non-convergence, and consider only those resamples among $B = 5000$ resamples where the estimated singular value from all algorithms satisfy inequality (4.3). Thus, the performance measures described in Eq. (4.1) and Eq. (4.2) are computed based on those valid resamples for every algorithm under consideration.

### 4.2. Simulation Settings

In order to construct the true data matrix $X$ from a pre-specified singular value decomposition, we use the coefficients of the first three orthogonal polynomial contrasts of order 10 as the left singular vectors, coefficients of first three orthogonal polynomial contrasts of order 4 as the right singular vectors and use the three nonzero singular values as 10, 5 and 3. The resulting data matrix then turns out to be

$$X = U \begin{bmatrix} 10 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 3 \end{bmatrix} V^T,$$

where
$U = \begin{bmatrix}
-0.49543369 & 0.52223297 & -0.4534252 \\
-0.38533732 & 0.17407766 & 0.1511147 \\
-0.27524094 & -0.08703883 & 0.3778543 \\
-0.16514456 & -0.26111648 & 0.3346710 \\
-0.05504819 & -0.34815531 & 0.1295501 \\
0.05504819 & -0.34815531 & -0.1295501 \\
0.16514456 & -0.26111648 & -0.3346710 \\
0.27524094 & -0.08703883 & -0.3778543 \\
0.38533732 & 0.17407766 & -0.1511147 \\
0.49543369 & 0.52223297 & 0.4534252
\end{bmatrix}$

\[V = \begin{bmatrix}
-0.6708204 & 0.5 & -0.2236068 \\
-0.2236068 & -0.5 & 0.6708204 \\
0.2236068 & -0.5 & -0.6708204 \\
0.6708204 & 0.5 & 0.2236068
\end{bmatrix} \).

To create each Monte Carlo resample, we add errors distributed according to some pre-specified distribution corresponding to each of the entries of the matrix $X$, and for each such resample, both the usual SVD and the robust algorithms for computing singular value decomposition are applied, and the summary measures described in Section 4.1 are computed to study their performances.

We simulate $B = 5000$ Monte Carlo resamples, and due to convergence issues mentioned in Section 4.1, only those resamples where all algorithms converge is taken into consideration to obtain the discrepancy measures. This allows the comparison of these summary measures based on the same number of resamples mitigating convergence issues. Our simulation scenarios are broadly divided into 5 categories denoted by (S1)-(S5).

(S1) The errors are distributed according to a standard normal distribution $N(0,1)$ (no outliers per se).

(S2) The errors are distributed according to a contaminated standard normal distribution namely

$e_{ij} \sim (1 - \epsilon)N(0,1) + \epsilon \delta_{25},$

where $\epsilon$ is the amount of contamination and $\delta_{25}$ denotes the degenerate distribution at 25. Based on the amount of contamination, we consider three sub-cases of this simulation setting.

(S2a) $\epsilon = 0.05$, denotes only 5% contamination, which corresponds to a relatively light amount of outlying observations.

(S2b) $\epsilon = 0.1$, denoting medium level contamination with the presence of 10% outlying values.

(S2c) $\epsilon = 0.15$, denoting heavy contamination with approximately 15% of the entries being same as the outlying observation of 25.

(S3) The errors are distributed according to a standard normal distribution with a block-based contamination as presented in Zhang, Shen and Huang (2013). Here, in each resample, the errors are generated from standard normal distribution and then a $2 \times 2$ block is randomly chosen to substitute the corresponding errors by the outlying observation of 25.

(S4) The errors are distributed according to a standard Cauchy distribution. This setup helps us to study the effect of heavy tailed errors in the robust estimation of SVD.

(S5) The errors are distributed according to a standard lognormal distribution, which is used to study the effect of an asymmetric error distribution with only positive support.

4.3. Simulation Results. Table 1 summarizes the comparative results of the usual SVD method, the three existing robust SVD algorithms (pcaSVD (Stacklies et al., 2007) and two variants of RobRSVD (Zhang and Pan, 2013)) and the proposed rSVDdpd for different
choices of robustness parameter $\alpha$, based on the performance measures mentioned in Section 4.1, for different simulation setups (S1)-(S5). For the sake of brevity, the detailed results on each singular value and vectors for each of the simulation setting are provided in Section S5 of the Supplementary Material.

Table 1

| Simulation Setup | Measure          | Existing methods for computing SVD | Choice of $\alpha$ in rSVDdpd | α = 0.1 | α = 0.3 | α = 0.5 | α = 0.7 | α = 1 |
|-----------------|------------------|-----------------------------------|------------------------------|---------|---------|---------|---------|-------|
|                 |                  | Usual SVD            | pcaSVD                     | RobSVD | RobRSVD | RobRSVD | RobRSVD |       |
| S1              | Sq. Bias         | 7.957                | 15.066                      | 8.808   | 1.684   | 7.952   | 7.94    | 7.925 |
|                 | Total MSE        | 10.456               | 24.242                      | 11.608  | 6.865   | 10.455  | 10.473  | 10.553 |
|                 | Diss (left)      | 0.701                | 1.198                       | 0.799   | 0.733   | 0.702   | 0.707   | 0.717 |
|                 | Diss (right)     | 0.418                | 0.968                       | 0.52    | 0.514   | 0.419   | 0.425   | 0.433 |
| S2a             | Sq. Bias         | 519.35               | 350.402                     | 523.859 | 17.672  | 294.047 | 18.138  | 10.877 |
|                 | Total MSE        | 729.83               | 793.612                     | 748.043 | 278.585 | 622.819 | 91.894  | 44.016 |
|                 | Diss (left)      | 1.93                 | 1.679                       | 1.933   | 1.324   | 1.613   | 0.944   | 0.896 |
|                 | Diss (right)     | 1.529                | 1.286                       | 1.516   | 1.074   | 1.163   | 0.615   | 0.576 |
| S2b             | Sq. Bias         | 2434.021             | 2110.99                     | 2488.98 | 193.664 | 662.604 | 2106.45 | 196.514 |
|                 | Total MSE        | 2712.69              | 2782.59                     | 2803.95 | 617.094 | 1453.404 | 938.705 | 621.155 |
|                 | Diss (left)      | 2.206                | 2.11                        | 2.203   | 1.932   | 1.666   | 1.444   | 1.363 |
|                 | Diss (right)     | 1.73                 | 1.633                       | 1.688   | 1.598   | 1.651   | 1.227   | 1.042 |
| S2c             | Sq. Bias         | 1677.949             | 1640.708                    | 1679.881| 1000.284| 376.85  | 210.34  | 178.10 |
|                 | Total MSE        | 1686.361             | 1654.5                      | 1688.708| 1248.276| 1634.409| 458.468 | 156.986 |
|                 | Diss (left)      | 2.052                | 2.003                       | 1.941   | 2.162   | 2.013   | 1.208   | 1.024 |
|                 | Diss (right)     | 1.924                | 1.836                       | 1.832   | 2.175   | 1.844   | 0.895   | 0.692 |
| S3              | Sq. Bias         | 41825.788            | 14224.145                   | 41779.81| 540.799 | 469.522 | 265.135 | 198.809 |
|                 | Total MSE        | 2171697.037          | 2163773.363                 | 2171711.033| 29149.731| 1629.485| 1197.881| 859.039 |
|                 | Diss (left)      | 2.089                | 1.989                       | 2.095   | 1.707   | 1.97    | 1.877   | 1.838 |
|                 | Diss (right)     | 1.603                | 1.489                       | 1.602   | 1.303   | 1.496   | 1.403   | 1.367 |
| S4              | Sq. Bias         | 93.633               | 77.499                      | 94.901  | 29.135  | 69.98   | 67.092  | 62.429 |
|                 | Total MSE        | 146.187              | 108.514                     | 149.13  | 49.046  | 85.102  | 83.272  | 78.83 |
|                 | Diss (left)      | 2.02                 | 2.034                       | 1.969   | 1.967   | 1.968   | 1.954   | 1.939 |
|                 | Diss (right)     | 1.785                | 1.81                        | 1.734   | 1.824   | 1.754   | 1.744   | 1.734 |

As shown in Table 1, the usual SVD generally leads to a biased estimator of the singular values for Gaussian errors which is also supported in well-established theory (Rudelson and Vershynin, 2010). For $\alpha = 0$, the performance of rSVDdpd is largely similar to the usual SVD (see Section S5 of Supplementary Material), because as $\alpha \to 0$, the minimum density power divergence estimation problem becomes same as the usual least squares minimization problem, as indicated in Remark 1. For the simulation setup (S1), pcaSVD algorithm turns out to most biased, followed by RobSVD, both of which have more bias and MSE than the usual SVD algorithm. Compared to the usual SVD, the proposed rSVDdpd achieves lesser bias as the robustness parameter $\alpha$ increases, but at the cost of higher variance and MSE. RobRSVD achieves the minimum bias and MSE in this scenario, but it shows a higher variance as well as a higher dissimilarity score than the proposed rSVDdpd algorithm.

Turning our attention to simulation setups (S2a), (S2b) and (S2c), we see that the usual SVD and existing robust SVD algorithms pcaSVD and RobSVD do not yield very reliable estimates of the singular values. Although RobRSVD provides reasonable estimates rSVD-dpd achieves lower bias and MSE for some choices of $\alpha$. In presence of random outlying observations, as in the case of simulation setup (S2a), (S2b) and (S2c), both the bias and MSE for rSVDdpd show reductions as the robustness parameter $\alpha$ is increased from 0 to 1. The dissimilarities of singular vectors are also found to decrease with an increase in $\alpha$. 
For the block level contamination in simulation setup (S3), we find that rSVDdpd has much better performances than the existing robust SVD algorithms in terms of all performance metrics. In presence of errors from a heavy tailed distribution as considered in simulation setup (S4), the results remain very similar. The proposed rSVDdpd algorithm provides least bias and MSE, and even with small robustness parameter \( \alpha = 0.1 \), our algorithm outperforms the existing robust SVD algorithms under consideration.

In simulation setup (S5) with lognormally distributed errors having positive support, rSVDdpd outperforms the usual SVD, pcaSVD and RobSVD methods by showing a reduction in both bias and MSE. However, as in the simulation setup (S1), RobRSVD is again found to provide estimates with least bias and MSE, but at a cost of higher variance and dissimilarity scores than rSVDdpd.

Although RobRSVD outputs better singular value estimates than rSVDdpd under normally and lognormally distributed errors, it does so at the cost of extremely high computational complexity. This is precisely due to the matrix inversion step to compute \((V^TWW^*V + 2\Omega_{u|v})^{-1}\) (see Eq. (9) of Zhang, Shen and Huang (2013)). Since the best known matrix inversion algorithm, i.e., a variant of Coppersmith-Winograd algorithm (Alman and Williams, 2021) achieves a computational complexity of \(O(n^{2.3728596})\) for inverting an \(n \times n\) matrix, it follows that each iteration of the RobRSVD algorithm has time complexity \(O(n^{2.3728596} + p^2.3728596)\). On the other hand, each iteration of rSVDdpd performs only a weighted average computation, which reduces its computational budget to \(O(n^2 + p^2)\).

To demonstrate this, we consider \(n \times p\) matrices with uniformly distributed entries for different choices of \(n\) and fixed \(p = 25\), and apply different methods of computing SVD on them. Table 2 summarizes the time taken (in units of milliseconds) to obtain the first singular value from different algorithms for different choices of \(n\), in a computer with Intel i5-8300H 2.30GHz processor with 8 GB of RAM. As seen from Table 2, the computational budget of rSVDdpd is similar to pcaSVD, which is lower by several orders of magnitude than RobSVD and RobRSVD. This extremely high computational cost of RobRSVD can be circumvented if the matrix \((V^TWW^*V + 2\Omega_{u|v})\) becomes a diagonal matrix, which happens if the penalty parameter is taken as zero and RobRSVD is reduced to its non-regularized variant RobSVD. However, as Table 1 shows, the RobSVD algorithm without the regularization cannot provide a reliable robust estimate of singular values, even using Huber’s robust loss function.

### Table 2

Summary of average time taken (in milliseconds) to obtain the first singular value and vectors of an \(n \times 25\) matrix with random entries from \(U(0, 1)\) via different SVD algorithms

| Number of rows (\(n\)) | Existing methods for computing SVD | Choice of \(\alpha\) in rSVDdpd |
|-------------------------|----------------------------------|-------------------------------|
|                         | Usual SVD | pcaSVD | RobSVD | RobRSVD | \(\alpha = 0.1\) | \(\alpha = 0.5\) | \(\alpha = 1\) |
| 5                       | 0.014     | 2.209  | 3.098  | 73.666  | 0.545      | 0.841      | 1.417      |
| 10                      | 0.011     | 4.388  | 3.966  | 99.077  | 0.975      | 1.508      | 2.100      |
| 25                      | 0.008     | 4.965  | 3.989  | 81.045  | 3.861      | 6.416      | 11.394     |
| 50                      | 0.013     | 8.519  | 7.824  | 149.4   | 4.396      | 7.104      | 12.526     |
| 100                     | 0.017     | 15.696 | 34.649 | 826.44  | 5.136      | 8.683      | 14.362     |
| 250                     | 0.026     | 32.066 | 402.125| 7839.942| 7.756      | 13.252     | 22.379     |
| 500                     | 0.041     | 35.828 | 2948.001| 54209.15| 13.622     | 21.413     | 32.404     |
| 750                     | 0.058     | 58.697 | 10363.441| 210494.564| 24.67      | 36.563     | 55.422     |
| 1000                    | 0.072     | 69.76  | 26282.893| 531362.110| 27.727     | 40.309     | 62.234     |
4.4. Choice of the Optimal Robustness Parameter. Through the extensive simulation studies, we have seen that higher values of the robustness parameter $\alpha$ tend to produce singular value estimates with smaller bias, under any kind of contamination. On the other side of the coin, the estimated singular values exhibit higher variance with an increase in $\alpha$. Therefore, it is necessary to determine an optimal choice of $\alpha$ which works equally well in practice balancing these two opposing effects.

Several authors have given procedures to select the optimal parameter $\alpha$ for the MDPD in i.i.d. situation (Hong and Kim, 2001; Warwick and Jones, 2005). Assuming $\hat{\theta}_\alpha$ to be the MDPD estimator for a specified $\alpha$, the one-step Warwick-Jones (OWJ) algorithm suggests choosing the optimum $\alpha$ by minimizing the mean squared error criterion $\mathbb{E}[(\hat{\theta}_\alpha - \theta^*)(\hat{\theta}_\alpha - \theta^*)^T]$, where $\theta^*$ is a pilot estimator. Based on extensive simulations, they proposed to take the pilot estimator as the MDPD estimate corresponding to $\alpha = 1$. Later, Ghosh and Basu (2015) extended this procedure for the INH (independent and non-homogeneous data) setup and provided extensive simulation exercise to analyze the dependence between the choice of pilot estimate $\theta^*$ and the optimal tuning parameter $\alpha$ selected by the procedure. To get rid of the dependence of this initial pilot estimate $\theta^*$, Basak, Basu and Jones (2021) has modified OWJ algorithm into an iterated WJ algorithm. In this algorithm, the obtained MDPD estimate $\hat{\theta}_\alpha$ at the optimal tuning parameter $\hat{\alpha}$ as chosen by OWJ algorithm is used as a pilot estimate at the next iteration.

While the mean squared error for the singular values can be unbounded, the mean squared error for the singular vectors are bounded, as $\|\hat{a} - a\|^2 \leq \|a\|^2 + \|a^*\|^2 + 2\max\{\|\hat{a}\|^2, \|a^*\|^2\}$ and, both $\hat{a}$ and $a^*$ are unit vectors. Therefore, application of OWJ or IWJ procedures with the usual mean squared error would bias the choice of $\alpha$ towards a minimizer of $\alpha$ minimizing only the MSE for singular values, ignoring the disparity in singular vectors, which may be minimized for a different choice of $\alpha$. Thus, we follow the idea of Huang, Shen and Buja (2009) and minimize the conditional MSE criterion

$$\text{tr}\left(\mathbb{E}\left[ (\hat{\lambda}_\alpha \hat{a}_\alpha - \lambda^* a^*) (\hat{\lambda}_\alpha \hat{a}_\alpha - \lambda^* a^*)^T | \hat{b}_\alpha \right] \right) + \text{tr}\left(\mathbb{E}\left[ (\hat{\lambda}_\alpha \hat{b}_\alpha - \lambda^* b^*) (\hat{\lambda}_\alpha \hat{b}_\alpha - \lambda^* b^*)^T | \hat{a}_\alpha \right] \right)$$

where $\hat{\lambda}_\alpha, \hat{a}_\alpha, \hat{b}_\alpha$ are respectively the rSVDdpd estimate of the first singular value and pair of first singular vectors, for robustness parameter $\alpha$. $\lambda^*, a^*$ and $b^*$ are the corresponding pilot estimates, and $\text{tr}(A)$ denotes the trace of the matrix $A$. Since conditional on $\hat{b}_\alpha$, the estimation of $\hat{\lambda}_\alpha \hat{a}_\alpha$ is same as the MDPD estimation in linear regression setup, we can express the first conditional expectation as

$$\text{tr}\left(\mathbb{E}\left[ (\hat{\lambda}_\alpha \hat{a}_\alpha - \lambda^* a^*) (\hat{\lambda}_\alpha \hat{a}_\alpha - \lambda^* a^*)^T \right] \right) = n\sigma^2 \left( 1 + \frac{\alpha^2}{1 + 2\alpha} \right)^{3/2} + \|\mathbb{E}((\hat{\lambda}_\alpha \hat{a}_\alpha) - \lambda^* a^*)\|^2_2$$

which easily follows from Theorem 6.2 of Ghosh and Basu (2013). The other conditional expectation can be expressed in a similar manner. Since all of the simulation setups show that the bias of rSVDdpd estimate is least when $\alpha = 1$, we decide to use the rSVDdpd estimate at $\alpha = 1$ as the pilot estimator. Thus, we consider the simplified criterion

$$(n + p)\hat{\sigma}^2 \left( 1 + \frac{\alpha^2}{1 + 2\alpha} \right)^{3/2} + \|\hat{\lambda}_1 \hat{a}_1\|^2_2 + \|\hat{\lambda}_1 \hat{b}_1\|^2_2$$

where $\hat{\sigma}^2$ is the normalized mean absolute deviation estimate suggested by Zhang, Shen and Huang (2013). We compute the value of this criterion for pre-defined choices of $\alpha$ and choose the particular $\alpha$ as the robustness parameter that minimizes this criterion. Note that, under no contamination, the variance term will dominate the bias term, hence $\alpha = 0$ will be chosen as the optimal robustness parameter as desired. On the other hand, if there is contamination, then the higher bias term would push the choice of the optimal $\alpha$ towards 1.
5. Video Surveillance Background Modelling. Having set up the procedure for robust SVD, we come back to the problem of our original interest. As described in Section 1, modelling the background in video surveillance is an interesting real-life problem where robust SVD can prove useful. The algorithmic task is to separate the static background of a video and the moving objects or the foreground, based on the input image frames from a surveillance video. Let us assume that the input video has $T$ frames and each frame is a grayscale image being $h$ pixels tall and $w$ pixels wide. Based on this input tensor of dimensions \((T, h, w)\) representing the video, one can construct a large matrix $M_{hw \times T}$, where the $j$-th column of the matrix $M$ is constructed with the $hw$ pixels values of the $j$-th frame of the given video. Since the subsequent frames of the video are correlated with each other precisely due to the common component of background, Candès et al. (2011) assumed a decomposition of the matrix $M = L + E$, where $L$ is a low-rank approximation (in particular, a rank one approximation) and $E$ is the relatively sparse residual part. Due to the assumption that $L$ is of low rank (or of rank one), all the columns of $L$ will be similar to each other and hence, effectively represent the static background object that does not change from one frame to another. On the other hand, the residual part $E$ is expected to recover everything that is not static, namely the moving objects in the video. As mentioned before in Section 1, due to insufficient capabilities of existing robust SVD methods, an alternative principal component pursuit is used to retrieve the low-rank approximation $L$. In this section, we use the proposed rSVDdpd instead to obtain $L$. For the demonstration of rSVDdpd based video surveillance modeling and object extraction, we choose University of Houston Camera Tampering Detection Dataset (UHCTD) (Mantini and Shah, 2019a).

With the problem of camera tampering detection (i.e., classification of tampered frames) in mind, Mantini and Shah (2019a) have compiled a comprehensive large scale dataset called UHCTD (University of Houston Camera Tampering Detection Dataset). The dataset contains surveillance videos of over 288 hours ranging across 6 days from two cameras. Three types of camera tampering method has been synthesized in the dataset; (a) covered, (b) defocused, and, (c) moved (see more details in Mantini and Shah (2019a)). These tampering are done uniformly over data to capture changing illumination between day and night. As described in Section 1, we would like to solve the problem of robust estimation of the background and the foreground content from the video frames as described above. This step is intended as a preprocessing step to aid the segmentation and classification tasks, where one may use simple thresholding algorithms (Phansalkar et al., 2011) for segmentation or classification algorithms mentioned in Mantini and Shah (2019b). For demonstration purposes, we shall consider 2 short clips from UHCTD dataset to implement the proposed rSVDdpd algorithm as well as the usual SVD algorithm. The other two robust algorithms (Hawkins, Liu and Young, 2001; Zhang, Shen and Huang, 2013) is found to be computationally challenging for such large matrices $M$. In particular, the matrix inversion step in the RobRSVD algorithm now requires the inversion of a very large matrix (namely $hw \times hw$) with a high condition number, turning the matrix into a computationally singular matrix which obstructs its implementation for the specific purpose of video background modelling. Result for two more video scene examples from the same UHCTD dataset have been demonstrated in the Supplementary Material accompanying this paper.

Figure 2 depicts the results pertaining to a scene from Camera B (Mantini and Shah, 2019a), where a noisy image is synthesized to obstruct the view of the camera. The background estimated from the usual SVD shows a clear indication of the noise even in the frames where camera tampering is not induced, and such an effect is amplified further by the presence of shadow-like artifacts in the estimated background content obtained from the usual SVD as shown in Figure 2d. In comparison, the background estimated via the proposed rSVDdpd shown in Figure 2e removes such artifacts and is less affected by the tampering.
This robustness property can also be seen in the estimated foreground contents, when the true frame is only the non-tampered background, the proposed rSVDDdpd outputs foreground content as a very dark image without any distinguishing feature as desired.

![Fig 2: The true images, estimated background and foreground by the usual SVD and the proposed rSVDDdpd for Video Scene 1](image)

Similar to the aforementioned tampered video scene, results from another video scene from Camera A (Mantini and Shah, 2019a) is shown in Figure 3. Here, the video was captured using the night-vision mode of the camera, which results in a grayscale video. As seen in Figures 3d and 3e, the estimated background content obtain by the usual SVD and the proposed rSVDDdpd differs significantly. Again, the noisy image appears in the extracted background by the usual SVD similar to a bump map, while the rSVDDdpd outputs a clearer
background. Generally, the usual SVD tends to produce a background that correlates with all frames with equal importance, while rSVDdpd tends to pick the dominant background only. Thus, any unauthorized change in the background can be identified by looking at the estimated foreground produced by rSVDdpd, namely, the frame where the estimated foreground tends to produce certain artifacts are tampered frames. If SVD is used instead to estimate the foreground, then all of the frames produces certain artifacts and thus a false positive occurs. As shown in Figures 3b and 3c, rSVDdpd outputs non-tampered foregrounds when the true frames are non-tampered, while the usual SVD results in tampered foreground contents for all frames irrespective of whether they were originally tampered or not.

Fig 3: The true images, estimated background and foreground by the usual SVD and the proposed rSVDdpd for Video Scene 2
6. Conclusion. In this paper we had set out to solve the problem of modelling video surveillance background content robustly, and we trust we have achieved that with the help of a novel algorithm for robustly performing singular value decomposition. This, we believe has been adequately demonstrated in our examples. At the same time, video surveillance background modelling is not the only domain where there is a need for applying a robust singular value decomposition technique. As described in Section 1.1, a plethora of techniques from an extensive range of disciplines use singular value decomposition as a core component of the methods. Unfortunately, with the emergence of the era of big data, it has become increasingly difficult to check or validate the authenticity, trustability and overall correctness of the data. As a result, most of the input data to these algorithms are highly susceptible to be contaminated by various forms of noises and outlying observations. Since, the standard SVD yields highly inaccurate results in presence of these outlying observations, the performance of these algorithms also degrade in presence of such contamination. To counter this problem, several robust mechanisms of computing SVD have been proposed so far (see Section 1.1), however, none of them are scalable to the large matrices encountered during different real-life applications. We trust this paper will help to fill this gap by providing a more scalable, trustworthy and efficient robust mechanism of obtaining SVD using the novel density power divergence. Also, the robustness parameter $\alpha$ in rSVDdpd can be adjusted to provide a smooth bridge between efficiency in estimation and robustness capabilities of the obtained singular value estimates (Basu et al., 1998).

Among many applications mentioned in Section 1.1 where the proposed rSVDdpd algorithm may substitute the usual SVD procedure for robust inference, one such interesting use-case is the stock-price modeling in the guise of factor analysis. In finance, the price of a stock is generally assumed to be composed of two components, a latent market risk specific to the economy of the whole country or the industry at an aggregate level, and an asset specific risk particular to the company at the individual level (Wang, 2017). Since the stock prices exhibit jumps and can change drastically within a short period of time, standard SVD often fails to capture this decomposition while robust SVD methods prove useful. A detailed example of this stock price modelling has been provided in Section S4.3 of the Supplementary material.

While the theoretical results depict various desirable properties for the algorithm and estimated singular values and vectors, the simulations corroborate the excellence of rSVDdpd over existing algorithms. However, any theoretical development of the asymptotic distributions of the estimated singular values and vectors could prove useful for yielding a confidence interval estimate, which needs further investigation.

The specific real-life application to the video surveillance background and foreground extraction, which have been the primary target of this paper, and the aforementioned latent market factor estimation in financial stock price modelling are only a few among the broad spectrum of the diverse possible applications where rSVDdpd could prove extremely useful. While rSVDdpd can be used to replace the standard SVD in various algorithms to counter data contamination, in some of the problems like community detection in graphs (Sarkar and Dong, 2011; Malliaros and Vazirgiannis, 2013), rSVDdpd fails to become beneficial. In such problems, the singular value decomposition is generally performed on the adjacency matrix or Laplacian matrix of the graph, and due to sparseness of these matrices, the robust lower-rank approximation becomes equal to the null matrix (matrix with all the entries equal to 0). Preserving robustness even in the scenarios of sparsity could be a direction for extension of the proposed method. Also, the existing regularized robust SVD algorithm RobrSVD (Zhang, Shen and Huang, 2013) imposes a smoothness assumption on the entries of the singular vectors $u$ and $v$, which increases its complexity manyfold making it inefficient to scale for large matrices. On the other hand, rSVDdpd does not assume such a condition, hence it obtains the singular values and vectors under a very general setup. If the smoothness
condition is known a-priori, the objective function of rSVDdpd can be modified to include a regularization term, but this will increase the time and memory complexity if implemented naively. Circumventing this problem while reaping the benefit of regularization is another useful future direction of research.

**Software Availability.** For broader dissemination of the proposed algorithm rSVD-dpd, we have also made a R package “rsvddpd” with an implementation of the algorithm. The package and all the codes have been made available in a github repository [https://github.com/subroy13/rsvddpd](https://github.com/subroy13/rsvddpd).
S1. A Brief Review of Minimum Density Power Divergence Estimator. Basu et al. (1998) introduced the density power divergence as a measure of discrepancy between two probability density functions, which being an M-estimator enjoys various theoretical properties. The density power divergence between the densities \( g \) and \( f \) is defined as

\[
d_{\alpha}(g,f) = \begin{cases} 
\int \left\{ f^{1+\alpha} - \left(1 + \frac{1}{\alpha}\right)f^\alpha g + \frac{1}{\alpha}g^{1+\alpha} \right\} \, d\lambda & \alpha > 0 \\
\int g \ln \left( \frac{g}{f} \right) \, d\lambda & \alpha = 0
\end{cases}.
\]

Here \( \ln(\cdot) \) denotes the natural logarithm. The control parameter \( \alpha \) provides a smooth bridge between robustness and efficiency.

In case of independent and identically distributed samples, \( Y_1, Y_2, \ldots, Y_n \), with true distribution function \( G \) and corresponding density \( g \), we model this unknown density by a parametric family of densities \( \mathcal{F}_\theta = \{f_\theta : \theta \in \Theta\} \). The proposed estimator is then obtained as

\[
\hat{\theta} = \arg \min_{\theta \in \Theta} d_{\alpha}(dG_n, f_\theta),
\]

where \( G_n \) is the empirical distribution function. This can be shown to be equivalent to

\[
\hat{\theta} = \arg \min_{\theta \in \Theta} \left[ \int f_{\theta}^{1+\alpha} - \left(1 + \frac{1}{\alpha}\right) \frac{1}{n} \sum_{i=1}^{n} f_\theta(Y_i)^\alpha \right].
\]

Later, Ghosh and Basu (2013) extended this work by allowing independent but not identically distributed data. In this case, the observed data \( Y_i \sim g_i \), where each \( g_i \) is an unknown density. Each true density \( g_i \) is modeled by a corresponding parametric family of densities \( \mathcal{F}_{i,\theta} = \{f_{i,\theta} : \theta \in \Theta\} \) for \( i = 1, 2, \ldots, n \). Finally, the proposed MDPD estimator is obtained as

\[
\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} \left[ \int f_{i,\theta}^{1+\alpha} - \left(1 + \frac{1}{\alpha}\right) f_{i,\theta}(Y_i)^\alpha \right].
\]

Various nice theoretical properties like consistency and asymptotic normality of the above MDPD estimator have been proved by Ghosh and Basu (2013).

S2. Smoothness of Stereographic Projection. The stereographic projection employed in the paper allows transformation between the model parameters and the natural parameters. It is given by

\[
T(\lambda,\{a_i\}_{i=1}^{n},\{b_j\}_{j=1}^{p},\sigma^2) = \left(\lambda,\left(\frac{a_i}{1-a_n}\right)_{i=1}^{(n-1)},\left(\frac{b_j}{1-b_n}\right)_{j=1}^{(p-1)},\sigma^2\right),
\]

and,
\[ \mathcal{T}^{-1}(\lambda, \{\alpha_i\}_{i=1}^{n-1}, \{\beta_j\}_{j=1}^{p-1}, \sigma^2) = \left(\lambda, \left\{ \frac{2\alpha_i}{U^2+1} \right\}_{i=1}^{n-1}, \frac{U^2-1}{U^2+1}, \left\{ \frac{2\beta_j}{V^2+1} \right\}_{j=1}^{p-1}, \frac{V^2-1}{V^2+1}, \sigma^2 \right), \]

where \( U^2 = \sum_{i=1}^{n-1} \alpha_i^2 \) and \( V^2 = \sum_{j=1}^{p-1} \beta_j^2 \). This one-one transformation \( \mathcal{T} \) from the parameter space \( \Theta \) to the natural parameter space \( \Xi \) given by Eq. (S2.1) and its inverse given by Eq. (S2.2) are twice continuously differentiable functions, as long as \( a^T \neq (0, \ldots, 0, 1) \) and \( b^T \neq (0, \ldots, 0, 1) \). The gradient and the jacobian of the inverse transformation \( \mathcal{T}^{-1} \) are important objects of study since they allow one to use the chain rule of differentiation and convert the first order and second order derivative of the objective function \( H_{n,p} \) with respect to the original parameter \( \theta \) to the derivatives with respect to the natural parameter \( \eta \). Focusing on only the left singular vector \( a \) and its corresponding natural parametric representation \( \alpha = \{\alpha_i\}_{i=1}^{n-1} \), we have

\[ \frac{\partial a_k}{\partial \alpha_i} = \begin{cases} (1-a_n) a_i & \text{if } k = n \\ (1-a_n) - a_i^2 & \text{if } k \neq n, k = i \\ -a_i a_k & \text{if } k \neq n, k \neq i \end{cases} \]

and

\[ \frac{\partial^2 a_k}{\partial \alpha_i \partial \alpha_j} = \begin{cases} (1-a_n)^2 - 2a_i^2 (1-a_n) & \text{if } k = n, j = i \\ -2a_i a_j (1-a_n) & \text{if } k = n, j \neq i \\ 2a_i^3 - 3(1-a_n) a_i & \text{if } k \neq n, k = i = j \\ 2a_i^2 a_k - a_k(1-a_n) & \text{if } k \neq n, k \neq i, i = j \\ 2a_i^2 a_j - (1-a_n) a_j & \text{if } k \neq n, k = i, i \neq j \\ 2a_i^2 a_i - (1-a_n) a_i & \text{if } k \neq n, k = j, i \neq j \\ -2a_i a_j a_k & \text{if } k \neq n, \text{ and } k, i, j \text{ are all distinct} \end{cases} \]

Similarly, one may obtain the derivatives with respect to the natural parameteric representation \( \beta \) as well.

**S3. Proofs of the Results.** Before proving the individual results, we note that \( V(y; c, d, \sigma^2) \) can also be expressed as

\[ V(y; c, d, \sigma^2) = \int \frac{1}{\sigma^{1+\alpha}} \phi \left( \frac{x-cd}{\sigma} \right)^{1+\alpha} dx - \left( 1 + \frac{1}{\alpha} \right) \frac{1}{\sigma^\alpha} \phi \left( \frac{y-cd}{\sigma} \right)^\alpha, \]

where \( \phi \) is the density function of the standard normal distribution. We shall be using this expression of \( V \) in the proofs.

**Theorem 8.** Let the data matrix \( X \) be such that \( X_{ij} \sim \mathcal{N}(\lambda^* a_i^* b_j^*, (\sigma^*)^2) \) i.e., \( X_{ij} \) follows a normal distribution with mean \( \lambda^* a_i^* b_j^* \) and variance \((\sigma^*)^2\), for \( i = 1, 2, \ldots, n, \ j = 1, 2, \ldots, p, \) with \( \sum_i (a_i^*)^2 = \sum_j (b_j^*)^2 = 1 \), and the elements \( X_{ij} \)'s are independently distributed. Then there exists a best fitting parameter given by \( \Theta^g = (\lambda^*, \{a_i^*\}_{i=1}^n, \{b_j^*\}_{j=1}^p, (\sigma^*)^2) \). Moreover, this parameter is unique up to the magnitude of each of its coordinates.
PROOF. We need to show that the given parameter $\theta^g$ satisfies the four conditions

1.  
   \[
   (S3.2) \quad \sum_{i=1}^{n} (a_i^g)^2 = 1 \quad \text{and} \quad \sum_{j=1}^{p} (b_j^g)^2 = 1.
   \]

2. For any $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$,
   \[
   (S3.3) \quad \lambda^g a_i^g = \arg \min_a \int V(x; a, b_j^g, (\sigma^g)^2)g_{ij}(x)dx.
   \]

3. For any $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$,
   \[
   (S3.4) \quad \lambda^g b_j^g = \arg \min_b \int V(x; a_i^g, b, (\sigma^g)^2)g_{ij}(x)dx.
   \]

4. For any $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, p$,
   \[
   (S3.5) \quad (\sigma^g)^2 = \arg \min_{\sigma^2} \int V(x; \lambda^g a_i^g, b_j^g, \sigma^2)g_{ij}(x)dx = \arg \min_{\sigma^2} \int V(x; a_i^g, \lambda^g b_j^g, \sigma^2)g_{ij}(x)dx.
   \]

Eq. (S3.2) is verified by the conditions given for $a_i^*$ and $b_j^*$’s. To verify Eq. (S3.3), note that with $b_j^g = b_j^*$ and $\sigma^g = (\sigma^*)$, the quantity in Eq. (S3.3) is same as minimizing

\[
(S3.6) \quad \int (\sigma^*)^{-(1+\alpha)} \phi \left(\frac{x - ab_j^*}{\sigma^*}\right)^{(1+\alpha)} dx - \left(1 + \frac{1}{\alpha}\right) \int (\sigma^*)^{-\alpha} \phi \left(\frac{x - ab_j^*}{\sigma^*}\right)^{\alpha} g_{ij}(x)dx + \frac{1}{\alpha} \int g_{ij}(x)^{(1+\alpha)} dx,
\]

as the last term is independent of the minimization over $a$. The quantity given in Eq. (S3.6) is the density power divergence between the normal density with mean $ab_j^*$ and variance $\sigma^*$ and, the true density $g_{ij}$. From Theorem 2.1 of Basu et al. (1998), it follows that this divergence is minimized if and only if two densities match. Since, it is given that $g_{ij}$ is the normal density with mean $\lambda^* a_i^* b_j^*$ and variance $(\sigma^*)^2$, it follows that Eq. (S3.6) is minimized when $a = \lambda^* a_i^*$. By exactly similar logic and interchanging the roles of $a_i$ and $b_j$, Eq. (S3.4) and Eq. (S3.5) can also be verified. This proves that $\theta^* = \left(\lambda^*, \{a_i\}_{i=1}^{n}, \{b_j\}_{j=1}^{p}, (\sigma^*)^2\right)$ is a “best” fitting parameter for the given setup.

In order to prove uniqueness, suppose $\theta = \left(\lambda, \{a_i\}_{i=1}^{n}, \{b_j\}_{j=1}^{p}, \sigma^2\right)$ be another “best” fitting parameter. Then again, the quantity

\[
(S3.7) \quad \int \sigma^{-(1+\alpha)} \phi \left(\frac{x - ab_j}{\sigma}\right)^{(1+\alpha)} dx - \left(1 + \frac{1}{\alpha}\right) \int \sigma^{-\alpha} \phi \left(\frac{x - ab_j}{\sigma}\right)^{\alpha} g_{ij}(x)dx + \frac{1}{\alpha} \int g_{ij}(x)^{(1+\alpha)} dx,
\]

is minimized at $a = \lambda a_i$, independently of the choice of $j$. However, Eq. (S3.7) can be made equal to its minimum value 0 if and only if $\sigma^2 = (\sigma^*)^2$, and $\lambda a_i b_j = \lambda^* a_i^* b_j^*$ (follows from Theorem 2.1 of Basu et al. (1998)). Since, both $\theta$ and $\theta^*$ are “best” fitting parameters, they must satisfy Eq. (S3.2). Hence, $(\lambda a_i)^2 = \sum_j (\lambda a_i b_j)^2 = \sum_j (\lambda^* a_i^* b_j^*)^2 = (\lambda^* a_i^*)^2$. Taking sum over the row index $i$ now gives $|\lambda| = |\lambda^*|$. Then, it easily follows that $|a_i| = |a_i^*|$ and $|b_j| = |b_j^*|$ as well. \qed
THEOREM 9. If a “best” fitting parameter for the matrix $X$ is $\theta^g = (\lambda^g, \{a_i^g\}_{i=1}^n, \{b_j^g\}_{j=1}^p, (\sigma^g)^2)$, then a “best” fitting parameter for the matrix $cX$ is $\tilde{\theta}^g = (c\lambda^g, \{a_i^g\}_{i=1}^n, \{b_j^g\}_{j=1}^p, (c\sigma^g)^2)$ for any real constant $c$.

PROOF. It is obvious that $\tilde{\theta}^g$ satisfies Eq. (S3.2) as $\theta^g$ is given to be a “best” fitting parameter.

Considering the matrix $Y = cX$, let us denote the true density of $Y_{ij}$ as $g^g_{ij}(-)$, as opposed to $g_{ij}(-)$ denoting the true density of $X_{ij}$. A change of variable formula yields that $g^g_{ij}(y) = (1/c)g_{ij}(y/c)$. Hence, from substitution principle of integration it follows that

$$ \int V(y; a, b^g_j, c^2(\sigma^g)^2) g^g_{ij}(y)dy $$

$$ = \left[ (c\sigma^g)^{-(1+\alpha)} \int \phi \left( \frac{x - ab^g_j}{c\sigma^g} \right)^{(1+\alpha)} dx - \left( 1 + \frac{1}{\alpha} \right) \int (c\sigma^g)^{-\alpha} \phi \left( \frac{y - ab^g_j}{c\sigma^g} \right)^{1+\alpha} \frac{1}{c} g_{ij} \left( \frac{y}{c} \right) dy \right] $$

$$ = c^{-\alpha} \int (\sigma^g)^{-(1+\alpha)} \phi \left( \frac{z - (a/c)b^g_j}{\sigma^g} \right)^{(1+\alpha)} dz - \left( 1 + \frac{1}{\alpha} \right) \int (\sigma^g)^{-\alpha} \phi \left( \frac{z - (a/c)b^g_j}{\sigma^g} \right)^{1+\alpha} g_{ij} \left( \frac{z}{\sigma^g} \right) dz $$

$$ = c^{-\alpha} \int V(z; a/c, b^g_j, (\sigma^g)^2) g_{ij}(z)dz. $$

Since, the latter is minimized at $a/c = \lambda^g a_i^g$ when the minimization is performed with respect to the first parameter, the former quantity $\int V(y; a, b_j, c^2(\sigma^g)^2) g_{ij}(y)dy$ is minimized at $a = c\lambda^g a_i^g$. This verifies Eq. (S3.3).

As in the above case, Eq. (S3.4) can also be established by interchanging the role of $\alpha$ and $b$ above.

For the parameter $\sigma$, it can be verified that a substitution principal again applies, and we obtain

$$ \int V(y; c\lambda^g a_i^g, b^g_j, \sigma^2) g^g_{ij}(y)dy = \int V(y; a_i^g, c\lambda^g b^g_j, \sigma^2) g_{ij}(y)dy $$

$$ = c^{-\alpha} \int V(z; \lambda a_i^g, b^g_j, \sigma^2/c^2) g_{ij}(z)dz. $$

Again by the hypothesis that $\theta^g$ is a “best” fitting parameter, the latter is minimized when $\sigma/c = \sigma^g$, hence the former is minimized at $\sigma^2 = c^2(\sigma^g)^2$. This verifies Eq. (S3.5).

THEOREM 10. If a “best” fitting parameter for the matrix $X$ is $\theta^g = (\lambda^g, \{a_i^g\}_{i=1}^n, \{b_j^g\}_{j=1}^p, (\sigma^g)^2)$, then a “best” fitting parameter for the matrix $PXQ^T$ where $P, Q$ are permutation matrices corresponding to the permutations $\pi_R$ and $\pi_C$, is $\tilde{\theta}^g = (\lambda^g, \{a_{\pi_R(i)}^g\}_{i=1}^n, \{b_{\pi_C(j)}^g\}_{j=1}^p, (\sigma^g)^2)$.

PROOF. Let, $Y = PXQ^T$. Then, Eq. (S3.2) is satisfied for the new setup as $\sum_{i=1}^n (a_{\pi_R(i)}^g)^2 = \sum_{i=1}^n (a_i^g)^2 = 1$ and similarly $\sum_{j=1}^p (b_{\pi_C(j)}^g)^2 = \sum_{j=1}^p (b_j^g)^2 = 1$.

To see that Eq. (S3.3) holds for the new setup with $\tilde{\theta}^g$, note that for every $j = 1, 2, \ldots p$, the minimizer of the integral in Eq. (S3.3) is $a_j^g$, independent of the choice of $j$. Now, considering Eq. (S3.3) for $\pi_R^{-1}(i)$ and $\pi_C^{-1}(j)$ instead of $i$ and $j$, we obtain

$$ \int V(y; a_{\pi_R(i)}^g, b^g_{\pi_C(j)}, \sigma^2) g^g_{ij}(y)dy = \int V(y; a_i^g, b_j^g, \sigma^2) g_{ij}(y)dy $$

$$ = c^{-\alpha} \int V(z; \lambda a_i^g, b_j^g, \sigma^2/c^2) g_{ij}(z)dz. $$
\begin{align}
\lambda^g a^g_{\pi^{-1}_R(i)} = \arg \min_a \int V \left( x; a, b^g_{\pi^{-1}_C(j)}, (\sigma^g)^2 \right) g_{\pi^{-1}_R(i), \pi^{-1}_C(j)}(x) \, dx.
\end{align}

However, \(a^g_{\pi^{-1}_R(i)}\) is the \(i\)-th entry of the permuted sequence \(\{a^g_{\pi_R(i)} : i = 1, 2, \ldots, n\}\), i.e., if we consider a vector \(a^g\) with its entries \(a^g_i\), then \(a^g_{\pi^{-1}_R(i)}\) is the \(i\)-th entry of \(Pa\). Similarly, \(b^g_{\pi^{-1}_C(j)}\) is the \(j\)-th entry of \(Qb\). And finally, the elements of the new matrix are \(Y_{ij} = X_{\pi_R(i), \pi_C(j)}\), thus the density for the element \(Y_{ij}\) is \(g_{\pi^{-1}_R(i), \pi^{-1}_C(j)}(y)\) which can be verified by a change of variable formula. Combining these, Eq. (S3.8) can be reformulated as

\begin{align}
\lambda^g(Pa^g)_i = \arg \min_a \int V \left( x; a, (Qb^g)_j, (\sigma^g)^2 \right) g_{Y_{ij}}(x) \, dx.
\end{align}

This shows that Eq. (S3.3) holds for new matrix \(PXQ^T\) with the given best fitting parameter \(\hat{\theta}^g\). Eq. (S3.4) can be verified by imitating the same proof, except interchanging the role of \(a\) and \(b\).

Finally, Eq. (S3.5) for the permuted matrix follows from noting that

\begin{align*}
\int V \left( x; \lambda^g a^g_{\pi^{-1}_R(i)}, b^g_{\pi^{-1}_C(j)}, (\sigma)^2 \right) g_{\pi^{-1}_R(i), \pi^{-1}_C(j)}(x) \, dx \\
= \int V \left( x; \lambda^g(Pa^g)_i, (Qb^g)_j, (\sigma)^2 \right) g_{Y_{ij}}(x) \, dx.
\end{align*}

\[\square\]

**Theorem 11.** Let \(\theta^* = (\lambda^*, \{a^*_i\}_{i=1}^n, \{b^*_j\}_{j=1}^p, (\sigma^*)^2)\) be the converged estimator of singular value of \(X\) obtained by rSVDdpd, starting from \(\theta^{(0)} = (\lambda^{(0)}, \{a^{(0)}_i\}_{i=1}^n, \{b^{(0)}_j\}_{j=1}^p, (\sigma^{(0)})^2)\). Then, starting with the new initial estimate \((c\lambda^{(0)}, \{a^{(0)}_i\}_{i=1}^n, \{b^{(0)}_j\}_{j=1}^p, (c(\sigma^{(0)})^2)\), the rSVDdpd estimator for the data matrix \(cX\) converges to \((\lambda^*, \{a^*_i\}_{i=1}^n, \{b^*_j\}_{j=1}^p, (\sigma^*)^2)\).

**Proof.** Let us denote \(\theta^{(t)} = (\lambda^{(t)}, \{a^{(t)}_i\}_{i=1}^n, \{b^{(t)}_j\}_{j=1}^p, (\sigma^{(t)})^2)\) as the estimate at the \(t\)-th iteration when the data matrix is \(X\). Also, let us denote \(\tilde{\theta}^{(t)} = (\tilde{\lambda}^{(t)}, \{\tilde{a}^{(t)}_i\}_{i=1}^n, \{\tilde{b}^{(t)}_j\}_{j=1}^p, (\tilde{\sigma}^{(t)})^2)\) to be the estimate in the \(t\)-th iteration when the data matrix is \(cX\). Clearly, it is then enough to show that

\[\tilde{\lambda}^{(t)} = c\lambda^{(t)}, \quad (\tilde{\sigma}^{(t)})^2 = c^2(\sigma^{(t)})^2, \quad \tilde{a}^{(t)}_i = a^{(t)}_i, \quad \tilde{b}^{(t)}_j = b^{(t)}_j, \quad i = 1, \ldots, n; \quad j = 1, \ldots, p; \quad t = 1, 2, \ldots.\]

This can be shown by using the principle of mathematical induction. For \(t = 0\), the claim is validated by the equivariance of the initial estimate. To show the inductive step, we notice that the iteration formulas given by

\begin{align}
\lambda^{(t+1/2)} a^{(t+1)}_i &= \arg \min_a \frac{1}{p} \sum_{j=1}^p V(x_{ij}; a, b^{(t)}_j, (\sigma^{(t)})^2), \quad i = 1, \ldots, n; \quad t = 0, 1, \ldots,
\end{align}

\begin{align}
\lambda^{(t+1)} b^{(t+1)}_j &= \arg \min_b \frac{1}{n} \sum_{i=1}^n V(x_{ij}; a^{(t+1)}_i, b, (\sigma^{(t)})^2), \quad j = 1, \ldots, p; \quad t = 0, 1, \ldots,
\end{align}

are used.
(S3.11) \[ (\sigma^2)^{(t+1)} = \arg\min_{\sigma^2} \frac{1}{np} \sum_{i=1}^{n} \sum_{j=1}^{p} V(x_{ij}; a_i^{(t+1)}, b_j^{(t+1)}, \sigma^2), \quad t = 0, 1, \ldots, \]

allows one to express the estimates at \((t+1)\)-th iteration as

\[
\lambda^{(t+1)/2}\tilde{a}_i^{(t+1)} = \arg\min_{a} \frac{1}{p} \sum_{j=1}^{p} V(cx_{ij}; a, b_j^{(t)}, \tilde{\sigma}^{(t)})
\]

\[
= \arg\min_{a} \frac{1}{p} \sum_{j=1}^{p} V(cx_{ij}; a, b_j^{(t)}, c\sigma^{(t)}), \text{ by induction hypothesis}
\]

\[
= \arg\min_{a} -\frac{1}{p} \sum_{j=1}^{p} \frac{(1 + \alpha)}{\alpha(2\pi)^{\alpha/2}(c\sigma^{(t)})^{\alpha}} \exp \left\{ -\alpha \frac{(cx_{ij} - ab_j^{(t)})}{c\sigma^{(t)}} \right\},
\]

by empirical version of Eq. (S3.1)

\[
= \arg\min_{a} -\frac{1}{p} \sum_{j=1}^{p} \frac{(1 + \alpha)}{\alpha(2\pi)^{\alpha/2}(c\sigma^{(t)})^{\alpha}} \exp \left\{ -\alpha \frac{(x_{ij} - (a/c)b_j^{(t)})}{\sigma^{(t)}} \right\}
\]

\[
= \arg\min_{a} \frac{1}{p} \sum_{j=1}^{p} V(x_{ij}; a/c, b_j^{(t)}, \sigma^{(t)})
\]

\[
= c\lambda^{(t+1)/2}a_i^{(t+1)},
\]

where the last line follows from the fact that the same iteration formula yields the minimizer \((a/c) = \lambda^{(t+1)/2}a_i^{(t+1)}\) due to the very definition of \(a_i^{(t+1)}\).

An identical chain of equality for Eq. (S3.10) can be used to show that \(\lambda^{(t+1)}b_j^{(t+1)} = c\lambda^{(t+1)}b_j^{(t+1)}\), while a similar derivation based on Eq. (S3.11) establishes \((\tilde{\sigma}^{(t+1)})^2 = (c\sigma^{(t+1)})^2\). Finally, since the estimates of the singular vectors are normalized and restricted to be in the parameter space \(\Theta = [0, \infty) \times S_n^+ \times S_p^+ \times [0, \infty)\), the inductive step follows. \(\square\)

**Theorem 12.** Let \(\theta^* = (\lambda^*, \{a_i^*\}_{i=1}^n, \{b_j^*\}_{j=1}^p, (\sigma^*)^2)\) be the converged estimator of singular value of \(X\) obtained by rSVDpd, starting from the initial estimate \(\theta^{(0)} = (\lambda^{(0)}, \{a_i^{(0)}\}_{i=1}^n, \{b_j^{(0)}\}_{j=1}^p, (\sigma^{(0)})^2)\). Also, let \(P\) and \(Q\) be the permutation matrices corresponding to the permutations \(\pi_R\) and \(\pi_C\) respectively. Then, starting with the new initial estimate \((\lambda^{(0)}, \{a_{\pi_R(i)}^{(0)}\}_{i=1}^n, \{b_{\pi_C(j)}^{(0)}\}_{j=1}^p, (\sigma^{(0)})^2)\), the rSVDpd estimator for the data matrix \(PXQ^T\) converges to \((\lambda^*, \{a_{\pi_R(i)}^*\}_{i=1}^n, \{b_{\pi_C(j)}^*\}_{j=1}^p, (\sigma^*)^2)\).

**Proof.** This proof is very similar to the proof of Theorem 11. We shall again denote \(\theta^{(t)} = (\lambda^{(t)}, \{a_i^{(t)}\}_{i=1}^n, \{b_j^{(t)}\}_{j=1}^p, (\sigma^{(t)})^2)\) as the estimate at the \(t\)-th iteration when the data matrix is \(X\) and \(\tilde{\theta}^{(t)} = (\tilde{\lambda}^{(t)}, \{\tilde{a}_i^{(t)}\}_{i=1}^n, \{\tilde{b}_j^{(t)}\}_{j=1}^p, (\tilde{\sigma}^{(t)})^2)\) as the estimate in the \(t\)-th iteration when the data matrix is \(PXQ^T\). Again, it is enough to show that

\[
\tilde{\lambda}^{(t)} = c\lambda^{(t)}, \quad (\tilde{\sigma}^{(t)})^2 = c^2(\sigma^{(t)})^2, \quad \tilde{a}_i^{(t)} = a_i^{(t)}, \quad \tilde{b}_j^{(t)} = b_j^{(t)}, \quad i = 1, \ldots n; j = 1, \ldots p; t = 1, 2, \ldots,
\]
which we shall show using the principle of mathematical induction. The initial case $t = 0$ follows from the equivariance of the initial estimate. Starting with the iteration formula Eq. (S3.9), we have

$$\overline{\lambda}^{(t+1/2)} a_{\pi_R(i)}^{(t+1)} = \arg \min_a \frac{1}{p} \sum_{j=1}^p V(x_{\pi_R(i), \pi_C(j)}; a, b_j^{(t)}, \sigma^{(t)})$$

$$= \arg \min_a \frac{1}{p} \sum_{j=1}^p V(x_{\pi_R(i), \pi_C(j)}; a, b_j^{(t)}, \sigma^{(t)}), \text{ by induction hypothesis}$$

$$= \arg \min_a \frac{1}{p} \sum_{j=1}^p V(x_{\pi_R(i), j}; a, b_j^{(t)}, \sigma^{(t)}), \text{ since, } \pi_C \text{ is a permutation}$$

$$= \lambda^{(t+1/2)} a_{\pi_R(i)}^{(t+1)}$$

where the last equality follows from the fact that the minimization criteria satisfied by $a_{\pi_R(i)}^{(t+1)}$ is same as minimization of the average $V$-values for the corresponding row index $\pi_R(i)$ of the data matrix $X$. The other equations $\lambda^{(t+1)} \tilde{b}_j^{(t+1)} = \lambda^{(t+1)} b_j^{(t)}$ and $(\tilde{\sigma}^{(t+1)})^2 = (\sigma^{(t+1)})^2$ follow similarly from Eq. (S3.10) and (S3.11) respectively. Again, restricting the parameter estimates in $\Theta = [0, \infty) \times S_n^+ \times S_p^+ \times [0, \infty)$, the inductive step follows.

**THEOREM 13.** Consider fixed $n$ and $p$. Under Assumptions (A1)-(A2), the estimates $\theta^{(t)} = (\lambda^{(t)}, \{a_i^{(t)}\}_{i=1}^n, \{b_j^{(t)}\}_{j=1}^p, (\sigma^2)^{(t)})$ obtained through Eq. (S3.9)-(S3.11) converge to the minimizer $\theta^*$ of $H_{n,p}$ provided that the initial starting value of the parameter $\theta$ in the algorithm, i.e. $\theta^{(0)}$, belongs to the compact set $\Theta$.

**PROOF.** We shall employ Theorem 10.3.3 of Argyros (2008) to show that our estimation process actually converges to the desired minimizer. Going by the notation of Argyros (2008), let $F_t$ denote one step of iteration. Since, each of the iteration steps aim to identify a minima and there is a possibility of existence of multiple minima, we have

$$(\lambda^{(t+1)}, \{a_i^{(t+1)}\}_{i=1}^n, \{b_j^{(t+1)}\}_{j=1}^p, (\sigma^2)^{(t+1)}) \in F_t (\lambda^{(t)}, \{a_i^{(t)}\}_{i=1}^n, \{b_j^{(t)}\}_{j=1}^p, (\sigma^2)^{(t)}) .$$

Before proceeding further, we identify the set of desired points $P$ as the singleton set $\{\theta^*\}$. Note that this is essentially a one step process. The conditions (C1)-(C3) of Theorem 10.3.3 of Argyros (2008) follows from the assumptions (A1)-(A2) and the specific iterative process given by Eq. (S3.9)-(S3.11).

1. Condition (C1) is easily satisfied by the definition of minima, as the iteration process applied on $\theta^*$ would retain at $\theta^*$ since it is the minimum of the function $H_{n,p}$, defined in Eq. (3.10) in the main paper, if we restrict our attention to $\Theta$.
2. Condition (C3) is implied by condition (A1).
3. To show that condition (C2) follows, we choose the Lyapunov function as $H_{n,p}$ itself. Note that, $H_{n,p}$ is necessarily continuous in its arguments. Due to the specific iteration rules in Eq. (S3.9)-(S3.11), all estimates are obtained in alternating fashion by minimizing the parameters subject to the most recent value of the other parameters. Thus is follows that
\[ \mathcal{H}_{n,p}(\lambda(t), \{a_i^{(t)}\}_{i=1}^n, \{b_j^{(t)}\}_{j=1}^p, (\sigma^2(t))) \geq \mathcal{H}_{n,p}(\lambda(t+1/2), \{a_i^{(t+1)}\}_{i=1}^n, \{b_j^{(t+1)}\}_{j=1}^p, (\sigma^2(t))) \]

\[ \geq \mathcal{H}_{n,p}(\lambda(t+1), \{a_i^{(t+1)}\}_{i=1}^n, \{b_j^{(t+1)}\}_{j=1}^p, (\sigma^2(t))) \]

\[ \geq \mathcal{H}_{n,p}(\lambda(t+1), \{a_i^{(t+1)}\}_{i=1}^n, \{b_j^{(t+1)}\}_{j=1}^p, (\sigma^2(t+1))). \]

Therefore, a full step of iteration monotonically decreases (may not be strictly) the function \( \mathcal{H}_{n,p} \). This shows that the chosen Lyapunov function is closed.

Now an application of Theorem 10.3.3 of Argyros (2008) completes the proof. \( \square \)

**Theorem 14.** Under Assumptions (B1)-(B5), the rSVDpd estimate \( \theta_{n,p}^* \) corresponding to the same \( \alpha \) from assumption (B4), is a consistent estimator of the sequence of the “best” fitting parameter values \( \theta_{n,p}^0 \), as both dimensions of the data matrix \( X_{n,p} \) (i.e. \( n \) and \( p \)) tend to infinity. In other words, \( \left( \theta_{n,p}^* - \theta_{n,p}^0 \right) \to 0 \) in probability as both \( n \to \infty \) and \( p \to \infty \) subject to a constant ratio in limit, i.e. \( \lim_{\substack{n \to \infty \ \ \ p \to \infty \ \ \ \frac{n}{p} = c}} \) with \( 0 < c < \infty \).

**Proof.** First, we observe that, the stereographic transformation mentioned in the discussion prior to Theorem 14 can be employed and would remain valid because of assumption (B2) and (B3).

Now, to show such consistency, we shall take a route similar to the one taken by Ghosh and Basu (2013) as in the case of the MDPE for INH setup. Instead of showing that the rSVDpd converged estimator, i.e., the minimizer \( \theta_{n,p}^* \) of \( \mathcal{H}_{n,p} \) is consistent for \( \theta_{n,p}^0 \), we shall show instead that \( \eta_{n,p}^* \) is consistent for \( \eta_{n,p}^0 \). By a slight abuse of notation throughout this proof, we shall use \( \mathcal{H}_{n,p}(\eta) \) to indicate \( \mathcal{H}_{n,p}(\mathcal{T}(\theta)) \). To prove that \( \eta_{n,p}^* \) is consistent for \( \eta_{n,p}^0 \), we shall show that for any sufficiently small \( r > 0 \), \( \mathcal{H}_{n,p}(\eta_{n,p}) > \mathcal{H}_{n,p}(\eta_{n,p}) \) for sufficiently large \( n \) and \( p \), for any \( \eta_{n,p} \) with \( \|\eta_{n,p} - \eta_{n,p}^0\|_2 = r \). This means that the value of \( \mathcal{H}_{n,p}(\eta_{n,p}) \) at the surface of the ball of radius \( r \) centered at \( \eta_{n,p}^0 \) would be higher than its value at \( \eta_{n,p}^0 \), and hence by smoothness of \( \mathcal{H}_{n,p} \), it is ensured that there will be a local minima inside that ball. Proceeding as in Ghosh and Basu (2013), we start by expanding \( \mathcal{H}_{n,p}(\eta_{n,p}) \) by its Taylor series expansion about \( \eta_{n,p}^0 \), for any fixed \( n \) and \( p \). For notational convenience, we suppress the subscripts from \( \eta \) and \( \eta^0 \) which should be obvious from the context. Thus

\[
\mathcal{H}_{n,p}(\eta) - \mathcal{H}_{n,p}(\eta^0) = \frac{\partial \mathcal{H}_{n,p}}{\partial \lambda} \bigg|_{\eta=\eta^0} (\lambda - \lambda^0) + \sum_{i=1}^{(n-1)} \frac{\partial \mathcal{H}_{n,p}}{\partial \alpha_i} \bigg|_{\eta=\eta^0} (\alpha_i - \alpha_i^0) \\
+ \sum_{j=1}^{(p-1)} \frac{\partial \mathcal{H}_{n,p}}{\partial \beta_j} \bigg|_{\eta_{n,p}=\eta_{n,p}^0} (\beta_j - \beta_j^0) + \frac{\partial \mathcal{H}_{n,p}}{\partial \sigma^2} \bigg|_{\eta_{n,p}=\eta_{n,p}^0} (\sigma^2 - (\sigma^0)^2) \\
+ \frac{1}{2} \sum_{k_1,k_2} \frac{\partial^2 \mathcal{H}_{n,p}}{\partial \eta_{k_1} \partial \eta_{k_2}} \bigg|_{\eta_{n,p}=\eta_{n,p}^0} (\eta_{k_1} - \eta_{k_1}^0)(\eta_{k_2} - \eta_{k_2}^0) \\
+ \frac{1}{3!} \sum_{k_1,k_2,k_3} \frac{\partial^3 \mathcal{H}_{n,p}}{\partial \eta_{k_1} \partial \eta_{k_2} \partial \eta_{k_3}} \bigg|_{\eta_{n,p}=\eta_{n,p}^0} (\eta_{k_1} - \eta_{k_1}^0)(\eta_{k_2} - \eta_{k_2}^0)(\eta_{k_3} - \eta_{k_3}^0) \\
= S_{1,1} + S_{1,2} + S_{1,3} + S_{1,4} + \frac{1}{2} S_2 + \frac{1}{6} S_3,
\]
where the quantities $S_{1,1}, S_{1,2}, S_{1,3}, S_{1,4}, S_2$ and $S_3$ respectively denotes the summands they are replacing. Here, $\eta_k$ denotes the $k$-th coordinate of the vector $\eta_{n,p}$ in the natural parameter space $\Xi \subset \mathbb{R}^{(n+p)}$. Also $\alpha_i$’s and $\beta_j$’s are the natural parametric representation of the elements of left $(a_{i,n})$ and right singular vectors $(b_{j,p})$ respectively, where the dimension subscripts ($n$ and $p$) have been suppressed for notational convenience as indicated before.

Clearly, the smoothness of $\mathcal{H}_{n,p}$ along with Assumption (B1) on the normality of the errors, indicates that $\int \mathcal{H}_{n,p}g(x)dx$ can be differentiated thrice with respect to $\eta_{n,p}$, and the derivative can be taken under the integral sign. Thus, it follows that

\[
\mathbb{E}
\left[
\frac{\partial \mathcal{H}_{n,p}}{\partial \eta_k}|_{\eta=\theta^*}
\right]
= \frac{\partial \mathbb{E}\mathcal{H}_{n,p}}{\partial \eta_k}|_{\eta=\theta^*}
= 0, \; k = 1, 2, \ldots (n + p),
\]

since the population version of the objective function $\mathbb{E}\mathcal{H}_{n,p}$ is minimized at the true parameter $\eta^\theta$. Thus, by a generalized version of Khinchin’s Weak Law of Large numbers, it follows that as $n$ and $p$ both increases to infinity, each of the first order partial derivatives goes in probability to 0. However, the problem arises as there are potentially infinitely many terms (as the parameter space increases in dimension). This jeopardizes any approach to naturally extending the proof of Theorem 3.1 of Ghosh and Basu (2013).

Let us consider each of the sums $S_{1,1}, S_{1,2}, S_{1,3}$ and $S_{1,4}$ pertaining to the first order derivative separately. Since $\frac{\partial \mathcal{H}_{n,p}}{\partial \lambda}|_{\eta=\theta^*}$ and $\frac{\partial \mathcal{H}_{n,p}}{\partial \sigma^2}|_{\eta=\theta^*}$ both converges in probability to 0, hence for sufficiently large $n$ and $p$, we have $|S_{1,1}| < r^3$ and $|S_{1,4}| < r^3$ with probability tending to 1.

Now to deal with the possibly infinite sums in $S_{1,2}$ or $S_{1,3}$, we shall try to compute their expectation and variance. By chain rule of differentiation, we have

\[
\frac{\partial \mathcal{H}_{n,p}}{\partial \alpha_i}|_{\eta=\theta^*} = \sum_{k=1}^{n} \frac{\partial \mathcal{H}_{n,p}}{\partial a_k}|_{\theta=\theta^*} \frac{\partial a_k}{\partial \alpha_i}|_{\eta=\theta^*}.
\]

Therefore,

\[
S_{n,p} = \sum_{i=1}^{(n-1)} \frac{\partial \mathcal{H}_{n,p}}{\partial \alpha_i}|_{\eta=\theta^*} = \sum_{i=1}^{(n-1)} \sum_{k=1}^{n} \frac{\partial \mathcal{H}_{n,p}}{\partial a_k}|_{\theta=\theta^*} \frac{\partial a_k}{\partial \alpha_i}|_{\eta=\theta^*} = \sum_{k=1}^{n} \frac{\partial \mathcal{H}_{n,p}}{\partial a_k}|_{\theta=\theta^*} \left( \sum_{i=1}^{(n-1)} \frac{\partial a_k}{\partial \alpha_i}|_{\eta=\theta^*} \right).
\]

Similar to Eq. (S3.12), one can verify that $\mathbb{E}\left[\frac{\partial \mathcal{H}_{n,p}}{\partial a_k}|_{\theta=\theta^*}\right] = 0$ for all $k = 1, 2, \ldots n$, and, therefore, Eq. (S3.13) implies that $\mathbb{E}(S_{n,p}) = 0$.

Turning to its variance, since the part of $\mathcal{H}_{n,p}$ dependent on $a_k$ would consist of only the $k$-th row of the data matrix $X$, which are assumed to be independently distributed in the current setup, it follows that

\[
\text{Cov}\left[\frac{\partial \mathcal{H}_{n,p}}{\partial a_k}|_{\theta=\theta^*}, \frac{\partial \mathcal{H}_{n,p}}{\partial a_l}|_{\theta=\theta^*}\right] = 0, \; 1 \leq k, l \leq n, \; \text{with} \; k \neq l.
\]

Therefore, it follows that
\[ \text{Var}(s_{n,p}) = \sum_{k=1}^{n} \left( \sum_{i=1}^{(n-1)} \frac{\partial a_k}{\partial \alpha_i} \right) \left( \frac{\partial a_k}{\partial \eta} \right)^2 \text{Var} \left( \frac{\partial H_{n,p}}{\partial a_k} \Bigg| \theta = \theta^* \right) \]

\[ = \sum_{k=1}^{n} \left( \sum_{i=1}^{(n-1)} \frac{\partial a_k}{\partial \alpha_i} \right) \left( \frac{\partial a_k}{\partial \eta} \right)^2 \left( \frac{1}{n^2} \frac{(2\pi)^{-\alpha}}{\sigma^g} (\sigma^g)^{-(2\alpha+2)} (1 + 2\alpha)^{-3/2} \sum_{j=1}^{p} (\lambda_j^g)^2 (b_j^g)^2 \right) \]

\[ = \sum_{k=1}^{n} \left( \sum_{i=1}^{(n-1)} \frac{\partial a_k}{\partial \alpha_i} \right) \left( \frac{\partial a_k}{\partial \eta} \right)^2 \left( \frac{1}{n^2} \frac{(2\pi)^{-\alpha}}{\sigma^g} (\sigma^g)^{-(2\alpha+2)} (1 + 2\alpha)^{-3/2} (\lambda^g)^2 \right). \]

From Cauchy-Schwartz inequality, it follows that \( \sum_{i=1}^{(n-1)} a_i^g \leq \sqrt{n-1} \) since \( \sum_{i=1}^{n} (a_i^g)^2 = 1 \), and also \( |a_i^g| \leq 1 \). Combining these two inequalities with Eq. (S2.3) yields

\[ \sum_{i=1}^{(n-1)} \frac{\partial a_i}{\partial \alpha_i} \bigg|_{\eta = \eta^g} = (1 - a_n) \sum_{i=1}^{(n-1)} a_i \leq 2\sqrt{n-1}, \]

and, for \( k \neq n \)

\[ \sum_{i=1}^{(n-1)} \frac{\partial a_k}{\partial \alpha_i} \bigg|_{\eta = \eta^g} = (1 - a_n) - a_k \sum_{i=1}^{(n-1)} a_i \leq 2 + \sqrt{n-1} \leq 2\sqrt{n-1}, \]

for any \( n \geq 5 \). Therefore, for large \( n \) and \( p \),

\[ \text{Var}(s_{n,p}) \leq 4(2\pi)^{-\alpha} (1 + 2\alpha)^{-3/2} (\lambda^g)^2 \frac{(\sigma^g)^{-(2\alpha+2)}}{np^2} = \Theta \left( \frac{(np)^{(2\alpha+2)/(\alpha+2)}}{np^2} \right), \]

since, \( (\sigma^g) = \Theta \left( \frac{(np)^{-1/(\alpha+2)}}{\alpha+2} \right) \). Now as \( 0 \leq \alpha \leq 1 < 2 \) implies \( \frac{2(2\alpha+2)}{\alpha+2} < 3 \) and as \( \lim_{n \to \infty} \frac{n}{p} = c < \infty \), it follows that \( \text{Var}(s_{n,p}) \to 0 \) as \( n, p \to \infty \). Thus, the quantity

\[ \left| \sum_{i=1}^{(n-1)} \left[ \frac{\partial H_{n,p}}{\partial \alpha_i} \bigg|_{\eta = \eta^g} \right] \right| \to 0, \]

with probability tending to one. Therefore, with \( |\alpha_i - \alpha_i^g| < r \), we have \( |S_{1,2}| < r^3 \), for sufficiently large \( n \) and \( p \), with probability tending to 1. Reversing the role of \( n \) and \( p \), and considering \( \sum_{j=1}^{p-1} \left[ \frac{\partial H_{n,p}}{\partial \beta_j} \bigg|_{\eta = \eta^g} \right] \) instead, one can show that \( |S_{1,3}| < r^3 \) for sufficiently large \( n \) and \( p \) with probability tending to 1. Thus, combining everything we obtain \( |S_1| \leq |S_{1,1}| + |S_{1,2}| + |S_{1,3}| + |S_{1,4}| < 4r^3 \) for sufficiently large \( n \) and \( p \), with probability tending to 1.

Now, turning our attention to the term \( S_2 \), we start by noting that

\[ \mathbb{E} \left[ \frac{\partial^2 H_{n,p}}{\partial \lambda^2} \bigg|_{\eta = \eta^g} \right] = (2\pi)^{-\alpha/2} (\sigma^g)^{-(\alpha+2)} (1 + \alpha)^{-1/2} \frac{\sum_i \sum_j (a_i^g)^2 (b_j^g)^2}{np} = \Theta \left( \frac{(\sigma^g)^{-(\alpha+2)}}{np} \right). \]
Because \( (\sigma^g) = \Theta((np)^{-1/(\alpha+2)}) \) by condition (B4), it then follows that asymptotically 
\[
\mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \lambda \partial \alpha_i} \bigg| \eta = \eta^p \right] \text{ is bounded away from zero, and, in particular, } \inf_{n,p} \mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \alpha_i} \bigg| \eta = \eta^p \right] > 0.
\]

The expected value of the mixed order derivatives, on the other hand, is equal to 0 as follows

\[
\mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \lambda \partial \alpha_i} \bigg| \eta = \eta^p \right] = \sum_{k=1}^{n} \mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \lambda \partial \alpha_k} \bigg| \theta = \theta^p \right] \frac{\partial a_k}{\partial \alpha_i} \bigg| \eta = \eta^p
\]

\[
= \sum_{k=1}^{n} (2\pi)^{-\alpha/2}(\sigma^g)^{-(\alpha+2)}(1 + \alpha)^{-1/2} \frac{\lambda^g d_k^g b_j^g}{np} \frac{\partial a_k}{\partial \alpha_i} \bigg| \eta = \eta^p
\]

\[
= \frac{\lambda^g (2\pi)^{-\alpha/2}(\sigma^g)^{-(\alpha+2)}(1 + \alpha)^{-1/2}}{np} \sum_{k=1}^{n} a_k^g \frac{\partial a_k}{\partial \alpha_i} \bigg| \eta = \eta^p = 0,
\]

since \( \sum_{i=1}^{n} a_k^g \frac{\partial a_k^g}{\partial \alpha_i} \bigg| \eta = \eta^p = 0 \). Similarly,

\[
\mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \alpha_i \partial \beta_j} \bigg| \eta = \eta^p \right] = 0.
\]

A chain rule of differentiation can be used to obtain the second order derivatives of \( H_{n,p} \) with respect to \( \alpha \)'s as follows

\[
\frac{\partial^2 H_{n,p}}{\partial \alpha_i \partial \alpha_j} = \sum_{k=1}^{n} \frac{\partial H_{n,p}}{\partial \alpha_k} \frac{\partial^2 a_k}{\partial \alpha_i \partial \alpha_j} + \sum_{k=1}^{n} \sum_{l=1}^{n} \frac{\partial^2 H_{n,p}}{\partial \alpha_k \partial \alpha_l} \frac{\partial a_k}{\partial \alpha_i} \frac{\partial a_l}{\partial \alpha_j}
\]

\[
= \sum_{k=1}^{n} \frac{\partial H_{n,p}}{\partial \alpha_k} \frac{\partial^2 a_k}{\partial \alpha_i \partial \alpha_j} + \sum_{k=1}^{n} \frac{\partial^2 H_{n,p}}{\partial \alpha_k \partial \alpha_l} \frac{\partial a_k}{\partial \alpha_i} \frac{\partial a_l}{\partial \alpha_j}.
\]

where the last equality follows from the fact that the first order derivative of \( H_{n,p} \) with respect to \( a_k \) would only yield nonzero quantity corresponding to \( k \)-th row, and for \( k \neq l \), \( \frac{\partial^2 H_{n,p}}{\partial \alpha_k \partial \alpha_l} = 0 \).

A simple calculation on the expectation now reveals that

\[
\mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \alpha_i \partial \alpha_j} \bigg| \eta = \eta^p \right] = \begin{cases} (\lambda^g)^2 \frac{(2\pi)^{-\alpha/2}(\sigma^g)^{-(\alpha+2)}(1 + \alpha)^{-1/2}}{np} \frac{(1 - a^g)^2}{a_i a_j} & \text{if } i = j, \\
0 & \text{if } i \neq j. 
\end{cases}
\]

Since \( a^g_i \neq \pm 1 \) by condition (B2), and \( \sigma^g = \Theta((np)^{-1/(\alpha+2)}) \) by condition (B4), it again follows that \( \inf_{n,p} \mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \alpha_i^2} \bigg| \eta = \eta^p \right] > 0 \). A very similar conclusion can be said about the quantities \( \mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \alpha_i \partial \beta_j} \bigg| \eta = \eta^p \right] \) as well.

For mixed order derivatives with respect to \( \sigma^2 \), as shown in Ghosh and Basu (2013) for normal linear regression setup, we would have

\[
\mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \alpha_i \partial \sigma^2} \bigg| \eta = \eta^p \right] = \mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \beta_j \partial \sigma^2} \bigg| \eta = \eta^p \right] = \mathbb{E}\left[ \frac{\partial^2 H_{n,p}}{\partial \lambda \partial \sigma^2} \bigg| \eta = \eta^p \right] = 0,
\]
and

\[
\mathbb{E} \left[ \frac{\partial^2 \mathcal{H}_{n,p}}{\partial (\sigma^2)^2} \bigg|_{\eta = \eta^*} \right] = (2\pi)^{-\alpha/2}(\sigma^9)^{-\alpha+4} \frac{(2 + \alpha^2)}{4(1 + \alpha)^{3/2}}.
\]

Again assumption (B4) implies that \( \inf_{n,p} \mathbb{E} \left[ \frac{\partial^2 \mathcal{H}_{n,p}}{\partial (\sigma^2)^2} \bigg|_{\eta = \eta^*} \right] > 0. \)

Let us consider the \((n + p) \times (n + p)\) matrix \( \Psi_{n,p} \) whose \((k_1, k_2)\)-th element is given by

\[
\mathbb{E} \left[ \frac{\partial^2 \mathcal{H}_{n,p}}{\partial \eta_{k_1} \partial \eta_{k_2}} \bigg|_{\eta = \eta^*} \right].
\]

Then, the previous calculations imply that \( \Psi_{n,p} \) is of full rank for any \( n \) and \( p \), and also \( \inf_{n,p} \text{tr}(\Psi_{n,p}) > 0 \), where \( \text{tr}(A) \) denotes the trace of the matrix \( A \). Also, due to assumption (B1), the integrated likelihood can be differentiated and the derivative can be obtained under the integral sign. Therefore, \( \mathbb{E} \left[ \frac{\partial^2 \mathcal{H}_{n,p}}{\partial \eta_{k_1} \partial \eta_{k_2}} \bigg|_{\eta = \eta^*} \right] = \frac{\partial \mathbb{E}(\Psi_{n,p})}{\partial \eta_{k_1} \partial \eta_{k_2}} \bigg|_{\eta = \eta^*} \), and hence, \( \Psi_{n,p} \) is the Fisher’s information matrix at the true parameter \( \eta^\theta \). Therefore, the matrix \( \Psi_{n,p} \) is positive semi-definite for each \( n \) and \( p \). Together with the result on full rank of \( \Psi_{n,p} \) and \( \inf_{n,p} \text{tr}(\Psi_{n,p}) > 0 \), it implies that \( \Psi_{n,p} \) is positive definite for each \( n \) and \( p \), and \( \inf_{n,p} \) minimum eigenvalue of \( \Psi_{n,p} = \lambda_0 > 0 \).

Now, we decompose \( S_2 \) by considering elements of \( \Psi_{n,p} \) as follows

\[
\sum_{k_1, k_2} \left[ \frac{\partial^2 \mathcal{H}_{n,p}}{\partial \eta_{k_1} \partial \eta_{k_2}} \bigg|_{\eta = \eta^*} \right] (\eta_{k_1} - \eta^\theta_{k_1})(\eta_{k_2} - \eta^\theta_{k_2})
\]

\[
= \sum_{k_1, k_2} \left[ \frac{\partial^2 \mathcal{H}_{n,p}}{\partial \eta_{k_1} \partial \eta_{k_2}} \bigg|_{\eta = \eta^*} \right] - \mathbb{E} \left[ \frac{\partial^2 \mathcal{H}_{n,p}}{\partial \eta_{k_1} \partial \eta_{k_2}} \bigg|_{\eta = \eta^*} \right] (\eta_{k_1} - \eta^\theta_{k_1})(\eta_{k_2} - \eta^\theta_{k_2})
\]

\[
+ \sum_{k_1, k_2} \mathbb{E} \left[ \frac{\partial^2 \mathcal{H}_{n,p}}{\partial \eta_{k_1} \partial \eta_{k_2}} \bigg|_{\eta = \eta^*} \right] (\eta_{k_1} - \eta^\theta_{k_1})(\eta_{k_2} - \eta^\theta_{k_2}).
\]

Now, applying an orthogonal transformation on \((\eta - \eta^\theta)\) to express it as a linear combination of eigenvectors of \( \Psi_{n,p} \), it follows that the second term can be made greater than or equal to \( \lambda_0 r^2 \). Also, it is evident that the first summation has expected value equal to 0. By a similar routine calculation, one can show that the variance of the first term goes to 0 as well. Therefore, for sufficiently large \( n \) and \( p \), with probability tending to 1, \( S_2 > (-r^3 + \lambda_0 r^2) \).

Finally, turning to \( S_3 \), we note that the expected values of the third order derivatives are bounded as shown below.

(S3.14)

\[
\mathbb{E} \left[ \frac{\partial^3 \mathcal{H}_{n,p}}{\partial \lambda^3} \bigg|_{\eta = \eta^*} \right] = M_1 \times \left( \frac{\sigma^9}{np} \right)^{-(\alpha+2)} \sum_{i,j} (a_i^\theta b_j^\theta)^3 \leq M_1 \times \left( \frac{\sigma^9}{np} \right)^{-(\alpha+2)},
\]

(S3.15)

\[
\mathbb{E} \left[ \frac{\partial^3 \mathcal{H}_{n,p}}{\partial \alpha_i^3} \bigg|_{\theta = \theta^*} \right] = M_2 \times \left( \frac{\sigma^9}{np} \right)^{-(\alpha+2)} \sum_{j} (\lambda_j^\theta b_j^\theta)^3 \leq M_2 \times \left( \lambda_j^\theta \right)^3 \left( \frac{\sigma^9}{np} \right)^{-(\alpha+2)},
\]
\[
E \left[ \frac{\partial^3 \mathcal{H}_{n,p}}{\partial b^3_j} \right]_{\theta = \theta^0} = M_3 \times \frac{(\mathbf{g}^{\theta})^{-(\alpha + 2)}}{np} \sum_i (\lambda^j a^j_i)^3 \leq M_3 \times \frac{(\mathbf{g}^{\theta})^{-(\alpha + 2)}}{np},
\]

where \(M_1, M_2, M_3\) are some finite constant. The inequalities follow from Cauchy-Schwartz inequality and the observation that \(\sum_i (a^j_i)^2 = \sum_j (b^j_j)^2 = 1\). Because of assumption (B4), the quantity \(\frac{(\mathbf{g}^{\theta})^{-(\alpha + 2)}}{np} = \Theta(1)\). Since these bounds now do not depend on the particular choice of \(i\) (or \(j\)), one can show that these expected values of third order derivatives are uniformly bounded for any \(n\) and \(p\). This would imply that the third order derivatives \(\frac{\partial^{3 \mathcal{H}_{n,p}}}{\partial \mathbf{g}^3} \) as random variables remain uniformly bounded for \(\eta^1, \eta^2, \eta^3 \in \{\lambda, \alpha_1, \ldots, \alpha_{(n-1)}, \beta_1, \ldots, \beta_{(p-1)}\}\). Let us denote this bound by \(\tilde{M}_1\).

However, the expectation of the third order derivative of \(\mathcal{H}_{n,p}\) with respect to \(\mathbf{g}^{\theta}\) is not finite in view of assumption (B4), since it is \(\Theta\left((\mathbf{g}^{\theta})^{-(\alpha + 4)}/np\right)\). To circumvent this problem, we use assumption (B5) that the random entries \(X_{ij}\) of the data matrix are uniformly bounded for any sufficiently large \(n\) and \(p\) with probability tending to 1. Therefore, \(\frac{\partial^{3 \mathcal{H}_{n,p}}}{\partial \mathbf{g}^3} \) being a continuous function of the entries \(X_{ij}\), also remains uniformly bounded for any sufficiently large \(n\) and \(p\) with probability tending to 1. We denote this bound by \(\tilde{M}_2\).

Combining these two bounds and taking \(M = \max\{M_1, M_2\}\), we obtain that \(S_3 \leq M \sum_{k_1, k_2, k_3} (\eta^1 - \eta^2_{k_1})(\eta^2_{k_2} - \eta^2_{k_3})(\eta^3_{k_3} - \eta^3_{k_1})\) for sufficiently large \(n\) and \(p\), with probability tending to 1. Since it is well known that in finite dimensional vector space the usual \(L_p\) norms are equivalent, therefore, \(\sum_{k_1, k_2, k_3} (\eta^1 - \eta^2_{k_1})(\eta^2_{k_2} - \eta^2_{k_3})(\eta^3_{k_3} - \eta^3_{k_1}) \leq C r^3\) for some finite constant \(C\). Therefore, for sufficiently large \(n\) and \(p\), \(S_3 \leq MC r^3\), with probability tending to 1.

Therefore, for sufficiently large \(n\) and \(p\)

\[
\mathcal{H}_{n,p}(\eta) - \mathcal{H}_{n,p}(\eta^0) > (-4r^3 + 4r^2 - r^3 - MC r^3) = \lambda_0 r^2 - (MC + 5)r^3,
\]

with probability tending to 1. Now, a choice of \(r < \lambda_0/(MC + 5)\) would ensure that \(\mathcal{H}_{n,p}(\eta) > \mathcal{H}_{n,p}(\eta^0)\) for any \(\eta\) satisfying \(||\eta - \eta^0||_2 = r\) (where \(|| \cdot ||_2\) denotes the Euclidean \(L_2\) norm). This is exactly what we intended to show at the beginning.

Finally, following the discussion at the beginning of the proof, we obtain that \(\eta^*\) is consistent for \(\eta^0\), and by continuous mapping theorem, we also have \(\theta^*\) consistent for \(\theta^0\). This completes the proof. \(\Box\)

**S4. Applications of rSVDdp.**

**S4.1. Analysis of UCSD Background Subtraction Dataset.** UCSD Background Subtraction Dataset (Mahadevan and Vasconcelos, 2010) consists of 18 video sequences from different surveillance cameras, of which we only choose to use 3 for the demonstration, namely the “freeway”, “boats” and “peds” video sequences. For the “freeway” video sequence, we add salt-and-pepper type noise to some of the image frames as tampering, and as shown in Figure 1 of the main paper, the estimated background and foreground obtained by rSVDdp turns out to be free of these noises for non-tampered frames. Here, we shall demonstrate the superiority of rSVDdp over usual SVD procedure for background modelling even without camera tampering by considering “boats” and “peds” video sequences.

Figure S1 shows the true image frames, the estimated background and foreground content by the usual SVD and rSVDdp procedure for the “boats” video sequence. This video captures the movement of a boat passing through a river and the constant undulation of waves
around it serves as noises in the background. Since the background estimated by the usual SVD tends to produce an average background by giving equal importance to each frame, hence it produces some artifacts at the position of the boat. Such an artifact in background also adds noise in the foreground, which is seen by an whitish area in all three foreground estimates. However, the rSVDdpd removes such noise and produces clearer background and foreground contents.

The results pertaining to the “peds” video sequence is described in Figure S2. The “peds” video sequence describes the movement of several pedestrians through a road and the shadows of the trees beside the road serves as natural noises present in the data. Although the background estimated by the usual SVD is not visually very different from the background estimated by the proposed rSVDdpd algorithm, the foreground estimates are different. Due to the averaging effect of usual SVD procedure, the foregrounds show the paths of the pedestrians as noisy artifacts. The rSVDdpd estimates of foreground contents are free of such noisy paths.
S4.2. Analysis of University of Houston Camera Tampering Dataset. University of Houston Camera Tampering Detection Dataset (UHCTD) is a large scale video surveillance dataset by Mantini and Shah (2019a) consisting of surveillance videos of over 288 hours ranging across 6 days from two cameras. Three types of camera tampering method has been synthesized in the dataset; (a) covered, (b) defocused, and, (c) moved, which are done uniformly over data to capture changing illumination between day and night. The results of foreground and background extraction for two of the short video clips have been shown in the main paper. Here, we shall consider two more examples which we shall refer to as Video Scenes 3 and 4 respectively.

In Video Scene 3, a noisy image is artificially synthesized to obstruct the view of the camera. As the results show in Figure S3, the background estimated from the usual SVD shows a clear indication of the noise similar to a bump map, resulting in noisy foreground contents even for the frames where camera tampering is not induced. Thus, the usual SVD based
Fig S3: The true images, estimated background and foreground by the usual SVD and the proposed rSVDdpd for Video Scene 3

procedure gives a false positive detection of tampering. However, rSVDdpd outputs a much robust and noise-free background, and the non-tampered frames remain less affected.

Figure S4 shows the results for Video Scene 4. Even if the camera is moved during the video to record two distinguished scenes, the change in the background is not intended (or natural), hence a rank one approximation should be used to model the background. In such situation, SVD tends to produce a background that correlates with all frames with equal importance, while rSVDdpd tends to pick the dominant background only. Thus, any unauthorized change in the background can be identified by looking at the estimated foreground produced by rSVDdpd, namely, the frame where the estimated foreground tends to produce certain artifacts are tampered frames. If SVD is used instead to estimate the foreground, then
all of the frames produces certain artifacts and thus a false positive occurs. As seen from Figure S4, the foreground contents as estimated by rSVDdpd remain fairly truthful in untampered frames, while the performance of the decomposition degrades in the tampered frames. Since all of the information about that particular frame is lost due to moved camera tampering, a regularization involving smoothness of the entries in the singular vector may prove more useful here.

S4.3. **Financial Application: Stock Price Latent Factor Analysis.** In finance, the price of a stock is generally assumed to be composed of two components, a latent market risk
specific to the economy of the whole country or the industry at an aggregate level, and an asset specific risk particular to the company at the individual level (Wang, 2017).

In order to estimate the latent market effect, a spatio-temporal data matrix $X_{T \times p}$ can be constructed with the time series data on the stock prices for $p$ stocks for $T$ timepoints. The entries $X_{ij}$ of the matrix $X$ denotes the price of $j$-th stock at the $i$-th timepoint in consideration. Next, we assume a rank one decomposition of the $X$ matrix as

$$X_{ij} \approx \lambda_i u_i v_j, \quad i = 1, 2, \ldots T, \ j = 1, 2, \ldots p,$$

where the relationship is approximate due to the existence of volatility in stock prices. Hence, the components of the left singular vector can be approximated as $u_i \approx \frac{1}{T} \sum_{j=1}^{p} X_{ij} v_j$, which is a linear combination of the prices of all the stocks at timepoint $i$. On the other hand, the components of the right singular vector can be expressed as $v_j \approx \frac{1}{T} \sum_{i=1}^{T} X_{ij} u_i$, which is again a linear combination of the prices of the $j$-th stock over the entire time horizon. Hence, the quantity $u_i$ can be used to denote an overall market performance indicator at timepoint $i$ and $v_j$ can be used to denote the market capitalization of stock $j$ throughout the entire period of study. Since the stock prices can change drastically, the standard SVD often fails to capture the true decomposition of $X$. To illustrate this point, a dataset containing the daily closing prices of top 50 stocks ($p = 50$) listed in the National Stock Exchange (NSE) of India was collected during the 6 month period from April 1, 2020 to September 30, 2020 ($T = 182$ timepoints). Based on these stock prices, we construct the spatio-temporal matrix $X_{182 \times 50}$ and perform the usual SVD along with the two existing methods of robust SVD and the proposed rSVDdpd algorithm. Figure S5a depicts the time series plots of closing prices of all 50 stocks. All the stocks retain their performances during the entire period of time except for Eicher Motors, which shows a drastic fall in its stock-price due to their loss in sales during the lockdown imposed for the global Coronavirus (COVID-19) pandemic. NIFTY50, a commonly used good measure of the market conditions, does not show a downfall, since the firms other than Eicher Motors do not exhibit such a downfall in their respective stock prices; so the latent market index also should exhibit a similar pattern as NIFTY50. The standard SVD based estimation of the latent market index, obtained through the rank-one decomposition of these data (on 50 stocks), gets heavily affected by the outlying values of Eicher Motor’s stock-prices and differs significantly from the NIFTY50 index; see Figure S5b for illustration. In contrast, all of the robust SVD estimates, namely the ones obtained by pcaSVD (Stacklies et al., 2007), RobSVD, RobRSVD (Zhang and Pan, 2013) and our proposed algorithm rSVDdpd mostly agree with the NIFTY50 index across the entire time horizon in consideration. While the output of RobRSVD is found to have smoothness properties, the NIFTY50 index does not exhibit such smoothness since the stock prices are highly volatile. In this regard, the other robust algorithms produce better estimates of the latent market index.

S5. Detailed Results of Simulation Studies. Table S1-S7 contain detailed results about the bias, MSE and dissimilarity measures for all singular values obtained from the usual SVD and the robust SVD techniques such as pcaSVD, RobRSVD and the proposed rSVDdpd, under all simulation setups (S1)-(S5).

REFERENCES

Aharon, M., Elad, M. and Bruckstein, A. (2006). K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on signal processing 54 4311–4322.

Alman, J. and Williams, V. V. (2021). A refined laser method and faster matrix multiplication. In Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA) 522–539. SIAM.

2https://www.kaggle.com/rohanrao/nifty50-stock-market-data/version/12
### TABLE S1
Comparison of performance measures for Model (S1)

(a) Bias and MSE of estimated singular values

| Method       | Measure | Singular Values | Total |
|--------------|---------|-----------------|-------|
|              |         | $\lambda_1$    | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |       |
| Usual SVD    | Bias    | 0.646           | 1.066     | 1.023        | 2.314        | 7.957  |
|              | MSE     | 1.34            | 1.82      | 1.575        | 5.72         | 10.456 |
| pcaSVD       | Bias    | 0.14            | 0.789     | 1.578        | 3.455        | 15.066 |
|              | MSE     | 2.449           | 2.967     | 4.568        | 14.257       | 24.242 |
| RobSVD       | Bias    | 0.6             | 1.072     | 1.119        | 2.459        | 8.808  |
|              | MSE     | 1.397           | 1.896     | 1.84         | 6.475        | 11.608 |
| RobRSVD      | Bias    | 0.255           | 0.191     | 0.004        | 1.258        | 1.684  |
|              | MSE     | 1.17            | 1.389     | 1.844        | 2.463        | 6.865  |
| $\alpha = 0$ | Bias    | 0.646           | 1.065     | 1.023        | 2.314        | 7.957  |
|              | MSE     | 1.34            | 1.82      | 1.575        | 5.72         | 10.455 |
| $\alpha = 0.1$ | Bias    | 0.643           | 1.065     | 1.023        | 2.314        | 7.952  |
|              | MSE     | 1.339           | 1.821     | 1.576        | 5.72         | 10.456 |
| $\alpha = 0.3$ | Bias    | 0.627           | 1.066     | 1.025        | 2.315        | 7.94   |
|              | MSE     | 1.335           | 1.832     | 1.584        | 5.723        | 10.473 |
| $\alpha = 0.5$ | Bias    | 0.598           | 1.067     | 1.032        | 2.316        | 7.925  |
|              | MSE     | 1.347           | 1.864     | 1.611        | 5.731        | 10.553 |
| $\alpha = 0.7$ | Bias    | 0.566           | 1.056     | 1.042        | 2.318        | 7.894  |
|              | MSE     | 1.372           | 1.884     | 1.678        | 5.746        | 10.68  |
| $\alpha = 1$ | Bias    | 0.535           | 0.998     | 1.04         | 2.324        | 7.764  |
|              | MSE     | 1.459           | 1.845     | 1.73         | 5.806        | 10.841 |

(b) Dissimilarity scores of estimated singular values

| Method       | Type      | Average dissimilarity scores | Total |
|--------------|-----------|------------------------------|-------|
|              |           | 1st vector | 2nd vector | 3rd vector |
| Usual SVD    | Left      | 0.053      | 0.232      | 0.415      | 0.701 |
|              | Right     | 0.026      | 0.15       | 0.242      | 0.418 |
| pcaSVD       | Left      | 0.165      | 0.433      | 0.6        | 1.198 |
|              | Right     | 0.117      | 0.388      | 0.463      | 0.968 |
| RobSVD       | Left      | 0.068      | 0.271      | 0.46       | 0.799 |
|              | Right     | 0.039      | 0.194      | 0.287      | 0.52  |
| RobRSVD      | Left      | 0.025      | 0.17       | 0.539      | 0.733 |
|              | Right     | 0.017      | 0.144      | 0.354      | 0.514 |
| $\alpha = 0$ | Left      | 0.053      | 0.232      | 0.415      | 0.701 |
|              | Right     | 0.026      | 0.15       | 0.242      | 0.418 |
| $\alpha = 0.1$ | Left     | 0.054      | 0.232      | 0.415      | 0.702 |
|              | Right     | 0.027      | 0.15       | 0.242      | 0.419 |
| $\alpha = 0.3$ | Left     | 0.057      | 0.234      | 0.416      | 0.707 |
|              | Right     | 0.029      | 0.152      | 0.243      | 0.425 |
| $\alpha = 0.5$ | Left     | 0.062      | 0.239      | 0.416      | 0.717 |
|              | Right     | 0.033      | 0.156      | 0.244      | 0.433 |
| $\alpha = 0.7$ | Left     | 0.067      | 0.247      | 0.42       | 0.734 |
|              | Right     | 0.038      | 0.163      | 0.248      | 0.449 |
| $\alpha = 1$ | Left      | 0.077      | 0.262      | 0.432      | 0.771 |
|              | Right     | 0.045      | 0.178      | 0.264      | 0.487 |
### Table S2

**Comparison of performance measures for Model (S2a)**

(a) Bias and MSE of the estimated singular values

| Method   | Measure | Singular Values | Total |
|----------|---------|-----------------|-------|
| **Usual SVD** | Bias    | $\lambda_1$ 18.535 | $\lambda_2$ 11.652 | $\lambda_3$ 5.166 | $\lambda_4$ 3.652 | 519.35 |
|          | MSE     | 427.262         | 217.524         | 65.886         | 19.159         | 729.83 |
| **pcaSVD** | Bias    | 8.596           | 9.404           | 9.911           | 9.479           | 350.402 |
|          | MSE     | 207.09          | 195.081         | 208.36          | 183.081         | 793.612 |
| **RobSVD** | Bias    | 18.313          | 11.929          | 5.495           | 3.999           | 523.859 |
|          | MSE     | 420.638         | 228.197         | 75.592          | 23.616          | 748.043 |
| **RobRSVD** | Bias    | 3.763           | 0.661           | 0.105           | 1.751           | 17.672 |
|          | MSE     | 81.608          | 18.362          | 7.211           | 7.621           | 114.802 |
| $\alpha = 0$ | Bias    | 18.535          | 11.652          | 5.166           | 3.652           | 519.347 |
|          | MSE     | 427.257         | 217.527         | 65.888          | 19.16           | 729.833 |
| $\alpha = 0.1$ | Bias    | 11.431          | 10.169          | 6.399           | 4.362           | 294.047 |
|          | MSE     | 250.391         | 213.492         | 114.521         | 44.415          | 622.819 |
| $\alpha = 0.3$ | Bias    | 2.697           | 2.01            | 1.237           | 2.301           | 18.138 |
|          | MSE     | 49.544          | 25.909          | 8.993           | 7.449           | 91.894 |
| $\alpha = 0.5$ | Bias    | 1.852           | 1.327           | 0.957           | 2.184           | 10.877 |
|          | MSE     | 28.118          | 8.497           | 2.263           | 5.138           | 44.016 |
| $\alpha = 0.7$ | Bias    | 1.535           | 1.227           | 0.909           | 2.178           | 9.433 |
|          | MSE     | 20.119          | 5.647           | 1.441           | 5.121           | 32.329 |
| $\alpha = 1$ | Bias    | 1.353           | 1.165           | 0.925           | 2.175           | 8.779 |
|          | MSE     | 15.98           | 4.498           | 1.542           | 5.125           | 27.144 |

(b) Dissimilarity scores of the estimated singular vectors

| Method   | Type | Average dissimilarity scores |
|----------|------|------------------------------|
|          | 1st vector | 2nd vector | 3rd vector | Total |
| **Usual SVD** | Left | 0.591 | 0.676 | 0.663 | 1.93 |
|          | Right | 0.461 | 0.527 | 0.541 | 1.529 |
| **pcaSVD** | Left | 0.387 | 0.616 | 0.677 | 1.679 |
|          | Right | 0.284 | 0.494 | 0.509 | 1.286 |
| **RobSVD** | Left | 0.603 | 0.664 | 0.666 | 1.933 |
|          | Right | 0.464 | 0.507 | 0.545 | 1.516 |
| **RobRSVD** | Left | 0.119 | 0.449 | 0.756 | 1.324 |
|          | Right | 0.106 | 0.395 | 0.573 | 1.074 |
| $\alpha = 0$ | Left | 0.592 | 0.676 | 0.662 | 1.93 |
|          | Right | 0.461 | 0.527 | 0.541 | 1.529 |
| $\alpha = 0.1$ | Left | 0.408 | 0.578 | 0.627 | 1.613 |
|          | Right | 0.283 | 0.419 | 0.461 | 1.163 |
| $\alpha = 0.3$ | Left | 0.123 | 0.325 | 0.497 | 0.944 |
|          | Right | 0.069 | 0.228 | 0.317 | 0.615 |
| $\alpha = 0.5$ | Left | 0.102 | 0.307 | 0.487 | 0.896 |
|          | Right | 0.056 | 0.213 | 0.307 | 0.576 |
| $\alpha = 0.7$ | Left | 0.097 | 0.304 | 0.484 | 0.885 |
|          | Right | 0.054 | 0.211 | 0.306 | 0.571 |
| $\alpha = 1$ | Left | 0.102 | 0.314 | 0.488 | 0.904 |
|          | Right | 0.058 | 0.221 | 0.311 | 0.59 |
## TABLE S3
Comparison of performance measures for Model (S2b)

| Method      | Measure       | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ | Total    |
|-------------|---------------|--------------|--------------|--------------|--------------|----------|
|             | Bias          | 27.882       | 20.696       | 12.304       | 6.342        | 1397.36  |
|             | MSE           | 857.868      | 502.381      | 231.016      | 70.562       | 1661.82  |
| pcaSVD      | Bias          | 19.394       | 15.586       | 16.372       | 14.115       | 1086.341 |
|             | MSE           | 581.39       | 393.275      | 402.921      | 328.873      | 1706.458 |
| RobSVD      | Bias          | 27.336       | 21.254       | 13.314       | 7.192        | 1427.869 |
|             | MSE           | 827.265      | 530.624      | 269.856      | 94.701       | 1722.445 |
| RobRSVD     | Bias          | 7.026        | 1.939        | 0.925        | 2.475        | 60.112   |
|             | MSE           | 185.595      | 50.699       | 22.239       | 20.053       | 278.585  |
| $\alpha = 0$| Bias          | 27.882       | 20.696       | 12.304       | 6.342        | 1397.36  |
|             | MSE           | 857.852      | 502.397      | 231.012      | 70.562       | 1661.82  |
| $\alpha = 0.1$| Bias       | 21.078       | 18.698       | 13.559       | 7.898        | 1039.977 |
|             | MSE           | 595.058      | 493.655      | 309.853      | 130.808      | 1529.374 |
| $\alpha = 0.3$| Bias       | 8.512        | 5.714        | 3.577        | 3.546        | 130.472  |
|             | MSE           | 236.939      | 137.621      | 76.999       | 42.781       | 494.34   |
| $\alpha = 0.5$| Bias       | 5.101        | 3.027        | 1.666        | 2.499        | 44.154   |
|             | MSE           | 132.838      | 63.568       | 25.129       | 16.145       | 237.68   |
| $\alpha = 0.7$| Bias       | 4.260        | 2.280        | 1.254        | 2.263        | 30.039   |
|             | MSE           | 106.199      | 36.904       | 14.312       | 9.829        | 167.244  |
| $\alpha = 1$| Bias          | 3.609        | 2.016        | 1.118        | 2.219        | 23.262   |
|             | MSE           | 84.144       | 29.108       | 10.352       | 9.283        | 132.886  |

(a) Bias and MSE of the estimated singular values

| Method      | Type   | Average dissimilarity scores | Total |
|-------------|--------|------------------------------|-------|
|             |        | 1st vector                  | 2nd vector | 3rd vector |       |
|             | Left   | 0.695 | 0.73 | 0.73 | 2.155 |
|             | Right  | 0.558 | 0.552 | 0.593 | 1.702 |
| pcaSVD      | Left   | 0.556 | 0.693 | 0.716 | 1.965 |
|             | Right  | 0.429 | 0.532 | 0.539 | 1.5   |
| RobSVD      | Left   | 0.701 | 0.724 | 0.723 | 2.148 |
|             | Right  | 0.544 | 0.525 | 0.595 | 1.665 |
| RobRSVD     | Left   | 0.195 | 0.636 | 0.841 | 1.672 |
|             | Right  | 0.177 | 0.56  | 0.634 | 1.371 |
| $\alpha = 0$| Left   | 0.695 | 0.73 | 0.73 | 2.155 |
|             | Right  | 0.558 | 0.551 | 0.593 | 1.702 |
| $\alpha = 0.1$| Left | 0.611 | 0.693 | 0.71 | 2.013 |
|             | Right  | 0.457 | 0.505 | 0.538 | 1.5   |
| $\alpha = 0.3$| Left | 0.26  | 0.448 | 0.575 | 1.283 |
|             | Right  | 0.162 | 0.336 | 0.401 | 0.899 |
| $\alpha = 0.5$| Left | 0.176 | 0.388 | 0.549 | 1.113 |
|             | Right  | 0.1  | 0.283 | 0.373 | 0.756 |
| $\alpha = 0.7$| Left | 0.16  | 0.378 | 0.546 | 1.084 |
|             | Right  | 0.089 | 0.275 | 0.368 | 0.733 |
| $\alpha = 1$| Left   | 0.153 | 0.38  | 0.545 | 1.078 |
|             | Right  | 0.088 | 0.276 | 0.367 | 0.731 |

(b) Dissimilarity scores of the estimated singular vectors
Table S4
Comparison of performance measures for Model (S2c)

| Method      | Measure | Singular Values | Total       |
|-------------|---------|-----------------|-------------|
|             |         | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |
| Usual SVD   | Bias    | 35.602        | 26.965      | 18.287      | 10.247       | 2434.021 |
|             | MSE     | 1352.447      | 789.122     | 407.999     | 163.122      | 2434.021 |
| pcaSVD      | Bias    | 28.761        | 21.826      | 21.292      | 18.816       | 2110.99  |
|             | MSE     | 1043.578      | 650.362     | 586.33      | 502.305      | 2782.575 |
| RobSVD      | Bias    | 34.449        | 27.626      | 19.894      | 11.971       | 2488.98  |
|             | MSE     | 1269.72       | 829.946     | 479.204     | 224.725      | 2803.595 |
| RobRSVD     | Bias    | 12.191        | 5.003       | 2.619       | 3.624        | 193.664  |
|             | MSE     | 378.895       | 131.611     | 58.818      | 47.77        | 617.094  |

α = 0

| Method      | Measure | Singular Values | Total       |
|-------------|---------|-----------------|-------------|
|             |         | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |
| Usual SVD   | Bias    | 35.602        | 26.965      | 18.286      | 10.248       | 2434.012 |
|             | MSE     | 1352.434      | 789.132     | 407.998     | 163.129      | 2712.693 |
| pcaSVD      | Bias    | 30.207        | 25.765      | 19.503      | 11.723       | 2094.11  |
|             | MSE     | 1071.965      | 788.786     | 498.291     | 228.751      | 2587.792 |

α = 0.1

| Method      | Measure | Singular Values | Total       |
|-------------|---------|-----------------|-------------|
|             |         | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |
| pcaSVD      | Bias    | 18.092        | 13.973      | 9.42        | 7.163        | 662.604  |
|             | MSE     | 638.009       | 427.691     | 246.907     | 140.797      | 1453.404 |

α = 0.3

| Method      | Measure | Singular Values | Total       |
|-------------|---------|-----------------|-------------|
|             |         | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |
| RobSVD      | Bias    | 13.149        | 8.771       | 5.304       | 4.478        | 298.008  |
|             | MSE     | 472.335       | 264.458     | 133.355     | 68.558       | 938.705  |

α = 0.5

| Method      | Measure | Singular Values | Total       |
|-------------|---------|-----------------|-------------|
|             |         | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |
| RobRSVD     | Bias    | 9.561          | 5.28        | 3.218       | 3.379        | 141.064  |
|             | MSE     | 347.343       | 147.051     | 76.771      | 40.99        | 612.155  |

(a) Bias and MSE of estimated singular values

| Method      | Type   | Average dissimilarity scores | Total |
|-------------|--------|-----------------------------|-------|
|             |        | 1st vector | 2nd vector | 3rd vector |       |
| Usual SVD   | Left   | 0.738      | 0.736      | 0.732      | 2.206 |
|             | Right  | 0.615      | 0.546      | 0.568      | 1.73  |
| pcaSVD      | Left   | 0.656      | 0.725      | 0.729      | 2.11  |
|             | Right  | 0.53       | 0.547      | 0.555      | 1.633 |
| RobSVD      | Left   | 0.735      | 0.738      | 0.73       | 2.203 |
|             | Right  | 0.586      | 0.526      | 0.576      | 1.688 |
| RobRSVD     | Left   | 0.301      | 0.754      | 0.878      | 1.932 |
|             | Right  | 0.274      | 0.662      | 0.662      | 1.598 |
| α = 0       | Left   | 0.738      | 0.736      | 0.732      | 2.206 |
|             | Right  | 0.615      | 0.547      | 0.568      | 1.73  |
| α = 0.1     | Left   | 0.71       | 0.732      | 0.729      | 2.17  |
|             | Right  | 0.569      | 0.528      | 0.553      | 1.651 |
| α = 0.3     | Left   | 0.439      | 0.579      | 0.648      | 1.666 |
|             | Right  | 0.317      | 0.439      | 0.471      | 1.227 |
| α = 0.5     | Left   | 0.324      | 0.504      | 0.615      | 1.444 |
|             | Right  | 0.218      | 0.381      | 0.443      | 1.042 |
| α = 0.7     | Left   | 0.278      | 0.475      | 0.61       | 1.363 |
|             | Right  | 0.18       | 0.358      | 0.428      | 0.965 |
| α = 1       | Left   | 0.253      | 0.459      | 0.601      | 1.313 |
|             | Right  | 0.159      | 0.343      | 0.429      | 0.93  |

(b) Dissimilarity scores of estimated singular values
### Table S5
Comparison of performance measures for Model (S3)

| Method      | Measure | Singular Values | Total    |
|-------------|---------|-----------------|----------|
|             |         | $\lambda_1$    | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |        |
| Usual SVD   | Bias    | 40.631          | 3.595     | 2.077       | 3.131       | 1677.949 |
|             | MSE     | 1656.192        | 14.731    | 5.096       | 10.342      | 1686.361 |
| pcaSVD      | Bias    | 40.068          | 3.328     | 2.56        | 4.199       | 1640.708 |
|             | MSE     | 1609.613        | 14.505    | 9.399       | 20.982      | 1654.5   |
| RobSVD      | Bias    | 40.639          | 3.463     | 2.259       | 3.356       | 1679.881 |
|             | MSE     | 1657.001        | 13.662    | 6.122       | 11.923      | 1688.708 |
| RobRSVD     | Bias    | 31.138          | 9.506     | 3.708       | 4.073       | 1090.284 |
|             | MSE     | 1044.612        | 132.858   | 38.958      | 31.848      | 1248.276 |
| $\alpha = 0$ | Bias   | 40.185          | 3.821     | 2.083       | 3.146       | 1643.663 |
|             | MSE     | 1618.95         | 16.293    | 5.116       | 10.452      | 1650.81  |
| $\alpha = 0.1$ | Bias  | 35.952          | 6.898     | 2.257       | 3.157       | 1355.176 |
|             | MSE     | 1443.766        | 167.503   | 11.993      | 11.147      | 1634.409 |
| $\alpha = 0.3$ | Bias  | 9.383           | 4.429     | 1.241       | 2.351       | 114.714  |
|             | MSE     | 331.917         | 111.703   | 7.797       | 6.991       | 458.468  |
| $\alpha = 0.5$ | Bias  | 4.26            | 2.21      | 0.817       | 2.105       | 28.128   |
|             | MSE     | 119.738         | 30.848    | 1.511       | 4.889       | 156.986  |
| $\alpha = 0.7$ | Bias  | 3.619           | 1.886     | 0.791       | 2.076       | 21.594   |
|             | MSE     | 91.007          | 16.125    | 1.454       | 4.743       | 113.33   |
|             |         | 1st vector      | 2nd vector | 3rd vector  |        |
| Usual SVD   | Left    | 0.643           | 0.747     | 0.662       | 2.052      |
|             | Right   | 0.548           | 0.778     | 0.598       | 1.924      |
| pcaSVD      | Left    | 0.647           | 0.696     | 0.66        | 2.003      |
|             | Right   | 0.581           | 0.669     | 0.585       | 1.836      |
| RobSVD      | Left    | 0.637           | 0.674     | 0.63        | 1.941      |
|             | Right   | 0.552           | 0.706     | 0.574       | 1.832      |
| RobRSVD     | Left    | 0.619           | 0.67      | 0.872       | 2.162      |
|             | Right   | 0.533           | 0.82      | 0.823       | 2.175      |
| $\alpha = 0$ | Left   | 0.682           | 0.778     | 0.652       | 2.113      |
|             | Right   | 0.587           | 0.81      | 0.588       | 1.985      |
| $\alpha = 0.1$ | Left  | 0.613           | 0.745     | 0.649       | 2.007      |
|             | Right   | 0.509           | 0.753     | 0.582       | 1.844      |
| $\alpha = 0.3$ | Left  | 0.205           | 0.437     | 0.565       | 1.208      |
|             | Right   | 0.143           | 0.349     | 0.403       | 0.895      |
| $\alpha = 0.5$ | Left  | 0.133           | 0.366     | 0.525       | 1.024      |
|             | Right   | 0.085           | 0.262     | 0.346       | 0.692      |
| $\alpha = 0.7$ | Left  | 0.128           | 0.36      | 0.523       | 1.012      |
|             | Right   | 0.083           | 0.255     | 0.34        | 0.678      |
| $\alpha = 1$ | Left    | 0.127           | 0.355     | 0.519       | 1.002      |
|             | Right   | 0.082           | 0.25      | 0.335       | 0.667      |

(a) Bias and MSE of estimated singular values

(b) Dissimilarity scores of estimated singular values
| Method  | Measure | singular values | Total  |
|---------|---------|----------------|--------|
|         | $\lambda_1$ | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |        |
| Usual SVD | Bias | 203.281 | 19.996 | 8.243 | 5.888 | 41825.788 |
|          | MSE   | 2168439.619 | 3083.955 | 131.383 | 42.08 | 2171697.037 |
| pcaSVD   | Bias | 84.392 | 58.404 | 46.287 | 39.353 | 14224.145 |
|          | MSE   | 1393235.11 | 399208.585 | 128128.542 | 242805.126 | 2163377.363 |
| RobSVD   | Bias | 203.132 | 20.103 | 8.512 | 6.382 | 41779.81 |
|          | MSE   | 2168433.257 | 3090.855 | 137.639 | 49.283 | 2171711.033 |
| RobRSVD  | Bias | 22.765 | 2.866 | 1.22 | 3.583 | 540.799 |
|          | MSE   | 24849.948 | 987.545 | 504.485 | 2807.752 | 29149.731 |
| $\alpha = 0$ | Bias | 203.281 | 19.996 | 8.243 | 5.888 | 41825.787 |
|          | MSE   | 2168439.618 | 3083.956 | 131.383 | 42.08 | 2171697.037 |
| $\alpha = 0.1$ | Bias | 16.52 | 10.781 | 7.025 | 5.568 | 469.522 |
|          | MSE   | 1294.238 | 192.846 | 91.297 | 51.104 | 1629.485 |
| $\alpha = 0.3$ | Bias | 11.827 | 8.236 | 5.736 | 4.951 | 265.135 |
|          | MSE   | 1002.537 | 109.666 | 52.163 | 33.515 | 1197.881 |
| $\alpha = 0.5$ | Bias | 10.043 | 7.175 | 4.995 | 4.639 | 198.809 |
|          | MSE   | 702.602 | 90.165 | 38.326 | 27.947 | 859.039 |
| $\alpha = 0.7$ | Bias | 9.347 | 6.506 | 4.5 | 4.373 | 169.082 |
|          | MSE   | 685.339 | 68.418 | 29.8 | 24.313 | 807.869 |
| $\alpha = 1$ | Bias | 8.607 | 5.796 | 4.054 | 4.068 | 140.645 |
|          | MSE   | 742.3 | 54.543 | 25.196 | 20.731 | 842.77 |

(a) Bias and MSE of estimated singular values

| Method  | Type | Average dissimilarity scores | Total  |
|---------|------|-----------------------------|--------|
|         |      | 1st vector | 2nd vector | 3rd vector |        |
| Usual SVD | Left | 0.648 | 0.712 | 0.728 | 2.089 |
|          | Right | 0.496 | 0.541 | 0.566 | 1.603 |
| pcaSVD   | Left | 0.569 | 0.697 | 0.723 | 1.989 |
|          | Right | 0.415 | 0.533 | 0.541 | 1.489 |
| RobSVD   | Left | 0.655 | 0.712 | 0.728 | 2.095 |
|          | Right | 0.501 | 0.529 | 0.571 | 1.602 |
| RobRSVD  | Left | 0.182 | 0.647 | 0.878 | 1.707 |
|          | Right | 0.175 | 0.527 | 0.601 | 1.303 |
| $\alpha = 0$ | Left | 0.648 | 0.712 | 0.728 | 2.089 |
|          | Right | 0.496 | 0.541 | 0.566 | 1.603 |
| $\alpha = 0.1$ | Left | 0.555 | 0.695 | 0.721 | 1.97 |
|          | Right | 0.417 | 0.529 | 0.55 | 1.496 |
| $\alpha = 0.3$ | Left | 0.486 | 0.678 | 0.713 | 1.877 |
|          | Right | 0.357 | 0.517 | 0.529 | 1.403 |
| $\alpha = 0.5$ | Left | 0.459 | 0.669 | 0.709 | 1.838 |
|          | Right | 0.333 | 0.51 | 0.523 | 1.367 |
| $\alpha = 0.7$ | Left | 0.441 | 0.661 | 0.708 | 1.809 |
|          | Right | 0.323 | 0.509 | 0.523 | 1.355 |
| $\alpha = 1$ | Left | 0.429 | 0.655 | 0.702 | 1.786 |
|          | Right | 0.306 | 0.509 | 0.51 | 1.324 |

(b) Dissimilarity scores of estimated singular values
### Table S7
Comparison of performance measures for Model (S5)

| Method       | Measure  | Singular Values | Total     |
|--------------|----------|-----------------|-----------|
|              |          | $\lambda_1$    | $\lambda_2$ | $\lambda_3$ | $\lambda_4$ |         |
| Usual SVD    | Bias     | 5.502           | 5.526      | 3.932       | 4.166       | 93.633  |
|              | MSE      | 75.251          | 34.51      | 17.7        | 18.726      | 146.187 |
| pcaSVD       | Bias     | 1.748           | 4.438      | 4.494       | 5.878       | 77.499  |
|              | MSE      | 11.454          | 26.444     | 27.979      | 42.636      | 108.514 |
| RobSVD       | Bias     | 5.224           | 5.469      | 4.16        | 4.516       | 94.901  |
|              | MSE      | 72.821          | 34.254     | 19.879      | 22.176      | 149.113 |
| RobRSVD      | Bias     | 1.721           | 3.882      | 2.148       | 2.548       | 29.135  |
|              | MSE      | 14.026          | 17.453     | 8.152       | 9.416       | 49.046  |
| $\alpha = 0$| Bias     | 3.719           | 5.112      | 3.713       | 4.072       | 70.339  |
|              | MSE      | 22.628          | 29.023     | 15.716      | 17.909      | 84.917  |
| $\alpha = 0.1$| Bias    | 3.499           | 5.219      | 3.741       | 4.062       | 69.982  |
|              | MSE      | 19.584          | 31.25      | 16.384      | 17.885      | 85.102  |
| $\alpha = 0.3$| Bias    | 2.811           | 5.248      | 3.844       | 4.108       | 67.092  |
|              | MSE      | 14.035          | 32.156     | 18.157      | 18.924      | 83.237  |
| $\alpha = 0.5$| Bias    | 2.296           | 5.092      | 3.772       | 4.123       | 62.429  |
|              | MSE      | 11.095          | 30.669     | 17.611      | 19.454      | 78.83   |
| $\alpha = 0.7$| Bias    | 1.903           | 4.858      | 3.628       | 4.032       | 56.64   |
|              | MSE      | 9.166           | 28.283     | 16.5        | 18.511      | 72.46   |
| $\alpha = 1$| Bias     | 1.465           | 4.568      | 3.364       | 3.885       | 49.424  |
|              | MSE      | 7.873           | 25.638     | 14.682      | 17.49       | 65.682  |

(a) Bias and MSE of estimated singular values

| Method       | Type  | Average dissimilarity scores | Total |
|--------------|-------|------------------------------|-------|
|              |       | 1st vector                  | 2nd vector | 3rd vector |       |
| Usual SVD    | Left  | 0.503                        | 0.792     | 0.725      | 2.02  |
|              | Right | 0.44                         | 0.74      | 0.604      | 1.785 |
| pcaSVD       | Left  | 0.624                        | 0.71      | 0.7        | 2.034 |
|              | Right | 0.543                        | 0.667     | 0.6        | 1.81  |
| RobSVD       | Left  | 0.521                        | 0.732     | 0.716      | 1.969 |
|              | Right | 0.447                        | 0.677     | 0.61       | 1.734 |
| RobRSVD      | Left  | 0.338                        | 0.853     | 0.777      | 1.967 |
|              | Right | 0.331                        | 0.846     | 0.647      | 1.824 |
| $\alpha = 0$| Left  | 0.459                        | 0.794     | 0.718      | 1.971 |
|              | Right | 0.404                        | 0.748     | 0.608      | 1.759 |
| $\alpha = 0.1$| Left   | 0.462                       | 0.788     | 0.718      | 1.968 |
|              | Right | 0.406                        | 0.74      | 0.607      | 1.754 |
| $\alpha = 0.3$| Left   | 0.467                       | 0.773     | 0.714      | 1.954 |
|              | Right | 0.412                        | 0.723     | 0.609      | 1.744 |
| $\alpha = 0.5$| Left   | 0.472                       | 0.761     | 0.71       | 1.943 |
|              | Right | 0.42                         | 0.713     | 0.601      | 1.734 |
| $\alpha = 0.7$| Left   | 0.486                       | 0.75      | 0.703      | 1.939 |
|              | Right | 0.43                         | 0.707     | 0.594      | 1.731 |
| $\alpha = 1$| Left  | 0.503                        | 0.74      | 0.691      | 1.934 |
|              | Right | 0.447                        | 0.702     | 0.578      | 1.726 |

(b) Dissimilarity scores of estimated singular values
(a) Time series data on Indian stock prices with NIFTY50 and Eicher Motors stock prices in highlight.

(b) Comparison of latent market index obtained from NIFTY50, the usual SVD method and the different robust SVD methods (in units of rupees).

Fig S5: Latent stock market index estimation: The top panel presents time series data on closing prices of top 50 stocks in National Stock Exchange (NSE) of India while the bottom panel presents the estimated latent market index based on the non-robust usual SVD method and the robust SVD methods (in unit of rupees).

AMMANN, L. P. (1993). Robust Singular Value Decompositions: A New Approach to Projection Pursuit. *Journal of the American Statistical Association* 88, 505–514.

ANANDARAJAN, M., HILL, C. and NOLAN, T. (2019). Semantic Space Representation and Latent Semantic Analysis. In *Practical Text Analytics* 77–91. Springer.

ARGYROS, I. K. (2008). *Convergence and Applications of Newton-type Iterations*. Springer New York.

BASAK, S., BASU, A. and JONES, M. C. (2021). On the ‘optimal’ density power divergence tuning parameter. *Journal of Applied Statistics* 48, 536–556.

BASU, A., HARRIS, I. R., HJORT, N. L. and JONES, M. C. (1998). Robust and Efficient Estimation by Minimising a Density Power Divergence. *Biometrika* 85, 549–559.
Bouwmans, T. and Zahzah, E. H. (2014). Robust PCA via Principal Component Pursuit: A review for a comparative evaluation in video surveillance. Computer Vision and Image Understanding 122 22 - 34.

Bustamam, A., Formaldin, S. and Siswanti, T. (2018). Clustering and analyzing microarray data of lymphoma using singular value decomposition (SVD) and hybrid clustering. In AIP Conference Proceedings 2023 020220. AIP Publishing LLC.

Candes, E. J., Li, X., Ma, Y. and Wright, J. (2011). Robust Principal Component Analysis? J. ACM 58.

Cheng, H., Zhang, Y., Lu, W. and Yang, Z. (2019). A bearing fault diagnosis method based on VMD-SVD and Fuzzy clustering. International Journal of Pattern Recognition and Artificial Intelligence 33 1950018.

Cichocki, A. and Amari, S.-I. (2010). Families of alpha-beta-and gamma-divergences: Flexible and robust measures of similarities. Entropy 12 1532–1568.

Cichocki, A., Cruces, S. and Amari, S.-I. (2011). Generalized Alpha-Beta Divergences and Their Application to Robust Nonnegative Matrix Factorization. Entropy 13.

Dappuri, B., Rao, M. P. and Sikha, M. B. (2020). Non-blind RGB watermarking approach using SVD in translation invariant wavelet space with enhanced Grey-wolf optimizer. Multimedia Tools and Applications 1–22.

Das, L., Das, J. K. and Nanda, S. (2017). Advanced protein coding region prediction applying robust SVD algorithm. In 2017 2nd International Conference on Man and Machine Interfacing (MAMI) 1-6.

Drineas, P., Frieze, A., Kannan, R., Vempala, S. and Vinay, V. (2004). Clustering large graphs via the singular value decomposition. Machine learning 56 9–33.

Fithian, W. and Josse, J. (2017). Multiple correspondence analysis and the multilogit bilinear model. Journal of multivariate analysis 157 87–102.

Franceschini, A., Lin, J., Von Mering, C. and Jensen, L. J. (2016). SVD-phy: improved prediction of protein functional associations through singular value decomposition of phylogenetic profiles. Bioinformatics 32 1085–1087.

Gabriel, K. R. and Zamir, S. (1979). Lower rank approximation of matrices by least squares with any choice of weights. Technometrics 21 489–498.

Gao, Q., Xia, W., Wan, Z., Xie, D. and Zhang, P. (2020). Tensor-SVD Based Graph Learning for Multi-View Subspace Clustering. In Proceedings of the AAAI Conference on Artificial Intelligence 34 3930–3937.

Garcia-Garcia, B., Bouwmans, T. and Silva, A. J. R. (2020). Background subtraction in real applications: Challenges, current models and future directions. Computer Science Review 35 100204.

Ghosh, A. and Basu, A. (2013). Robust estimation for independent non-homogeneous observations using density power divergence with applications to linear regression. Electronic Journal of Statistics 7 2420 – 2456.

Ghosh, A. and Basu, A. (2015). Robust estimation for non-homogeneous data and the selection of the optimal tuning parameter: the density power divergence approach. Journal of Applied Statistics 42 2056–2072.

Giraud, L., Langou, J. and Rozloznik, M. (2005). The loss of orthogonality in the Gram-Schmidt orthogonalization process. Computers & Mathematics with Applications 50 1069–1075.

Golub, G. H. and Reinsch, C. (1971). Singular value decomposition and least squares solutions. In Linear Algebra 134–151. Springer.

Greenacre, M. (2017). Correspondence analysis in practice, CRC press.

Grondin, F. and Glass, J. (2019a). SVD-PHAT: A fast sound source localization method. In ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 4140–4144. IEEE.

Grondin, F. and Glass, J. (2019b). Multiple sound source localization with SVD-PHAT. arXiv preprint arXiv:1906.11913.

Hawkins, D. M., Liu, L. and Young, S. S. (2001). Robust Singular Value Decomposition Technical Report No. 122, National Institute of Statistical Sciences (NISS).

He, X. and Shao, Q.-M. (2000). On parameters of increasing dimensions. Journal of Multivariate Analysis 73 120–135.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval 50–57.

Hong, C. and Kim, Y. (2001). Automatic selection of the turning parameter in the minimum density power divergence estimation. Journal of the Korean Statistical Society 30 453–465.

Horn, D. and Axsel, I. (2003). Novel clustering algorithm for microarray expression data in a truncated SVD space. Bioinformatics 19 1110–1115.

Huang, J. Z., Shen, H. and Buja, A. (2009). The analysis of two-way functional data using two-way regularized singular value decompositions. Journal of the American Statistical Association 104 1609–1620.

Huber, P. J. (1973). Robust Regression: Asymptotics, Conjectures and Monte Carlo. The Annals of Statistics 1 799 – 821.

Huber, P. J. (2004). Robust statistics 523. John Wiley & Sons.
Jung, K.-M. (2010). Robust Singular Value Decomposition Based on Weighted Least Absolute Deviation Regression. Communications for Statistical Applications and Methods 17 803-810.

Kamal, K. K. and Khangar, N. V. (2017). Multiple correspondence analysis and its applications. Electronic Journal of Applied Statistical Analysis 10 432–462.

Ke, Q. and Kanade, T. (2005). Robust L1 Norm Factorization in the Presence of Outliers and Missing Data by Alternative Convex Programming In Proceedings of (CVPR) Computer Vision and Pattern Recognition 1 739 - 746.

Kumar, N., Nasser, M. and Sarker, S. C. (2011). A new singular value decomposition based robust graphical clustering technique and its application in climatic data. Journal of Geography and Geology 3 327.

Liang, F. (2007). Use of SVD-based probit transformation in clustering gene expression profiles. Computational Statistics & Data Analysis 51 6355–6366.

Liu, L., Hawkins, D. M., Ghosh, S. and Young, S. S. (2003). Robust singular value decomposition analysis of microarray data. Proceedings of the National Academy of Sciences 100 13167–13172.

Mahadevan, V. and Vasconcelos, N. (2010). Spatiotemporal Saliency in Dynamic Scenes. IEEE Transactions on Pattern Analysis and Machine Intelligence 32 171-177.

Maheswari, N. and Duraiswamy, K. (2008). CLUST-SVD: Privacy preserving clustering in singular value decomposition. World Journal of Modelling and Simulation 4 250–256.

Malliaros, F. D. and Vazirgiannis, M. (2013). Clustering and community detection in directed networks: A survey. Physics reports 533 95–142.

Mantini, P. and Shah, S. K. (2019a). UHCTD: A Comprehensive Dataset for Camera Tampering Detection. In 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS) 1–8.

Mantini, P. and Shah, S. K. (2019b). Camera Tampering Detection using Generative Reference Model and Deep Learned Features. In VISIGRAPP (5: VISAPP) 85–95.

Mardia, K. V., Kent, J. T., Bibby, J. M. and Holmes, S. (2005). Multivariate Analysis. Probability and Mathematical Statistics Series. Elsevier Academic Press.

Markovsky, I. and Van Huffel, S. (2007). Overview of total least-squares methods. Signal processing 87 2283–2302.

Meenakshi, K., Swaraja, K. and Kora, P. (2020). A hybrid matrix factorization technique to free the watermarking scheme from false positive and negative problems. Multimedia Tools and Applications 1–36.

Mehta, L. M. (2004). Random matrices. Elsevier.

Mirzal, A. (2016). Clustering and latent semantic indexing aspects of the singular value decomposition. International Journal of Information and Decision Sciences 8 53–72.

Özer, H., Sankur, B. and Memon, N. (2005). An SVD-based audio watermarking technique. In Proceedings of the 7th workshop on Multimedia and security 51–56.

Phansalkar, N., More, S., Sabale, A. and Joshi, M. (2011). Adaptive local thresholding for detection of nuclei in diversity stained cytology images. In 2011 International Conference on Communications and Signal Processing 218–220. IEEE.

Portnoy, S. et al. (1984). Asymptotic behavior of M-estimators of p regression parameters when p^2 / n is large. I. Consistency. The Annals of Statistics 12 1298–1309.

Rey, W. (2007). Total singular value decomposition. Robust SVD, regression and location-scale.

Rezaei, A. and Khalili, M. (2019). A Robust Blind Audio Watermarking Scheme Based on DCT-DWT-SVD. In Fundamental Research in Electrical Engineering 101–113. Springer.

Rudelson, M. and Vershynin, R. (2010). Non-asymptotic theory of random matrices: extreme singular values. In Proceedings of the International Congress of Mathematicians 2010 (ICM 2010) (In 4 Volumes) Vol. I: Plenary Lectures and Ceremonies Vols. II–IV: Invited Lectures 1576–1602. World Scientific.

Saied, R. A. (2012). SVD based image processing applications: state of the art, contributions and research challenges. arXiv preprint arXiv:1211.7102.

Sarkar, S. and Dong, A. (2011). Community detection in graphs using singular value decomposition. Physical Review E 83 046114.

Schur, J. (1911). Bemerkungen zur Theorie der beschränkten Bilinearformen mit unendlich vielen Veränderlichen.

Sitara, K. and Mehtre, B. M. (2019). Automated camera sabotage detection for enhancing video surveillance systems. Multimedia Tools and Applications 78 5819–5841.

Stacklies, W., Redestig, H., Schulz, M., Walthers, D. and Selbig, J. (2007). pcaMethods – A Bioconductor package providing PCA methods for incomplete data. Bioinformatics 23 1164–1167.

R Core Team (2021). R: A Language and Environment for Statistical Computing R Foundation for Statistical Computing, Vienna, Austria.

Thakkar, F. N. and Srivastava, V. K. (2017). A blind medical image watermarking: DWT-SVD based robust and secure approach for telemedicine applications. Multimedia Tools and Applications 76 3669–3697.
TRACY, C. A. and WIDOM, H. (1993). Introduction to random matrices. In Geometric and Quantum Aspects of Integrable Systems 103–130. Springer.

WANG, D. (2017). Adjustable Robust Singular Value Decomposition: Design Analysis and Application to Finance. Data 2 29.

WARRICK, J. and JONES, M. (2005). Choosing a robustness tuning parameter. Journal of Statistical Computation and Simulation 75 581–588.

ZEKRI, H., MOKHTARI, A. R. and COHEN, D. R. (2016). Application of singular value decomposition (SVD) and semi-discrete decomposition (SDD) techniques in clustering of geochemical data: an environmental study in central Iran. Stochastic Environmental Research and Risk Assessment 30 1947–1960.

ZHANG, L. and PAN, C. (2013). RobRSVD: Robust Regularized Singular Value Decomposition R package version 1.0.

ZHANG, L., SHEN, H. and HUANG, J. (2013). Robust regularized singular value decomposition with application to mortality data. The Annals of Applied Statistics 7.

ZHANG, A. and XIA, D. (2018). Tensor SVD: Statistical and computational limits. IEEE Transactions on Information Theory 64 7311–7338.

ZHANG, D., GUO, J., LEI, X. and ZHU, C. (2016). Note: Sound recovery from video using SVD-based information extraction. Review of Scientific Instruments 87 086111.

ZHAO, M. and JIA, X. (2017). A novel strategy for signal denoising using reweighted SVD and its applications to weak fault feature enhancement of rotating machinery. Mechanical Systems and Signal Processing 94 129–147.