AN EXTRAPOLATED AND PROVABLY CONVERGENT ALGORITHM FOR NONLINEAR MATRIX DECOMPOSITION WITH THE RELU FUNCTION

NICOLAS GILLIS*, MARGHERITA PORCELLI†, AND GIOVANNI SERAGHITI‡

Abstract. Nonlinear matrix decomposition (NMD) with the ReLU function, denoted ReLU-NMD, is the following problem: given a sparse, nonnegative matrix X and a factorization rank r, identify a rank-r matrix Θ such that $X \approx \max(0, \Theta)$. This decomposition finds application in data compression, matrix completion with entries missing not at random, and manifold learning. The standard ReLU-NMD model minimizes the least squares error, that is, $\|X - \max(0, \Theta)\|_F^2$. The corresponding optimization problem is nondifferentiable and highly nonconvex. This motivated Saul to propose an alternative model, Latent-ReLU-NMD, where a latent variable Z is introduced and satisfies $\max(0, Z) = X$ while minimizing $\|Z - \Theta\|_F^2$ ("A nonlinear matrix decomposition for mining the zeros of sparse data", SIAM J. Math. Data Sci., 2022). Our first contribution is to show that the two formulations may yield different low-rank solutions Θ; in particular, we show that Latent-ReLU-NMD can be ill-posed when ReLU-NMD is not, meaning that there are instances in which the infimum of Latent-ReLU-NMD is not attained while that of ReLU-NMD is. We also consider another alternative model, called 3B-ReLU-NMD, which parameterizes $\Theta = WH$, where W has r columns and H has r rows, allowing one to get rid of the rank constraint in Latent-ReLU-NMD. Our second contribution is to prove the convergence of a block coordinate descent (BCD) applied to 3B-ReLU-NMD and referred to as BCD-NMD. Our third contribution is a novel extrapolated variant of BCD-NMD, dubbed eBCD-NMD, which we prove is also convergent under mild assumptions. We illustrate the significant acceleration effect of eBCD-NMD compared to BCD-NMD, and also show that eBCD-NMD performs well against the state of the art on real-world data sets.

1. Introduction. Nonlinear matrix decomposition (NMD) can be defined as follows: given a data matrix $X \in \mathbb{R}^{n \times m}$, a factorization rank r, NMD looks for a rank-r matrix $\Theta \in \mathbb{R}^{m \times n}$ such that $X \approx f(\Theta)$, where $f(\Theta)$ applies the scalar function f element-wise on Θ, that is, $[f(\Theta)]_{i,j} = f(\Theta_{i,j})$ for all i, j. The choice of f depends on the type of data at hand. For example, one might choose $f(x) = \text{sign}(x)$ for binary data in $\{-1, 1\}$ [11], or if X has a multiplicative structure, one can consider the Hadamard product decomposition with $f(x) = x^2$, which is also efficient for the compression of dense images [9, 18] and for representing probabilistic circuits [27]. In this work, we assume X to be nonnegative and sparse, opting for f to be the ReLU function, $f(\cdot) = \max(0, \cdot)$. This NMD decomposition, with $X \approx \max(0, \Theta)$, is referred to as the ReLU decomposition. Intuitively, the ReLU decomposition aims at finding a matrix Θ that substitutes the zero entries of the target matrix X with negative values in order to decrease its rank. Minimizing the least squares error leads to an optimization problem referred to as ReLU-NMD:

$$\min_{\Theta} \|X - \max(0, \Theta)\|_F^2 \quad \text{such that} \quad \text{rank}(\Theta) \leq r. \tag{1.1}$$

Note that some degree of sparsity on the matrix X is an essential condition for ReLU-NMD to be meaningful in practice. In fact, if X has mostly positive entries, then Θ will also have mostly positive entries and the ReLU nonlinear activation will be active only for few entries, leading to a model that is similar to standard low-rank matrix approximations such as the truncated singular value decomposition (TSVD). However, it is interesting to note that ReLU-NMD will always lead to a lower error than the TSVD. In fact, let X_r be an optimal rank-r approximation of $X \geq 0$ (e.g., obtained with the TSVD), then $\|X - X_r\|_F^2 \geq \|X - \max(X_r, 0)\|_F^2$ since negative entries in X_r, if any, are replaced by zeros which better approximate the corresponding nonnegative entries in X.

ReLU-NMD was introduced by Saul in [34, 33]. It is non-differentiable and non-convex, and hence difficult to tackle. Indeed, to the best of our knowledge, there exists only one algorithm that addresses

*University of Mons, Rue de Houdain 9, 7000 Mons, Belgium. Email: nicolas.gillis@umons.ac.be

†Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Viale Morgagni 40/44, 50134 Firenze, Italia. ISTI–CNR, Via Moruzzi 1, Pisa, Italia. Member of the INdAM Research Group GNCS. Email: margherita.porcelli@unifi.it.

‡Corresponding author. University of Mons, Rue de Houdain 9, 7000 Mons, Belgium, and Dipartimento di Ingegneria Industriale, Università degli Studi di Firenze, Viale Morgagni 40/44, 50134 Firenze, Italia. Member of the INdAM Research Group GNCS. Email: giovanni.seraghiti@umons.ac.be.

(1.1) directly, by parametrizing $\Theta = WH$ and using a coordinate descent approach, optimizing one entry of W and H at a time; see [4]. All other existing approaches rely on Saul's alternative formulation, dubbed Latent-ReLU-NMD, and defined as

$$\min_{Z,\Theta} \|Z - \Theta\|_F^2 \quad \text{such that} \quad \begin{cases} \operatorname{rank}(\Theta) \leq r, \\ \max(0, Z) = X. \end{cases} \tag{1.2}$$

The objective function in (1.2) is differentiable, while the subproblems in variables Z and Θ can be solved up to global optimality; see Section 4. However, for a general matrix X, the link between the solutions of (1.1) and (1.2) has not been explored in the literature. Our first contribution, in Section 3, is to understand better the connection between the two.

A third formulation was proposed in [35] using the parametrization $\Theta = WH$, where $W \in \mathbb{R}^{m \times r}$ and $H \in \mathbb{R}^{r \times n}$, which is common practice for low-rank matrix approximation, as the rank of Θ does not exceed r [43, 38]. This formulation, referred to as 3B-ReLU-NMD model, is

$$\min_{Z,W,H} \|Z - WH\|_F^2 \quad \text{such that} \quad \max(0, Z) = X, \tag{1.3}$$

where 3B stands for 3 blocks of variables. The objective function in (1.3) is again jointly nonconvex, but it is convex with respect to each matrix variable, W, H and Z, and can be solved computationally faster, as the rank constraint is not explicitly involved. Problem (1.3) can be tackled using the block coordinate descent (BCD) method, optimizing each matrix variable sequentially [35, 41].

Outline and contribution. Section 2 motivates the study of ReLU-NMD by presenting several applications. In Section 3, we clarify the relationship between ReLU-NMD and Latent ReLU-NMD. In particular, we provide a matrix X for which the infimum of the rank-1 Latent-ReLU-NMD is not attained, showing the ill-poseness of Latent-ReLU-NMD (Lemma 3.1). On the other hand, for the same matrix X, ReLU-NMD has a non-empty set of optimal rank-1 solutions (Lemma 3.2). This shows that, in general, the two models do not share the same set of optimal low-rank solutions, that is, the sets

$$\{\Theta^* \mid \Theta^* \text{ optimal for ReLU-NMD (1.1)}\}, \quad \text{and} \quad \{\widehat{\Theta} \mid (\widehat{Z}, \widehat{\Theta}) \text{ optimal for Latent-ReLU-NMD (1.2)}\},$$

do not necessarily coincide. However, we show that any feasible point $(\widehat{Z}, \widehat{\Theta})$ of the Latent-ReLU-NMD model satisfies the relation

$$\|X - \max(0, \widehat{\Theta})\|_F \leq 2\|\widehat{Z} - \widehat{\Theta}\|_F.$$

This guarantees that any approximated solution of Latent-ReLU-NMD provides an approximated solution for ReLU-NMD whose residual is at most twice that of Latent-ReLU-NMD.

In Section 4, we first briefly review the state-of-the-art algorithms for solving Latent-ReLU-NMD and 3B-ReLU-NMD (Section 4.1). We then prove convergence of the BCD scheme for solving 3B-ReLU-NMD (BCD-NMD) (Section 4.2), demonstrating that it falls within the framework of BCD with strictly quasi-convex subproblems [16]. In addition, we introduce a novel extrapolated version of BCD-NMD (eBCD-NMD) (Section 4.3), adapting the well-known LMaFit algorithm for matrix completion [43]. We prove subsequence convergence of eBCD-NMD (Theorem 4.7). To the best of our knowledge, BCD-NMD and eBCD-NMD are the first algorithms for solving 3B-ReLU-NMD with convergence guarantees.

In Section 5, we provide extensive numerical experiments comparing our new approach, eBCD-NMD, with the state of the art. We test the algorithms on various applications: matrix completion with ReLU sampling, Euclidean distance matrix completion (EDMC), compression of sparse data, and low-dimensional embedding. In particular, our experiments show that eBCD-NMD consistently accelerates BCD-NMD and performs favorably with the state of the art, while having stronger theoretical guarantees.

2. Applications of ReLU-NMD. In this section, we first show the link between the ReLU decomposition and neural networks. Then we present some applications of the ReLU decomposition, illustrating how nonlinear models can be used in a variety of different contexts.

2.1. Neural network interpretation. The increasing interest in nonlinear models can be attributed to the rising popularity of neural networks, which use nonlinear functions as activations in their hidden layers. Furthermore, low-rank nonlinear models, involving matrix decompositions, have been studied in the context of neural networks [11, 23, 28, 36]. Let us discuss the link between ReLU-NMD and ReLU networks. Figure 2.1 illustrates a one hidden-layer neural network. Let $a_i \in \mathbb{R}^p$ be

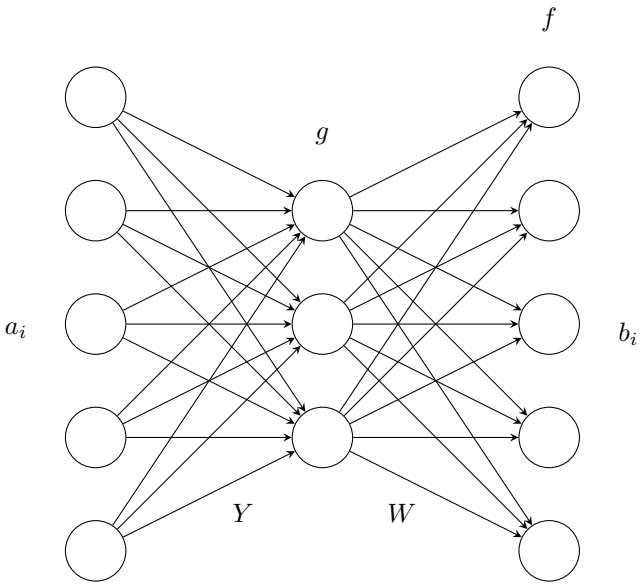

Figure 2.1: One hidden-layer neural network with 5 input and output nodes and 3 hidden nodes.

p-dimensional input and $b_i \in \mathbb{R}^m$ the corresponding output vectors, for $i = 1, \ldots, n$. Define the input and output matrices $A = [a_1, \ldots, a_n] \in \mathbb{R}^{p\times n}$ and $B = [b_1, \ldots, b_n] \in \mathbb{R}^{m\times n}$, respectively. We also denote $Y \in \mathbb{R}^{r\times p}$ the matrix of the weights between the input layer and hidden layer, and $W \in \mathbb{R}^{m\times r}$ the matrix of the weights between the hidden layer and the output layer. The functions g and f are the activation functions for the first and second layer, respectively. We are interested in finding an approximation of a given data X using this neural network with one hidden layer. If we choose the error measure to be the least squares error, we need to minimize

$$\|X - B\|_F^2 \;=\; \|X - f(Wg(YA))\|_F^2. \tag{2.1}$$

This shows that the hidden representation of each data point is a nonlinear combination of the inputs of the network, that is, $h_i = g(Ya_i) \in \mathbb{R}^r$, or $H = g(YA) \in \mathbb{R}^{r\times n}$ in matrix form. NMD is a relaxation of (2.1) in which the hidden representation of the input matrix, H, can be any real matrix of appropriate dimensions. Furthermore, the number of hidden neurons is equal to the rank of the decomposition, and it needs to be chosen a priori as when designing the structure of the network. If the number of hidden neurons is smaller than the dimension of the inputs, the hidden layer is a low-dimensional embedding of the data.

2.2. Compression of sparse data. An intuitive application of ReLU decomposition models is the compression of sparse data. Let $X \in \mathbb{R}^{m\times n}$ be a sparse and nonnegative matrix with $\mathrm{nnz}(X)$ non-zero entries, and let $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{r\times n}$ be such that $X \approx \max(0, WH)$. As long as $r(m + n) < \mathrm{nnz}(X)$, storing the factors W and H requires less memory than storing X. ReLU models can outperform the standard linear compression techniques such as the TSVD on sparse data, such as

sparse images or dictionaries [34, 33, 35, 41, 4]. An interesting example is the identity matrix of any dimension that can be exactly reconstructed using a rank-3 ReLU decomposition [11, 34].

2.3. Matrix completion with ReLU sampling. The class of matrix completion problems with entries missing not at random (MNR) is characterized by the fact that the probability of an entry being missing depends on the matrix itself. For example, when completing a survey, people are likely to share nonsensitive information such as the name or the surname, while they might be more hesitant to disclose their phone number. Therefore, we might expect the missing entries to be concentrated in some area of the matrix more than others. Even though MNR matrix completion is still a relatively unexplored area, dropping the assumption that missing entries are independent of the matrix is a challenging problem that attracts more and more people as the increasing number of related works testifies [6, 29, 13, 7, 26]. Among them, Liu et al. [26] investigate the matrix completion problem when only positive values are observed, matrix completion with ReLU sampling. This instance of matrix completion falls within the MNR problems since the missing entries depend on the sign of the target matrix. Assume that X is a nonnegative and sparse matrix and that there exists a rank-r matrix Θ such that $X = \max(0, \Theta)$. In this case, finding a ReLU decomposition is equivalent to recovering the negative values, observing only the positive ones; thus, it is equivalent to matrix completion with ReLU sampling. This implies that we can use the algorithms for computing a ReLU decomposition to address this matrix completion problem, as illustrated in [26].

Inspired by the connection with ReLU sampling matrix completion, we introduce an equivalent formulation of 3B-ReLU-NMD that we will use throughout the paper. Let $\Omega = \{(i, j) \mid X_{ij} > 0\}$ be the set of positive entries of X and let $P_\Omega(X)$ be equal to X in Ω and zero elsewhere. We rewrite 3B-ReLU-NMD in (1.3) as

$$\text{argmin}_{Z,W,H} \frac{1}{2}\|Z - WH\|_F^2 \quad \text{such that} \quad P_\Omega(Z) = P_\Omega(X), \quad P_{\Omega^C}(Z) \leq 0, \tag{2.2}$$

where Ω^C is the complement of Ω and a reformulation of the constraint $\max(0, Z) = X$ is introduced. Using the matrix factorization $\Theta = WH$ is a standard approach in matrix completion [5, 19, 20, 21, 38], see [30, 25] for detailed surveys on the topic. To our knowledge, the problem (2.2), without the constraint $P_{\Omega^C}(Z) \leq 0$, was first addressed in [43] in the context of matrix completion. The novelties in our model are the additional information that the missing entries have negative signs, which translates into the constraints $P_{\Omega^C}(Z) \leq 0$, and the explicit dependence of the known index set Ω from the target matrix X. Since the formulations (1.3) and (2.2) are equivalent, we will refer to both as 3B-ReLU-NMD.

2.4. Euclidean distance matrix completion (EDMC). We provide now one practical example of the MNR matrix completion problem that can be solved using the ReLU decomposition. It is the recovery of a low-rank matrix for which only the smallest or the largest entries are observed. Let Θ be a low-rank matrix, and assume that we observe all the entries in Θ smaller than a known threshold d. We want to recover the missing entries, which are the ones larger than d. We can model this problem using a rank-1 modified ReLU decomposition: $X = \max(0, dee^T - \Theta)$, where e denotes the vector of all ones of appropriate dimension. The matrix X is nonzero exactly in the entries where Θ is smaller than d. Of course, an analogous model can be derived when only the largest entries are observed. The 3B-ReLU-NMD formulation in (2.2) can be easily adapted to include the rank-1 modification, that is,

$$\text{argmin}_{Z,W,H} \frac{1}{2}\|dee^T - WH - Z\|_F^2 \quad \text{such that} \quad P_\Omega(Z) = P_\Omega(X), \quad P_{\Omega^C}(Z) \leq 0. \tag{2.3}$$

The same idea applies also to the Latent-ReLU-NMD and all the algorithms can be adjusted to take into account the additional rank-one term. This formulation is particularly interesting when the matrix Θ contains the squared distances between a collection of points $p_1, \ldots, p_n \in \mathbb{R}^d$. This problem is often referred to as Euclidean distance matrix completion (EDMC) [1, 12, 22, 39], and is particularly meaningful in sensor network localization problems where sensors can only communicate with nearby

sensors. Let $P = [p_1, \ldots, p_n] \in \mathbb{R}^{d\times n}$, then the corresponding Euclidean distance matrix is $\Theta_{i,j} = \|p_i - p_j\|_2^2$ for all i, j, so that $\Theta = e\text{diag}(P^TP)^T + \text{diag}(P^TP)e^T + 2P^TP$ and it has at most rank $r = d+2$ and thus the rank of the factorization in (2.3) is known in advance. We will show in Section 5 that this slightly modifed ReLU-NMD model allows us to solve EDMC in this context.

2.5. Manifold learning. Manifold learning aims at learning a norm-preserving and neighborhood-preserving mapping of high-dimensional inputs into a lower-dimensional space [24]. Given a collection of points $\{z_1, \ldots, z_m\}$ in $\mathbb{R}^N$, Saul [33] looks for a faithful representation $\{y_1, \ldots, y_m\}$ in $\mathbb{R}^r$, with $r < N$. Let $\tau \in (0, 1)$, Saul defines an embedding to be τ-faithful if

$$\max(0, \langle y_i, y_j\rangle - \tau\|y_i\|\|y_j\|) = \max(0, \langle z_i, z_j\rangle - \tau\|z_i\|\|z_j\|). \tag{2.4}$$

Equation (2.4) implies that the embedding must preserve norms (take $i = j$ to obtain $(1-\tau)\|y_i\|^2 = (1-\tau)\|z_i\|^2$) as well as the small angles between two points (z_i, z_j): when $\cos(z_i, z_j) > \tau$, we need $\cos(z_i, z_j) = \cos(y_i, y_j)$. Large angles have to remain large but do not need to be preserved, that is, $\cos(z_i, z_j) \leq \tau$ only requires $\cos(y_i, y_j) \leq \tau$. The condition (2.4) defines a similarity between points based on norm and angles, which differs substantially from the more common concept of pairwise distances. This makes it suitable, for example, in text analysis where the z_i's are sparse vectors of word counts.

This embedding strategy is dubbed threshold similarity matching (TSM). Saul proposes three main steps to compute such an embedding. First, for each couple of points (z_i, z_j), compute the right-hand side in (2.4) and construct the sparse, nonnegative similarity matrix

$$X_{ij} = \max(0, \langle z_i, z_j\rangle - \tau\|z_i\|\|z_j\|) \quad \text{for all } i, j.$$

Second, compute a ReLU-decomposition, $X \approx \max(0, \Theta)$, and the matrix $\max(0, \Theta)$ represents the similarities between points in the lower-dimensional space. Finally, the embedded points are recovered from the low-rank approximation Θ; see [33] for more details.

3. ReLU-NMD vs Latent-ReLU-NMD. In this section, we explore the link between ReLU-NMD in (1.1) and Latent-ReLU-NMD in (1.2). In particular, we aim to determine if the optimal low-rank approximations provided by the two models coincide. First, when X contains negative values, Latent-ReLU-NMD is infeasible, since there exists no latent variable Z such that $X = \max(0, Z)$. In contrast, ReLU-NMD still admits feasible solutions. Of course, the ReLU decomposition is more meaningful when X is nonnegative which is the case we focus on in the following.

In the exact case, when there exists a low-rank matrix Θ such that $X = \max(0, \Theta)$, ReLU-NMD and Latent-ReLU-NMD are equivalent: for any low-rank solution Θ^* of (1.1), there exists a latent matrix Z^*, such that (Z^*, Θ^*) is a solution of (1.2) and vice versa. Indeed, in the exact case there always exists a trivial latent solution $Z^* = \Theta^*$. We show in what follows that this is not true in general, when an exact decomposition does not exist.

We demonstrate first that there exists a specific matrix X such that the rank-1 Latent-ReLU-NMD is ill-posed since the infimum of (1.2) is not attained (see Lemma 3.1). Moreover, by showing that for the same matrix X, ReLU-NMD admits a non-empty set of optimal solutions (Lemma 3.2), we establish that the two formulations do not necessarily yield the same optimal approximations. Note that we could not establish the general well-poseness of ReLU-NMD, which is a direction of future research.

LEMMA 3.1. *Let $0 < \epsilon < 1/\sqrt{2}$ be a fixed parameter, $r = 1$, and*

$$X = \begin{pmatrix} 1 & 0 \\ \epsilon & 1 \end{pmatrix}. \tag{3.1}$$

Then the infimum of Latent-ReLU-NMD in (1.2) is not attained.

Proof. We need to show that there exists no solution of Latent-ReLU-NMD attaining the infimum of (1.2). Let us prove the result by contradiction, and assume there exists a solution to (1.2), denoted

$(\widehat{Z}, \widehat{\Theta})$. First note that, given $\widehat{Z}$, the optimal $\widehat{\Theta}$ must be a best rank-1 approximation of $\widehat{Z}$. Hence $\widehat{Z}$ is such that the distance with its best rank-1 approximation is minimized and such that it satisfies the constraint $\max(0, \widehat{Z}) = X$. Let us denote the i-th singular values of Z as $\sigma_i(Z)$, and $\widehat{Z}_1$ be a best rank-one approximation of $\widehat{Z}$. By the Eckart-Young theorem, $\|\widehat{Z} - \widehat{Z}_1\|_F^2 = \sigma_2(Z)^2$ since $\widehat{Z}$ is a 2-by-2 matrix. Hence $\widehat{Z}$ must belong to

$$\text{argmin}_Z \, \sigma_2(Z)^2 \quad \text{such that} \quad \max(0, Z) = X. \tag{3.2}$$

Equivalently, we want to find $b \leq 0$ such that the matrix $Z = \begin{pmatrix} 1 & b \\ \epsilon & 1 \end{pmatrix}$ has the lowest possible second singular value. Since Z is a 2×2 matrix, we can compute the eigenvalues of ZZ^T, solve $\det(ZZ^T - \sigma I) = 0$ and get an explicit expression for the second singular value of Z: for a general matrix $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$, the second singular value is given by

$$\sigma_2 = \sqrt{\frac{S_1 - S_2}{2}},$$

where

$$S_1 = a^2 + b^2 + c^2 + d^2, \quad S_2 = \sqrt{(a^2 + b^2 - c^2 - d^2)^2 + 4(ac + bd)^2}.$$

We rewrite (3.2) as

$$\text{argmin}_{b<0} \; \ell(b) := \frac{2 + b^2 + \epsilon^2 - \sqrt{(b^2 - \epsilon^2)^2 + 4(b + \epsilon)^2}}{2}. \tag{3.3}$$

One can verify that $\lim_{b\to-\infty} \ell(b) = \epsilon^2$, while $\ell(b) > \epsilon^2$ for every $\epsilon \leq 1/\sqrt{2}$ and $b < 0$. This means the infimum of $\ell(b)$ is not attained by any finite value of b, and hence $\widehat{Z}$ does not exist. ∎

We demonstrate in the following lemma that rank-1 ReLU-NMD admits an optimal solution for X as in (3.1).

LEMMA 3.2. *Let X as in (3.1) and $r = 1$, then for every $0 < \epsilon < \frac{1}{\sqrt{2}}$, the infimum of ReLU-NMD is attained by all the matrices of the form*

$$\Theta = \begin{pmatrix} 1 & -v \\ -\frac{1}{v} & 1 \end{pmatrix}, \quad v \in \mathbb{R}_+, \tag{3.4}$$

with optimal value equal to ϵ^2.

Proof. Let us denote the error function of ReLU-NMD as $F(\Theta) = \|X - \max(0, \Theta)\|_F^2$. Note that 2×2 rank-1 matrices have limited patterns that the signs of their entries must follow. For instance, if the matrix does not contain zeros, the number of positive and negative entries must be even, otherwise the rank cannot be equal to one. We proceed by analyzing three different cases.

Case 1: The rank-1 matrix Θ contains one zero column or row. In this case, the residual $F(\Theta)$ cannot be less than 1 since the nonlinear approximation $\max(0, \Theta)$ has at least one zero in the diagonal.

Case 2: The rank-1 matrix Θ has no zero rows or columns, and it contains two negative entries. If there is a negative entry on the diagonal, then the error is larger than one. Otherwise, the positive entries on the diagonal can be set to 1 to minimize the error, which is exactly the matrix in (3.4), with error ϵ^2.

Case 3: The rank-1 matrix Θ contains only positive entries. If $\Theta \geq 0$, ReLU-NMD corresponds to the following problem without nonlinear activations:

$$\text{argmin}_\Theta \|X - \Theta\|_F^2 \;\; \text{such that} \;\; \Theta \geq 0.$$

Therefore, we need to show that there exists no rank-1 matrix with nonnegative entries that achieves a residual value $F(\Theta)$ strictly smaller than ϵ^2. Let $\widehat{\sigma}_2$ be the second singular value of X as in (3.1). If rank(Θ)=1, then $\|X-\Theta\|_F^2 \geq \widehat{\sigma}_2^2$ and using equation (3.3),

$$\widehat{\sigma}_2^2 = \frac{2+\epsilon^2-\epsilon\sqrt{\epsilon^2+4}}{2}.$$

Simple calculations show that $\widehat{\sigma}_2^2$ is strictly larger than ϵ^2 for any $0<\epsilon<\frac{1}{\sqrt{2}}$. ∎

As a consequence of Lemma 3.1 and Lemma 3.2, we have the following corollary.

COROLLARY 3.3. *ReLU-NMD and Latent-ReLU-NMD may not share the same set of optimal low-rank solutions.*

We established the non equivalence between the low-rank solutions of ReLU-NMD and Latent-ReLU-NMD. However, in practice, we are interested in computing accurate solutions of ReLU-NMD even though they might not be optimal. Numerous numerical results show that the ReLU decomposition obtained by Latent-ReLU-NMD achieves a small residual error also for ReLU-NMD [34, 35, 41]. The following theorem formalizes this observation, showing that any feasible point $(\widehat{\Theta},\widehat{Z})$ of the Latent model has a residual value for $\widehat{\Theta}$ in the original ReLU-NMD that is upper bounded by twice the residual of Latent-ReLU-NMD.

THEOREM 3.4. *Let $X\in\mathbb{R}^{n\times m}$ be a nonnegative matrix and denote as $(\widehat{\Theta},\widehat{Z})$ a feasible point of the Latent-ReLU-NMD model in (1.2), then*

$$\|X-\max(0,\widehat{\Theta})\|_F^2 \leq 2\|\widehat{Z}-\widehat{\Theta}\|_F^2. \tag{3.5}$$

Proof. We have

$$\begin{aligned}\|X-\max(0,\widehat{\Theta})\|_F^2 &= \|X-\widehat{Z}+\widehat{Z}-\max(0,\widehat{\Theta})-\min(0,\widehat{\Theta})+\min(0,\widehat{\Theta})\|_F^2\\ &\leq \|X-\widehat{Z}+\min(0,\widehat{\Theta})\|_F^2+\|\widehat{Z}-\widehat{\Theta}\|_F^2.\end{aligned} \tag{3.6}$$

Let us define $\Omega=\{(i,j)\mid X(i,j)>0\}$ and $\widehat{\Omega}=\{(i,j)\mid \widehat{\Theta}(i,j)>0\}$, and Ω^C and $\widehat{\Omega}^C$ their respective complements. By assumption $\widehat{Z}$ is a feasible solution of (1.2) and hence $\max(0,\widehat{Z})=X$, therefore

$$\begin{aligned}\|\widehat{Z}-\widehat{\Theta}\|_F^2 = &\sum_{(i,j)\in\Omega\cap\widehat{\Omega}}(X_{ij}-\widehat{\Theta}_{ij})^2+\sum_{(i,j)\in\Omega\cap\widehat{\Omega}^C}(X_{ij}-\widehat{\Theta}_{ij})^2+\\ &+\sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}}(\widehat{Z}_{ij}-\widehat{\Theta}_{ij})^2+\sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}^C}(\widehat{Z}_{ij}-\widehat{\Theta}_{ij})^2.\end{aligned}$$

Similarly, observe that $Z_{i,j}\leq 0$ whenever $X_{i,j}=0$, and hence

$$\begin{aligned}&\|X-\widehat{Z}+\min(0,\widehat{\Theta})\|_F^2 =\\ &=\sum_{(i,j)\in\Omega\cap\widehat{\Omega}^C}\widehat{\Theta}_{ij}^2+\sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}}\widehat{Z}_{ij}^2+\sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}^C}(\widehat{Z}_{ij}-\widehat{\Theta}_{ij})^2\\ &\leq\sum_{(i,j)\in\Omega\cap\widehat{\Omega}^C}(X_{ij}-\widehat{\Theta}_{ij})^2+\sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}}(\widehat{Z}_{ij}-\widehat{\Theta}_{ij})^2+\sum_{(i,j)\in\Omega^C\cap\widehat{\Omega}^C}(\widehat{Z}_{ij}-\widehat{\Theta}_{ij})^2\\ &\leq\|\widehat{Z}-\widehat{\Theta}\|_F^2.\end{aligned} \tag{3.7}$$

The first inequality holds because

$$\begin{aligned}&\widehat{\Theta}_{ij}^2\leq(X_{ij}-\widehat{\Theta}_{ij})^2 \text{ for } (i,j)\in\Omega\cap\widehat{\Omega}^C, \text{ and}\\ &\widehat{Z}_{ij}^2\leq(\widehat{Z}_{ij}-\widehat{\Theta}_{ij})^2 \text{ for } (i,j)\in\Omega^C\cap\widehat{\Omega}.\end{aligned}$$

Combining (3.6) and (3.7), we get (3.5). ∎

A straightforward consequence of Theorem 3.4 is that a solution of Latent-ReLU-NMD with small error provides a ReLU-NMD solution with small error, even though the two models do not necessarily yield the same approximation.

4. Algorithms for the ReLU decomposition. This section is dedicated to the algorithms for computing ReLU decompositions and to the study of their convergence. In Section 4.1, we briefly review the state-of-the-art algorithms for solving Latent-ReLU-NMD and 3B-ReLU-NMD. In Section 4.2, we discuss the convergence of the BCD-NMD scheme. Finally, we present our new algorithm eBCD-NMD along with its convergence in Section 4.3.

4.1. Previous works. Let us start with Latent-ReLU-NMD, since we use the algorithms designed to solve Latent-ReLU-NMD as baselines in the numerical comparison in Section 5. To our knowledge, all existing algorithms for Latent-ReLU-NMD are alternate optimization methods in which each block of variables is updated sequentially [16, 17, 32, 31, 40] and lack convergence guarantees.

The first approach is the so-called Naive-ReLU-NMD, introduced by Saul in [34], which employs a basic alternating scheme over Z and Θ with closed-form solutions for both. Similar approaches have been studied in matrix completion, see for example [5, 8, 10, 15]. At each iteration, firstly Θ is fixed and then the optimal Z is computed solving the subproblem

$$\text{argmin}_Z \|Z - \Theta\|_F^2 \quad \text{such that} \quad \max(0, Z) = X. \tag{4.1}$$

The solution of (4.1) is given by the projection of Θ over the set $\{Z \ : \ \max(0, Z) = X\}$, that is,

$$Z_{ij} = \begin{cases} X_{ij} & \text{if } X_{ij} > 0 \\ \min(0, \Theta_{ij}) & \text{if } X_{ij} = 0. \end{cases}$$

Once Z is updated, Θ is updated solving

$$\text{argmin}_\Theta \|Z - \Theta\|_F^2 \quad \text{such that} \quad \text{rank}(\Theta) \leq r,$$

whose optimal solution(s) can be obtained with the TSVD of Z. Despite its simplicity, this scheme is highly reliable in practice, with the most costly operation at each iteration being the computation of a TSVD.

An accelerated, heuristic version of the Naive-NMD scheme was proposed in [35]. Specifically, the so-called Aggressive-ReLU-NMD (A-NMD) employs an extrapolation step for both Z and Θ while the extrapolation parameter changes adaptively at each iteration, depending on the value of the residual. A-NMD is an example where, despite its lack of theoretical guarantees, extrapolation can be considerably faster [2, 43].

Another algorithm for solving Latent-ReLU-NMD is the so-called expectation-minimization NMD (EM-NMD) [34]. This approach is based on a Gaussian latent variable model parametrized by the low-rank matrix Θ and a parameter σ. For each entry of the original matrix, a Gaussian latent variable $Z_{ij} = \mathcal{N}(\Theta_{ij}, \sigma^2)$ is sampled. The method estimates Θ and σ by maximizing the likelihood of the data with respect to Θ and σ. The optimization is carried out using the Expectation-Minimization (EM-NMD) scheme that consists of two main steps: the E-step which computes posterior mean and variance of the Gaussian latent variable model, and the M-step re-estimates the parameters of the model from the posterior mean and variance.

The BCD scheme for solving 3B-ReLU-NMD in (2.2), referred to as BCD-NMD, consists of an alternating optimization procedure over three blocks of variables (Z, W, H). BCD-NMD belongs to the class of exact BCD methods, where a global optimizer for each of the subproblem is computed sequentially. Let $A^\dagger$ denote the Moore-Penrose inverse of A and let (H^k, Z^k, W^k) be the current iteration.

The $(k+1)$-th step of BCD-NMD is as follows:

$$
\begin{aligned}
Z^{k+1} &= \operatorname{argmin}_Z \{\|Z - W^k H^k\|_F^2 \quad \text{s.t.} \quad \max(0, Z) = X\} = P_\Omega(X) + P_{\Omega^C}(\min(0, W^k H^k)), \\
W^{k+1} &= \operatorname{argmin}_W \|Z^{k+1} - W H^k\|_F^2 = Z^{k+1}(H^k)^\dagger, \\
H^{k+1} &= \operatorname{argmin}_H \|Z^{k+1} - W^{k+1} H\|_F^2 = ((W^{k+1})^T)^\dagger Z^{k+1}.
\end{aligned} \tag{4.2}
$$

The solution of the subproblems for $W \in \mathbb{R}^{m\times r}$ and $H \in \mathbb{R}^{r\times n}$ are analogous. In particular, the first is separable by rows of W, leading to m least squares problems in r variables with the data vector of dimension n, while the subproblem for H is separable by column and equivalent to n least squares problems of r variables and data vector of dimension m. Both these subproblems do not necessarily have a unique solution. In fact, uniqueness is guaranteed only if W^k and H^k remain full rank throughout the iterations. Alternatively, uniqueness is ensured by adding a Tikhonov regularization term or a proximal term in each of the subproblems [16, 17, 3]; however, it is not necessary to prove the convergence of the BCD-NMD algorithm.

On the contrary, the subproblem in the latent variable Z admits a unique global minimum, which is the projection of the product $W^k H^k$ onto the feasible, convex set

$$
\{Z \in \mathbb{R}^{m\times n} \ : \ P_\Omega(Z) = P_\Omega(X), \ P_{\Omega^C}(Z) \le 0\}. \tag{4.3}
$$

The overall BCD-NMD algorithm is reported in Algorithm 4.1 and has a computational cost of $O(r|\Omega^C| + (m+n)r^2)$.

Algorithm 4.1 BCD-NMD: block coordinate descent for ReLU-NMD

Input: X, W^0, H^0, maxit.
Output: low-rank factors W^k and H^k.

1: Set $\Omega = \{(i,j) \mid X_{ij} > 0\}$
2: **for** $k = 0, 1, \ldots,$ maxit **do**
3: $\quad Z^{k+1} = P_\Omega(X) + P_{\Omega^C}(\min(0, W^k H^k))$
4: $\quad W^{k+1} = Z^{k+1}(H^k)^\dagger$
5: $\quad H^{k+1} = ((W^{k+1})^T)^\dagger Z^{k+1}$
6: **end for**

Similarly to the Naive scheme for Latent-ReLU-NMD, adding extrapolation might be beneficial also for BCD-NMD. In particular, the work [35] proposes to perform an extrapolation step on the latent variable Z and on the low-rank product WH, resulting in the so-called 3B-NMD algorithm. Moreover, Wang et al. [41, 42] added a Tikhonov regularization for both the variables W and H in the 3B-ReLU-NMD formulation. Their algorithm is dubbed as ReLU-NMD-T and it is based on the 3B-NMD scheme but performs two more extrapolation steps following the computation of W and H. Even though numerical experience shows that both the accelerated algorithms outperform the original BCD-NMD, convergence still remains an open problem. For the numerical experiments in Section 5, we consider only 3B-NMD since we are interested in comparing algorithms that solve the non-regularized 3B-ReLU-NMD.

4.2. Convergence of BCD-NMD. In this section, we discuss the convergence to a stationary point of BCD-NMD (Algorithm 4.1) which, to the best of our knowledge, has not been studied before. Indeed, our goal is to prove that BCD-NMD is convergent under the assumption that a limit point of the sequence generated by the scheme exists. We proceed by showing that the method falls into the more general framework of exact block coordinate descent (BCD), where one of the subproblems is strictly quasi-convex.

Specifically, the general BCD scheme is studied by Grippo et al. [16] when the objective function is continuously differentiable over the Cartesian product of m closed, convex sets, one for each block of variables. We first observe that the 3B-ReLU-NMD objective function is continuously differentiable and defined over three blocks of variables Z, W, and H. The latent variable Z is the only constrained variable, and it is easy to show that the constraint set in (4.3) is convex. Therefore, the 3B-ReLU-NMD is a particular instance of the more general problem considered in [16].

THEOREM 4.1. *Any limit point of the sequence generated by BCD-NMD (Algorithm 4.1) is a stationary point of problem (2.2).*

Proof. The global convergence of the BCD scheme for a general function with m blocks of variables is analyzed under generalized convexity assumptions in [16] (Proposition 5). The key hypotheses are: 1) the existence of a limit point of the sequence generated by the BCD algorithm, and 2) strict quasi-convexity of at least $m-2$ subproblems. Therefore, since 3B-ReLU-NMD has three blocks of variables, we just need one subproblem to be strictly quasi-convex. The first subproblem in the variable Z is a convex, quadratic problem over a convex set, making it strictly quasi-convex, implying the convergence of BCD-NMD. ∎

The assumption of the existence of the limit point is an undesirable hypothesis. However, it is necessary because the level sets of the objective function are not bounded. Therefore, similarly to Latent-ReLU-NMD, 3B-ReLU-NMD might have instances in which the minimum is not attained (see Theorem 3.2). This means that, in pathological cases, the sequence generated by BCD-NMD may diverge simply because the infimum of the function is not attained by any finite value. Thus, the hypothesis regarding the existence of a limit point of the BCD-NMD sequence cannot be weakened. One possible way to overcome this limitation is by adding a regularization term to the objective function (2.2), which makes the level sets compact, as in [41]. However, this is not the purpose of this work which focuses on the non-regularized formulation.

4.3. Extrapolated block coordinate descent NMD (eBCD-NMD). We present now a new extrapolated version of Algorithm 4.1, namely extrapolated BCD-NMD (eBCD-NMD). Our goal is to accelerate BCD-NMD while preserving convergence to a stationary point under mild assumptions. The eBCD-NMD method is an adaptation of the well-known LMaFit algorithm [43] for matrix completion. The standard matrix completion formulation in [43] is similar to the one we consider in (2.2), although it does not include the inequality constraint $P_{\Omega^C}(Z) \leq 0$, which models the crucial information that the missing entries are negative. Consequently, we derive a new optimization scheme, differing from the one in [43], as the update of Z presents an additional nonlinear term; see equation (4.4). Moreover, we relax the hypothesis in [43] that require $\{W^k\}$ and $\{H^k\}$ to remain full-rank for all iterations. We present two different versions of the eBCD-NMD algorithm that, starting from the same point, produce the same low-rank approximation W^kH^k, yet the factors W^k and H^k vary. In what follows, we omit the iteration index, k, to simplify the notation, as we focus on a single iteration eBCD-NMD. We assume we have a point (Z, W, H) from which we perform one iteration.

4.3.1. First version of eBCD-NMD. Let $\alpha \geq 1$, then the first version of the eBCD-NMD scheme uses the following update: given the iterate (Z, W, H), it computes

$$
\begin{aligned}
Z_\alpha &= \alpha Z + (1-\alpha)WH, \\
\widehat{W}(\alpha) &= Z_\alpha H^\dagger, \\
\widehat{H}(\alpha) &= (\widehat{W}(\alpha)^T)^\dagger Z_\alpha, \\
\widehat{Z}(\alpha) &= P_\Omega(X) + P_{\Omega^C}(\min(0, \widehat{W}(\alpha)\widehat{H}(\alpha))),
\end{aligned}
\tag{4.4}
$$

where $\left(\widehat{Z}(\alpha), \widehat{W}(\alpha), \widehat{H}(\alpha)\right)$ provides the next iterate. Similarly to BCD-NMD, the main computational cost for the scheme in (4.4) is solving the two least squares problems for computing $\widehat{W}(\alpha)$ and $\widehat{H}(\alpha)$.

The optimization scheme in (4.4) differs from the one in (4.2) because of the new variable Z_α which is used in place of the original latent variable Z to update W and H. Moreover, if we assume the matrix H to be full-rank, we can derive an intuitive interpretation of the update of Z_α as an *indirect extrapolation step*. Since H is full-rank we have $H^\dagger = H^T(HH^T)^{-1}$ and thus

$$\widehat{W}(\alpha) = Z_\alpha H^T(HH^T)^{-1} = \alpha Z H^T(HH^T)^{-1} + (1-\alpha)WHH^T(HH^T)^{-1} = \alpha W_{\text{BCD}} + (1-\alpha)W,$$

where W_{BCD} is the update of the variable W that we would have had if we performed one step of the BCD-NMD scheme (4.2). Moreover, if we define $\beta = \alpha - 1$, then

$$\widehat{W}(\alpha) = \alpha W_{\text{BCD}} + (1-\alpha)W = (1+\beta)W_{\text{BCD}} - \beta W = W_{\text{BCD}} + \beta(W_{\text{BCD}} - W), \tag{4.5}$$

that is, an extrapolation step for the BCD-NMD update of W, provided that $\alpha \in (1,2)$, that means $\beta \in (0,1)$. Unfortunately, such an intuitive interpretation does not apply to the update of the variable H. Note that the step in (4.5) is properly referred to as an extrapolation step if β is in the interval (0,1) or equivalently α is in the interval (1,2), but our scheme potentially allows for larger values that exceed the upper bound.

4.3.2. Improved version of eBCD-NMD. The second version of the eBCD-NMD does not require solving any least squares problem and the main computational cost is associated with a single QR factorization of a matrix in $\mathbb{R}^{m\times r}$. We specify that this scheme can be adapted to design alternative versions of BCD-NMD and 3B-NMD as well. We state the following lemma from [43] that allows us to derive the second version of eBCD-NMD.

LEMMA 4.2. *Let $\widehat{W}(\alpha)$ be the update of W for the scheme in (4.4), then*

$$\mathcal{R}(\widehat{W}(\alpha)) = \mathcal{R}(Z_\alpha H^T),$$

where $\mathcal{R}(A)$ denotes the range of matrix A.

Proof. Let $UDV^T = H$ be the economy form SVD of H, then

$$\widehat{W}(\alpha) = Z_\alpha H^\dagger = Z_\alpha V D^\dagger U^T.$$

Moreover, since $Z_\alpha H^T = Z_\alpha V D U^T$, we get $\mathcal{R}(\widehat{W}(\alpha)) = \mathcal{R}(Z_\alpha H^T)$. ∎

Lemma 4.2 shows that the range space of $\widehat{W}(\alpha)$ can be obtained from that of $Z_\alpha H^T$. Let $Q(\alpha)$ be a matrix whose columns contain an orthogonal basis for $\mathcal{R}(Z_\alpha H^T)$. Then, using Lemma 4.2 and from the second and third lines in (4.4), we get

$$\widehat{W}(\alpha)\widehat{H}(\alpha) = \widehat{W}(\alpha)\widehat{W}(\alpha)^\dagger Z_\alpha = Q(\alpha)Q(\alpha)^T Z_\alpha. \tag{4.6}$$

This shows that the low-rank approximation produced by the eBCD-NMD is nothing else than the orthogonal projection of Z_α into $\mathcal{R}(Z_\alpha H^T)$. Justified by the relation in (4.6), we now state the improved version of eBCD-NMD. Given an iterate (Z, W, H), we compute

$$\begin{aligned}
&Z_\alpha = \alpha Z + (1-\alpha)WH,\\
&Q(\alpha) \to \text{ Orthogonal basis of } \mathcal{R}(Z_\alpha H^T),\\
&W(\alpha) = Q(\alpha),\\
&H(\alpha) = W(\alpha)^T Z_\alpha = Q(\alpha)^T Z_\alpha,\\
&Z(\alpha) = P_\Omega(X) + P_{\Omega^C}(\min(0, W(\alpha)H(\alpha))),
\end{aligned} \tag{4.7}$$

to obtain the next iterate $(Z(\alpha), W(\alpha), H(\alpha))$. Equation (4.6) guarantees that the low-rank approximations computed by the scheme in (4.4) and in (4.7) are equal, that is $\widehat{W}(\alpha)\widehat{H}(\alpha) = W(\alpha)H(\alpha)$. In practice, $Q(\alpha)$ can be computed from the economy QR decomposition of $Z_\alpha H^T$. Note that if $\text{rank}(Z_\alpha H^T) = r_1 < r$, the first r_1 columns of the Q matrix from the economy QR decomposition of $Z_\alpha H^T$ are not, in general, an orthogonal basis of the range space of the matrix. To ensure this property, a modified version of the QR decomposition using column pivoting must be used instead. Specifically, there exists $Q(\alpha) \in \mathbb{R}^{m\times r_1}$, containing an orthogonal basis of $Z_\alpha H^T$, such that $Z_\alpha H^T \Pi = Q(\alpha)R(\alpha)$, where Π is a permutation matrix and $R(\alpha) \in \mathbb{R}^{r_1\times r}$ [14]. Therefore, we potentially allow for a drop in the rank of the approximation whenever $Z_\alpha H^T$ is rank-deficient. However, we have never encountered this case in practical situations because $Z_\alpha H^T$ is likely to have rank at least r while the factors $Q(\alpha)$ and $H(\alpha)$ have rank exactly r.

We proceed by defining the residual of our algorithm, denoted as $S(\alpha)$,

$$S(\alpha) = Z(\alpha) - W(\alpha)H(\alpha) = P_\Omega(X - W(\alpha)H(\alpha)) - P_{\Omega^C}(\max(0, W(\alpha)H(\alpha))). \tag{4.8}$$

We highlight an interesting relation between the variable Z_α and the residual. Let S denote the residual matrix of the iterate (Z, W, H), defined as in (4.8) with Z, W, and H instead of $Z(\alpha)$, $W(\alpha)$ and $H(\alpha)$ respectively, then

$$\begin{aligned} Z_\alpha &= \alpha Z + (1-\alpha)WH = \alpha(P_\Omega(X) + P_{\Omega^C}(\min(0, WH)) - WH) + WH \\ &= \alpha(P_\Omega(X - WH) - P_{\Omega^C}(\max(0, WH))) + WH \\ &= WH + \alpha S. \end{aligned} \tag{4.9}$$

Equation (4.9) shows that Z_α is indeed the low-rank approximation WH at step k, to which we add a multiple of the residual S. This suggests a second interpretation of the update of Z_α as a residual corrected step of the low-rank approximation WH.

Restarting scheme of the extrapolation parameter. We need now to determine a rule to select the extrapolation parameter α at each iteration. Extrapolating a variable means adding information from the past iterate, thus the extrapolation parameter α affects how much relevant is the past information in the current update. Of course, not all the values of α are acceptable to progress in minimizing the residual, and a good choice of the parameter is crucial. We present now the updating scheme of the extrapolation parameter α, inspired by the one introduced in [43]. The main idea is to choose the parameter α such that the residual of the function is decreased sufficiently between two consecutive iterations.

At each step, we start from a candidate value for α, and evaluate the ratio $\delta(\alpha)$ between the norms of the residual $S(\alpha)$ at iteration $k+1$ and S at iteration k, defined as

$$\delta(\alpha) = \frac{\|S(\alpha)\|_F}{\|S\|_F}. \tag{4.10}$$

If $\delta(\alpha) < 1$, we declare our update successful and the new iterate is set to $(W^{k+1}, H^{k+1}, Z^{k+1}) = (W(\alpha), H(\alpha), Z(\alpha))$. Furthermore, let $\bar{\delta}$ be a fixed threshold, if $\delta(\alpha) > \bar{\delta} \in (0, 1)$, meaning we do not consider the decrease of the residual fast enough, we increase α by setting $\alpha \leftarrow \min(\alpha + \mu, \bar{\alpha})$, where $\mu = \max(\mu, 0.25(\alpha - 1))$. If α reaches the upper bound $\bar{\alpha}$, we restart the parameter by setting $\alpha = 1$. This correction does not appear in [43], but we added it to guarantee the convergence of the our algorithm. On the contrary, if $\delta(\alpha) \geq 1$, the step is declared unsuccessful, we do not accept the update, and we compute a standard BDC-NMD update instead by setting $\alpha = 1$. Consequently, either we find an $\alpha \in (1, \bar{\alpha})$ that allows a decrease of the residual or we recover one iteration of the BCD-NMD, which is guaranteed to decrease the residual. Therefore, the new iterate of the algorithm produces a lower value in the residual with respect to the previous iteration.

The overall eBCD-NMD scheme is described in Algorithm 4.2, where we consider the updating scheme in (4.7). We choose the improved version of eBCD-NMD because (a) it exhibits faster performance in practice; (b) computing one QR factorization is in general more stable than solving two consecutive least squares problems; (c) one of the factors in the decomposition is orthogonal and this helps demonstrate crucial properties in the convergence analysis, such as boundness of the sequence generated by the eBCD-NMD algorithm. Let us also mention that, in [43], convergence is proved for a scheme similar to the one in (4.4), instead of the scheme in (4.7) as done in this paper.

Algorithm 4.2 eBCD-NMD

Input: X, Z^0,W^0, H^0, $\bar{\alpha}$ (default = 4), μ (default = 0.3), $\bar{\delta}$ (default = 0.8), maxit

Set $\Omega = \{(i,j) \mid X_{ij} > 0\}$ and $\alpha_0 = 1$.
for $k = 0, \ldots,$ maxit **do**
  $Z_{\alpha_k} = \alpha_k Z^k + (1-\alpha_k) W^k H^k$
  Compute $Q(\alpha_k)$ orthogonal basis of $\mathcal{R}(Z_{\alpha_k}(H^k)^T)$
  $W(\alpha_k) = Q(\alpha_k)$
  $H(\alpha_k) = Q(\alpha_k)^T Z_{\alpha_k}$
  $Z(\alpha_k) = P_\Omega(X) + P_{\Omega^C}(\min(0, W(\alpha_k)H(\alpha_k)))$
  if $\delta(\alpha_k) \geq 1$ **then**
    Set α_k to 1
    $(W^{k+1}, H^{k+1}, Z^{k+1}) = (W^k, H^k, Z^k)$
  else
    $(W^{k+1}, H^{k+1}, Z^{k+1}) = (W(\alpha_k), H(\alpha_k), Z(\alpha_k))$
    if $\delta(\alpha_k) \geq \bar{\delta}$ **then**
      $\mu = \max(\mu, 0.25(\alpha_k - 1))$ and $\alpha_{k+1} = \min(\alpha_k + \mu, \bar{\alpha})$
      if $\alpha_{k+1} = \bar{\alpha}$ **then**
        $\alpha_{k+1} = 1$,
      end if
    else
      $\alpha_{k+1} = \alpha_k$.
    end if
  end if
end for

4.3.3. Convergence analysis. We present now the convergence analysis of Algorithm 4.2. The proofs of the lemmas can be found in the Appendix. The main steps of the convergence analysis are summarized as follows.

Step 1 We first prove that there exists a range of values larger than 1 for α such that the scheme (4.7) decreases the residual, that is,

$$\|S\|_F^2 - \|S(\alpha)\|_F^2 > 0. \tag{4.11}$$

Specifically, we split the residual reduction in (4.11) into the contribution given by the update of each variable separately. In particular, Lemma 4.3 characterizes the residual reduction after the update of W and H, showing that it is nonnegative. Lemma 4.4 and Lemma 4.5 describe the residual reduction after updating the entries of Z.

Step 2 We provide the KKT conditions for problem (2.2); see (4.18).

Step 3 We use the residual reduction that we characterized in Step 1 to prove subsequence convergence of the eBCD-NMD algorithm in Theorem 4.7, showing that a subsequence of the eBCD-NMD iterates satisfies the KKT conditions at the limit. This is done analyzing all possible situations that the restarting scheme arises.

Step 1. Let us denote $\sigma_{WH}(\alpha)$ the variation of the residual after the update of W and H and $\sigma_Z(\alpha)$ the variations after updating the entries of Z, that is,

$$\|S\|_F^2 - \|S(\alpha)\|_F^2 = \underbrace{\|S\|_F^2 - \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - Z_\alpha\|_F^2}_{=:\sigma_{WH}(\alpha)} + \underbrace{\frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - Z_\alpha\|_F^2 - \|S(\alpha)\|_F^2}_{=:\sigma_Z(\alpha)}$$

Let U be an orthogonal basis of $\mathcal{R}(H^T)$. If H is full rank, we can define the orthogonal projections onto $\mathcal{R}(W(\alpha))$ and $\mathcal{R}(H^T)$ as:

$$\begin{aligned} E(\alpha) &= W(\alpha)W(\alpha)^T = Q(\alpha)Q(\alpha)^T, \\ P &= UU^T = H^T(HH^T)^{-1}H. \end{aligned} \tag{4.12}$$

The following three lemmas characterize $\sigma_{WH}(\alpha)$ and $\sigma_Z(\alpha)$ (the proofs can be found in the appendix).

LEMMA 4.3. *Given (Z, W, H) where H is full rank, let $(W(\alpha), H(\alpha))$ be computed by the scheme (4.7), then for any $\alpha \geq 1$,*

$$\sigma_{WH}(\alpha) = \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - WH\|_F^2 = \|SP\|_F^2 + \|E(\alpha)S(I-P)\|_F^2 \geq 0, \tag{4.13}$$

where $E(\alpha)$ and P are defined in (4.12).

Therefore, after the first two steps, the residual reduction is strictly positive unless $W(\alpha)H(\alpha) = WH$. We look at the residual reduction after updating the entries of Z in Ω^C (Lemma 4.4) and in Ω (Lemma 4.5). In the following lemmas, we assume that there exists an interval $[1, \widehat{\alpha}]$, such that $Z_\alpha H^T$ and ZH^T have the same rank. This hypothesis is needed to ensure the continuity of the residual reduction $\sigma_Z(\alpha)$ in a range of $\alpha > 1$; more details are given in the Appendix.

LEMMA 4.4. *Let $(W(\alpha), H(\alpha))$ be generated by the scheme in (4.7). Furthermore, assume there exists $\widehat{\alpha} > 1$ such that $\operatorname{rank}(Z_\alpha H^T) = \operatorname{rank}(ZH^T)$ for $\alpha \in [1, \widehat{\alpha}]$, then there exists $\alpha_* > 1$ such that for every $\alpha \in (1, \min(\alpha_*, \widehat{\alpha})]$,*

$$\frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_{\Omega^C}(S(\alpha))\|_F^2 \geq 0. \tag{4.14}$$

Moreover, if $P_{\Omega^C}(\min(0, W(1)H(1))) \neq P_{\Omega^C}((\min(0, WH))$, a strict inequality holds.

A similar result is proved in [43] for matrix completion. In their less restrictive setting, the residual reduction after updating the entries of Z in Ω^C is independent of α and nonnegative. However, due to the additional constraint $P_{\Omega^C}(Z) \leq 0$ in the 3B-ReLU-NMD formulation, the residual reduction in (4.14) of the eBCD-NMD algorithm is only locally nonnegative in a range where α is larger than one.

Analogously to Lemma 4.4, we now state a similar result that characterizes the residual reduction when the entries of Z in Ω are updated.

LEMMA 4.5. *Let $(W(\alpha), H(\alpha))$ be generated by the scheme in (4.7). Furthermore, assume there exists $\widehat{\alpha} > 1$ such that rank($Z_\alpha H^T$)=rank(ZH^T), for $\alpha \in [1, \widehat{\alpha}]$. Then*

$$\lim_{\alpha \to 1^+} \frac{1}{\alpha^2}\|P_\Omega(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_\Omega(S(\alpha))\|_F^2 = 0. \tag{4.15}$$

The following corollary gathers the findings from Lemma 4.3, Lemma 4.4, and Lemma 4.5.

COROLLARY 4.6. *Let $(W(\alpha), H(\alpha), Z(\alpha))$ be generated by the scheme in (4.7). Furthermore, assume there exists $\widehat{\alpha} > 1$ such that rank($Z_\alpha H^T$)=rank(ZH^T) for $\alpha \in [1, \widehat{\alpha}]$. If $P_{\Omega^C}(\min(0, WH)) \neq P_{\Omega^C}(\min(0, W(1)H(1)))$, then there exists $\tilde{\alpha} > 1$ such that*

$$\|S\|_F^2 - \|S(\alpha)\|_F^2 = \sigma_{WH}(\alpha) + \sigma_Z(\alpha) > 0 \quad \forall \alpha \in [1, \min(\widehat{\alpha}, \tilde{\alpha})]. \tag{4.16}$$

Corollary 4.6 guarantees that there exist some values of the extrapolation parameter α strictly larger than 1 resulting in a positive residual reduction for eBCD-NMD (4.7).

Step 2. Before stating the actual convergence theorem, we need to introduce the optimality conditions or KKT conditions for problem (2.2). Let us define the Lagrangian function as

$$\mathcal{L}(W,H,Z,\Lambda,\Sigma) = \frac{1}{2}\|Z-WH\|_F^2 + \langle \Lambda, P_\Omega(Z-X)\rangle + \langle \Sigma, P_{\Omega^C}(Z)\rangle, \tag{4.17}$$

where the dual variables are such that $\Lambda = P_\Omega(\Lambda)$ and $\Sigma = P_{\Omega^C}(\Sigma)$:

$$\begin{aligned}
\text{Stationarity condition: } & \nabla_W \mathcal{L}(Z,W,H) = -(Z-WH)H^T = 0,\\
& \nabla_H \mathcal{L}(Z,W,H) = -W^T(Z-WH) = 0,\\
& \nabla_Z \mathcal{L}(Z,W,H,\Lambda,\Sigma) = Z - WH + \Lambda + \Sigma = 0,\\
\text{Primal feasibility: } & P_\Omega(Z-X) = 0,\\
& P_{\Omega^C}(Z) \le 0,\\
\text{Dual feasibility: } & \Sigma \ge 0,\\
\text{Complementary slackness: } & \langle \Sigma, P_{\Omega^C}(Z)\rangle = 0,
\end{aligned} \tag{4.18}$$

where the multiplier matrices Λ and Ω measure the residual in Ω and Ω^C respectively, that is, $\Lambda = P_\Omega(WH - Z)$ and $\Sigma = P_{\Omega^C}(\max(0, WH))$. Our goal is to prove that, under some mild assumptions, a subsequence of the sequence generated by eBCD-NMD satisfies the KKT condition at the limit. Specifically, one can verify that the eBCD-NMD sequence satisfies the complementary slackness condition and the stationarity condition for Z at any point. Therefore, we just need to show that the optimality conditions for W and H are satisfied by a subsequence of the eBCD-NMD scheme at the limit, that is,

$$\lim_{k\to\infty} \nabla_W \mathcal{L}(Z^k, W^k, H^k) = 0, \qquad \lim_{k\to\infty} \nabla_H \mathcal{L}(Z^k, W^k, H^k) = 0. \tag{4.19}$$

Step 3. We are now ready to prove the convergence result for Algorithm 4.2.

THEOREM 4.7. *Let $(W^{k+1}, H^{k+1}, Z^{k+1})$ be generated by eBCD-NMD (Algorithm 4.2), and assume that $\{P_{\Omega^C}(W^kH^k)\}$ is bounded for all k. Then there exists a bounded subsequence of $(W^{k+1}, H^{k+1}, Z^{k+1})$ whose limit point satisfies the KKT conditions (4.18).*

Proof. We first discuss the boundness of the two sequences $\{W^{k+1}\}$ and $\{H^{k+1}\}$ produced by Algorithm 4.2. Te boundness of $\{P_{\Omega^C}(W^kH^k)\}$ implies that $\{Z^k\}$ and $\{W^kH^k\}$ are bounded. This also means that $\{Z_{\alpha_k}\}$ is a bounded sequence by construction. Moreover, the sequence $\{W^{k+1}\}$ contains orthogonal matrices whose columns are orthogonal bases for $\mathcal{R}(Z_{\alpha_k}(H^k)^T)$, thus it is bounded. Finally, $H^{k+1} = (W^{k+1})^T Z_{\alpha_k}$ and since $\{W^{k+1}\}$ and $\{Z_{\alpha_k}\}$ are bounded, then also $\{H^{k+1}\}$ is bounded.

We observe that at every iteration H^{k+1} has always the same rank as W^{k+1}. This means that H^{k+1} is full rank for every k and Lemma 4.3 can be applied. In fact, it follows from Lemma 4.2 that $\mathcal{R}(Q^{k+1}) = \mathcal{R}(Z_{\alpha_k}(H^k)^T) \subseteq \mathcal{R}(Z_{\alpha_k})$; that implies

$$\mathcal{R}(I) = \mathcal{R}((Q^{k+1})^T Q^{k+1}) \subseteq \mathcal{R}((Q^{k+1})^T Z_{\alpha_k}) = \mathcal{R}(H^{k+1}),$$

where I is the identity matrix of dimension equal to the number of columns of Q^{k+1} or equivalently to the rank of $Z_{\alpha_k}(H^k)^T$.

Next, we define the set of the iteration indices where the update of Z^k results in a positive residual reduction. Let $\mathcal{F} = \{k \ : \ \sigma_Z(\alpha_k) \ge 0\}$ and $\mathcal{F}^C$ its complement. Therefore, denoting $E^k = E(\alpha_k)$, since H^{k+1} is full rank, we can use Lemma 4.3 and summing over all $k \in \mathcal{F}$, we get

$$\|S^0\|_F^2 \ge \sum_{k\in\mathcal{F}} \sigma_{WH}(\alpha_k) = \sum_{k\in\mathcal{F}} \|S^kP^k\|_F^2 + \|E^kS^k(I-P^k)\|_F^2. \tag{4.20}$$

We analyze separately two different scenarios.

Case 1: assume $|\mathcal{F}^C| < \infty$. It follows from (4.20) that

$$\lim_{k\to\infty, k\in\mathcal{F}} \|S^k P^k\|_F^2 = 0, \quad \lim_{k\to\infty, k\in\mathcal{F}} \|E^k S^k (I - P^k)\|_F^2 = 0, \tag{4.21}$$

which implies

$$\lim_{k\to\infty, k\in\mathcal{F}} \|E^k S^k\|_F^2 = 0.$$

Let us recall that $P_\Omega(X) = P_\Omega(Z^k)$, $P_{\Omega^C}(Z^k) = P_{\Omega^C}(\min(0, W^k H^k))$ and using (4.12), we have

$$S^k P^k = (Z^k - W^k H^k) P^k = (Z^k - W^k H^k) U^k (U^k)^T.$$

Since U^k is an orthonormal basis for $\mathcal{R}((H^k)^T)$ and $\{H^k\}$ bounded,

$$\lim_{k\to\infty, k\in\mathcal{F}} (Z^k - W^k H^k)(H^k)^T = -\nabla_W \mathcal{L}(Z^k, W^k, H^k) = 0.$$

Moreover,

$$E^k S^k = E^k S^{k+1} + E^k (S^k - S^{k+1}) = W^{k+1}(W^{k+1})^T (Z^{k+1} - W^{k+1} H^{k+1}) + E^k (S^k - S^{k+1}). \tag{4.22}$$

Furthermore

$$\begin{aligned} \|S^k - S^{k+1}\|_F^2 =& \|P_\Omega(W^{k+1}H^{k+1} - W^k H^k)\|_F^2 + \|P_{\Omega^C}(\max(0, W^{k+1}H^{k+1}) - \max(0, W^k H^k))\|_F^2 \\ \leq& \|W^{k+1}H^{k+1} - W^k H^k\|_F^2 + \|\max(0, W^{k+1}H^{k+1}) - \max(0, W^k H^k)\|_F^2 \\ \leq& 2\|W^{k+1}H^{k+1} - W^k H^k\|_F^2. \end{aligned} \tag{4.23}$$

Using equation (4.13) and the fact that $\alpha_k \leq \bar{\alpha}$ by construction, we have

$$\|W^{k+1}H^{k+1} - W^k H^k\|_F^2 \leq (\bar{\alpha})^2 (\|S^k P^k\|_F^2 + \|E^k S^k (I - P^k)\|_F^2). \tag{4.24}$$

Combining (4.24) and (4.21) we get

$$\lim_{k\to\infty, k\in\mathcal{F}} \|W^{k+1}H^{k+1} - W^k H^k\|_F^2 = 0,$$

thus, from (4.23),

$$\lim_{k\to\infty, k\in\mathcal{F}} \|S^k - S^{k+1}\|_F^2 = 0. \tag{4.25}$$

Therefore, from (4.25) and (4.22), along with the boundness of $\{W^k\}$, we conclude that

$$\lim_{k\to\infty, k\in\mathcal{F}} (W^{k+1})^T (Z^{k+1} - W^{k+1} H^{k+1}) = \lim_{k\to\infty, k\in\mathcal{F}} -\nabla_H \mathcal{L}(Z^{k+1}, W^{k+1}, H^{k+1}) = 0.$$

Case 2: $|\mathcal{F}^C| = \infty$. Let us analyze two subcases.
Case 2a: $|\{k \in \mathcal{F}^C \mid \delta(\alpha_k) > \bar{\delta}\}| < \infty$. For k sufficiently large, we have

$$\|S^{k+1}\|_F^2 \leq \bar{\delta} \|S^k\|_F^2.$$

Therefore, it holds $\lim_{k\to\infty, k\in\mathcal{F}^C} \|S^k\|_F^2 = 0$, which means that the sequence converges to a global minimizer of the problem.

Case 2b: $|\{k \in \mathcal{F}^C \mid \delta(\alpha_k) > \bar{\delta}\}| = \infty$. The eBCD-NMD set the acceleration parameter α_k to 1 for an infinite number of iterations. Denote $\mathcal{J}_1$ the set of indices in which $\alpha_k = 1$, a similar relation to (4.20) holds:

$$\|S^0\|_F^2 \geq \sum_{k \in \mathcal{J}_1} \sigma_{WH}(\alpha_k) = \sum_{k \in \mathcal{J}_1} \|S^k P^k\|_F^2 + \|E^k S^k (I - P^k)\|_F^2.$$

Therefore, one can replicate the proof for Case 1 and the subsequence whose indexes are in $\mathcal{J}_1$ satisfies the optimality conditions at the limit. ∎

We emphasize that, similarly to BCD-NMD, we need a strong assumption, namely the boundness of $\{P_{\Omega^C}(W^k H^k)\}$ which guarantees that the eBCD-NMD sequence remains bounded. This hypothesis serves the same purpose as the existence of the limit point in the BCD-NMD algorithm, since it allows us to exclude pathological cases where the infimum of the problem is not attained.

5. Numerical experiments. In this section, we present numerical experiments on synthetic and real-world data sets, for different applications of ReLU decomposition: matrix completion with ReLU sampling, the recovery of Euclidean distance matrices, low-dimensional embedding on text data, and the compression sparse dictionaries. Our main purpose is to show that eBCD-NMD behaves favorably compared to the state of the art, namely Naive-NMD [34], EM-NMD [34], A-NMD [35], and 3B-NMD [35], while having convergence guarantees.

In all experiments, we use random initialization: entries of W and H are sampled from a Gaussian distribution, `W = randn(m,r)` and `H = randn(r,n)` in MATLAB. We then scale them so that the initial point matches the magnitude of the original matrix, that is, we multiply W by $\sqrt{\|X\|_F}/\|W\|_F$ and H by $\sqrt{\|X\|_F}/\|H\|_F$. If not otherwise specified, we stop each algorithm when we reach a small relative residual, namely $\Gamma_k = \|Z^k - W^k H^k\|_F / \|X\|_F \leq 10^{-9}$, or a fixed time limit is reached. The time limit will be selected depending on the size of the data. For the algorithms solving Latent-ReLU-NMD, the low-rank product $W^k H^k$ in Γ_k is substituted with the matrix Θ^k. The first stopping criteria can be achieved only if the target matrix admits an (almost) exact ReLU decomposition. The algorithms are implemented in MATLAB R2021b on a 64-bit Samsung/Galaxy with 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz and 8 GB of RAM, under Windows 11 version 23H2. The code is available online from https://github.com/giovanniseraghiti/ReLU-NMD .

The parameters of eBCD-NMD are the ones specified in Algorithm 4.2. For the parameters of the other methods, we use the setting from [35].

5.1. Matrix completion with ReLU sampling. In this subsection, we consider the same experiment used in [26] for symmetric matrix completion with ReLU sampling, adapted to the non-symmetric case. Specifically, we randomly generate the entries W and H from a Gaussian distribution, as for our initializations, and we construct $\Theta^t = WH$, and set $X = \max(0, \Theta^t)$ in the noiseless case and $\max(0, \Theta^t + N)$ in the noisy case, where $N = \sigma * \tilde{N} * \|\Theta^t\|_F / \|\tilde{N}\|_F$, $\tilde{N} =$ `randn(m,n)`, and $\sigma \in (0, 1)$. In our experiment, we set $m = n = 1000$, $r = 20$ and $\sigma = 10^{-2}$. Note that since W, H and N are Gaussian, X is approximately 50% sparse. We present the loglog plot of the average residual Γ_k against the average CPU time over 20 different randomly generated matrices X in Figure 5.1. The algorithms are stopped when the objective functions in (1.2) and (1.3) achieve a relative residual Γ_k below 10^{-9} in the noiseless case, or below $\sigma = 10^{-2}$ in the noisy case. We also report the average CPU time to reach these values in Table 5.1. Figure 5.1 shows that all the algorithms reach the tolerance on the relative residual. Moreover, eBCD-NMD outperforms all the other methods, being considerably faster both in the noiseless and noisy cases, as Table 5.1 confirms. In fact, compared to the second best, eBCD-NMD is on average more than twice faster in the noiseless case, and more than three time faster in the noisy case (each time, the second best is 3B-NMD).

5.2. EDMC. We consider now some test problems for EDMC. Specifically, we generate 200 points uniformly at random in $[0, 10]^3$ (upper left picture in Figure 5.2), or randomly clustered in six clusters

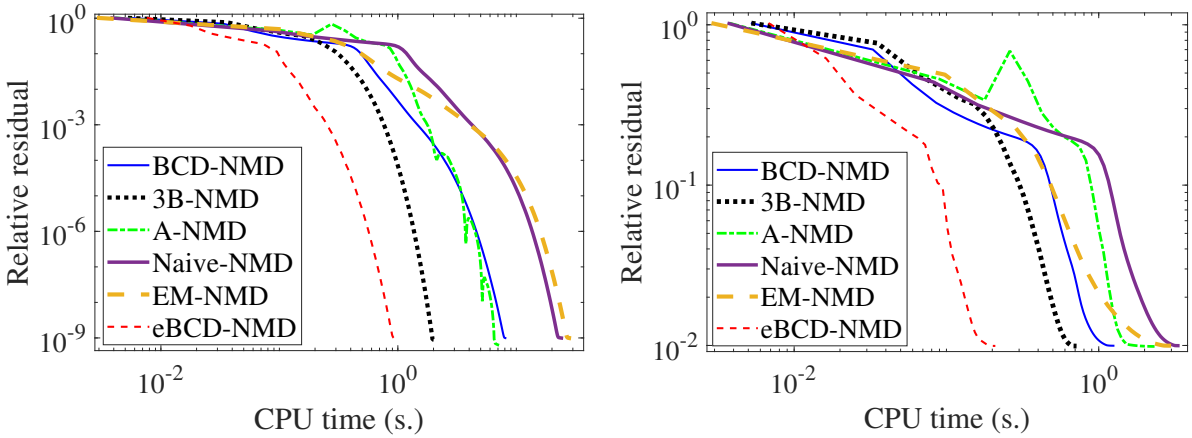

Figure 5.1: Matrix completion with ReLU sampling: evolution of the average relative residual Γ_k w.r.t. CPU time for the noiseless case (left image) and the noisy case (right image).

	BCD-NMD	eBCD-NMD	3B-NMD	Naive-NMD	A-NMD	EM-NMD
Noiseless	7.92 ± 0.16	**0.90 ± 0.04**	1.95 ± 0.04	22.0 ± 0.78	6.37 ± 0.33	26.7 ± 0.70
Noisy	1.14 ± 0.05	**0.19 ± 0.01**	0.66 ± 0.02	3.17 ± 0.12	1.54 ± 0.19	2.70 ± 0.15

Table 5.1: Average CPU time in seconds and standard deviation to reach a tolerance on the relative residual Γ_k of 10^{-9} in the noiseless case, and of 10^{-2} in the noisy case. Best values are highlighted in bold.

(bottom, left picture in Figure 5.2) in a three-dimensional space. In this example, we chose the centroid $(\bar{x}, \bar{y}, \bar{z})$ of the i-th cluster in $[-10, 10]^3$ uniformly at random, and we generate the points in MATLAB using $(\bar{x}, \bar{y}, \bar{z}) + \text{std} * \texttt{randn(1,3)}$, where std denotes the standard deviation that we fixed to std=3. We compute the Euclidean distance matrix Θ^t containing the squared distance between all couples of points and we construct our target matrix $X = \max(0, dee^T - \Theta^t)$, where d represents the largest value in the matrix that we want to observe. The upper bound on the rank is fixed to $r = 5$ which is equal to the rank of Θ^t. In this experiment, we assume the threshold d to be known, but one might be interested in considering d as an unknown of the model. We aim to recover all the entries of Θ^t larger than d. In practice, the threshold d is adapted to achieve the percentage of observed entries in the x-axis of the pictures on the right of Figure 5.2. Specifically, we display the average relative error of the low-rank approximation with the Euclidean distance matrix Θ^t for different percentages of observed entries, over 10 random experiments. Moreover, all algorithms have been adapted to include the rank-1 modification in (2.3). We excluded the EM-NMD algorithm from the analysis because adapting it to the rank-1 modified model appears less straightforward than for the other algorithms. We also consider as a baseline the rank-$(r+1)$ reconstruction given by the original BCD-NMD method solving the 3B-ReLU-NMD of rank $r+1$, without considering any rank-1 modification ($(r+1)$BCD-NMD). In this case, we compute a rank-$(r+1)$ decomposition of X and evaluate the relative error with $dee^T - \Theta^t$. We stop the algorithms when a tolerance of 10^{-9} on the relative residual Γ_k is reached or after 60 seconds. We observe from Figure 5.2 that the distribution of the points affects the recovery performance of the algorithms:

- When the points are randomly distributed, 30% of the entries needs to be observed for the eBCD-NMD to achieve an error below 10^{-7}.

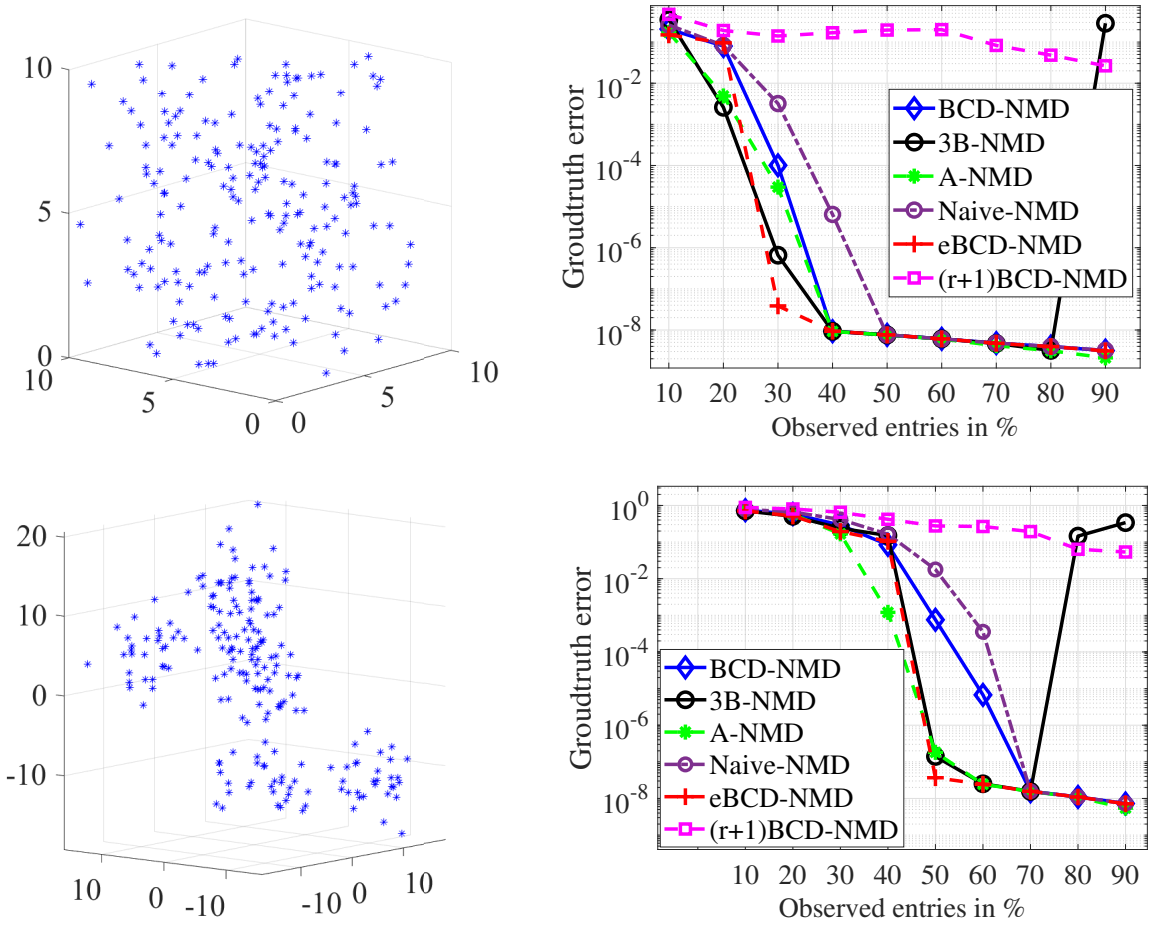

Figure 5.2: Euclidean distance matrix completion. Distribution of points on the left and relative error with the Euclidean distance matrix of the points in the right. The x-axis represents the percentage of known entries.

- When the points are clustered, 50% of the entries needs to be observed for the eBCD-NMD to achieve an error below 10^{-7}.

Including the rank-1 modification in the model allows for a significantly more accurate reconstruction than solving a rank-$(r+1)$ 3B-ReLU-NMD. In general, A-NMD, eBCD-NMD and 3B-NMD provide the best reconstructions. However, 3B-NMD fails at least once in both examples when the percentage of observed entries is above 80%. This behavior might be due to the aggressive extrapolation procedure that 3B-NMD uses. We emphasize that neither the 3B-ReLU-NMD nor the Latent-ReLU-NMD formulations impose any constraints to ensure that the solution is symmetric. However, if the number of observed entries is sufficiently large, the algorithms still recover the original symmetric solution.

5.3. Low-dimensional embedding. In this subsection, we test eBCD-NMD within the threshold similarity matching (TSM) framework of Saul [33] for dimensionality reduction. Specifically, ReLU-NMD is employed to find a lower-dimensional representation of the similarity matrix of the input points ; see Section 2 for the details. Our focus is on text data, where each row of the data matrix corresponds to a document, each column to a word, and the entries (i, j) of the matrix is the number of occurrences

of the jth word in the ith document. We consider two different well-known and already pre-pocessed text-word data sets from [44], that are *k1b* which has 2340 documents and 21839 words and *hitech* that has 2301 documents and 10080 words. The threshold parameter τ in the TSM framework is selected according to the statistical analysis proposed in [33], that yields $\tau = 0.17$ for *k1b* and $\tau = 0.08$ for *hitech*.

Our objective is to evaluate how eBCD-NMD performs in comparison to the state-of-the-art algorithms for computing ReLU decomposition in this context. The quality of the embedding is evaluated using the mean angular deviation (MAD) metric as suggested in [33]. The MAD measures the difference between the angles that the algorithm tries to preserve (namely the ones below $\cos^{-1}(\tau)$). Define $\mathcal{P}(\tau) = \{(i,j) \mid \langle z_i, z_j\rangle - \tau\|z_j\|\|z_i\| > 0\}$, the MAD is given by

$$\Delta = \frac{1}{|\mathcal{P}(\tau)|} \sum_{(i,j)\in\mathcal{P}(\tau)} \left| \cos^{-1}\left(\frac{\langle z_i, z_j\rangle}{\|z_i\|\|z_j\|}\right) - \cos^{-1}\left(\frac{\langle y_i, y_j\rangle}{\|y_i\|\|y_j\|}\right)\right|,$$

where the z_i's are the higher-dimensional points, and the y_i's the lower-dimensional embeddings. Figure 5.3 displays the MAD and average running time per iteration for increasing values of the approximation rank. We run all the algorithms with a fixed time limit of 60 seconds.

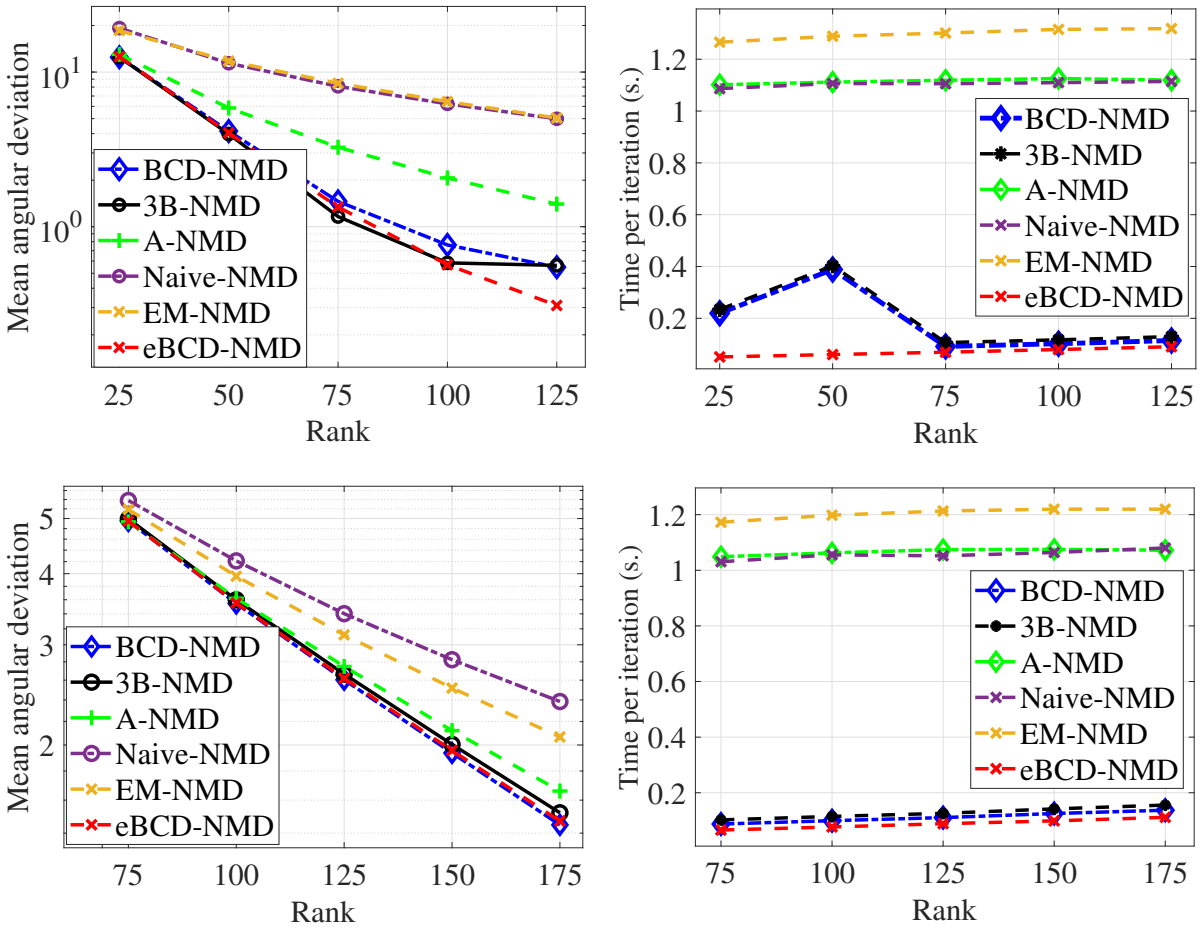

Figure 5.3: MAD analysis and average iteration time for increasing values of the rank in the x-axes. *k1b* dataset on top and *hitech* dataset at the bottom.

Even though we are not able to identify one algorithm that outperforms the others, Figure 5.3 shows that solving 3B-ReLU-NMD is to be preferred in this application. Indeed, BCD-NMD, eBCD-NMD and 3B-NMD achieve the lowest MAD in both experiments. In fact, these three algorithms have the lowest average iteration time, being more than one second faster per iteration than Naive-NMD, A-NMD and EM-NMD. For this reason, BCD-NMD, eBCD-NMD and 3B-NMD perform more iterations than Naive-NMD, A-NMD and EM-NMD in 60 seconds, resulting in more accurate approximations.

5.4. Compression of sparse data. The last application is the compression of sparse data. We consider sparse greyscale images: 10000 randomly selected images from MNIST and Fashion MNIST (fMNIST), both of size 784×10000, and on two well-known sparse images, Phantom (256×256) and Satellite (256×256). We also use sparse, structured matrices from various applications from [10]: Trec11 (235×1138), mycielskian (767×767), lock (1074×1074), and beaconfd (173×295).

We choose the approximation rank r to ensure a desired level of compression. Specifically, when computing the ReLU factors W and H of a sparse matrix $X \in \mathbb{R}^{m \times n}$ with $\text{nnz}(X)$ non zero entries, the number of entries of WH is $r(n+m)$. Hence, the rank r is set to the closest integer such that $r \leq 0.5 \cdot \text{nnz}(X)/(n+m)$, which means we reduced the memory requirement by 50%. We compare eBCD-NMD with state-of-the-art methods for ReLU decomposition, and we compute the TSVD of the same rank as baseline.

Table 5.2 shows the relative error $\frac{\|X - \max(0, \bar{W}\bar{H})\|_F}{\|X\|_F}$, where $\bar{W}$ and $\bar{H}$ are the factors of the rank-r approximation produced by any algorithm. We run each algorithm for 120 seconds and display the average error over 10 different random initializations. We observe that ReLU models outperform the

Data set	BCD-NMD	eBCD-NMD	3B-NMD	Naive-NMD	A-NMD	EM-NMD	TSVD
MNIST	12.4%	11.8%	**11.6%**	12.5%	11.7%	13.2%	25.8%
fMNIST	9.7%	9.3%	9.3%	9.7%	**9.2%**	9.9%	14.0%
Phantom	5.1%	**4.1%**	4.3%	5.4%	4.5%	5.6%	19.2%
Satellite	14.0%	**13.6%**	13.7%	14.4%	14.0%	14.8%	26.0%
Trec11	27.4%	**26.5%**	**26.5%**	27.2%	26.3%	31.1%	57.3%
mycielskian	0.37%	**0.05%**	0.06%	0.53%	0.06%	3.6%	58.0%
lock1074	0.3%	**0.001%**	**0.001%**	1.5%	0.006%	66.7%	67.2%
beaconfd	20.6%	20.4%	**20.0%**	20.9%	20.7%	26.8%	39.1%

Table 5.2: Sparse matrices compression. Comparison of the ReLU decomposition error in percentage on various data sets. The best values are highlighted in bold.

linear alternative TSVD, as all the algorithms yield approximations that are significantly more accurate than TSVD. Additionally, eBCD-NMD and 3B-NMD produce the most accurate solutions in seven out of eight examples; moreover, eBCD-NMD outperforms BCD-NMD, achieving a smaller compression error for all the data sets considered. We also observe that the structure of the matrix has an impact on the effectiveness of the compression using ReLU decomposition.

Finally, we propose two visual examples of 50% compression for MNIST and Phantom in Figure 5.4, where we compare TSVD with the approximation $\max(0, \Theta)$ provided by the eBCD-NMD algorithm. In both instances, one can visually appreciate how the eBCD-NMD approximation is consistently more accurate than TSVD, as further confirmed by the average error reported in Table 5.2.

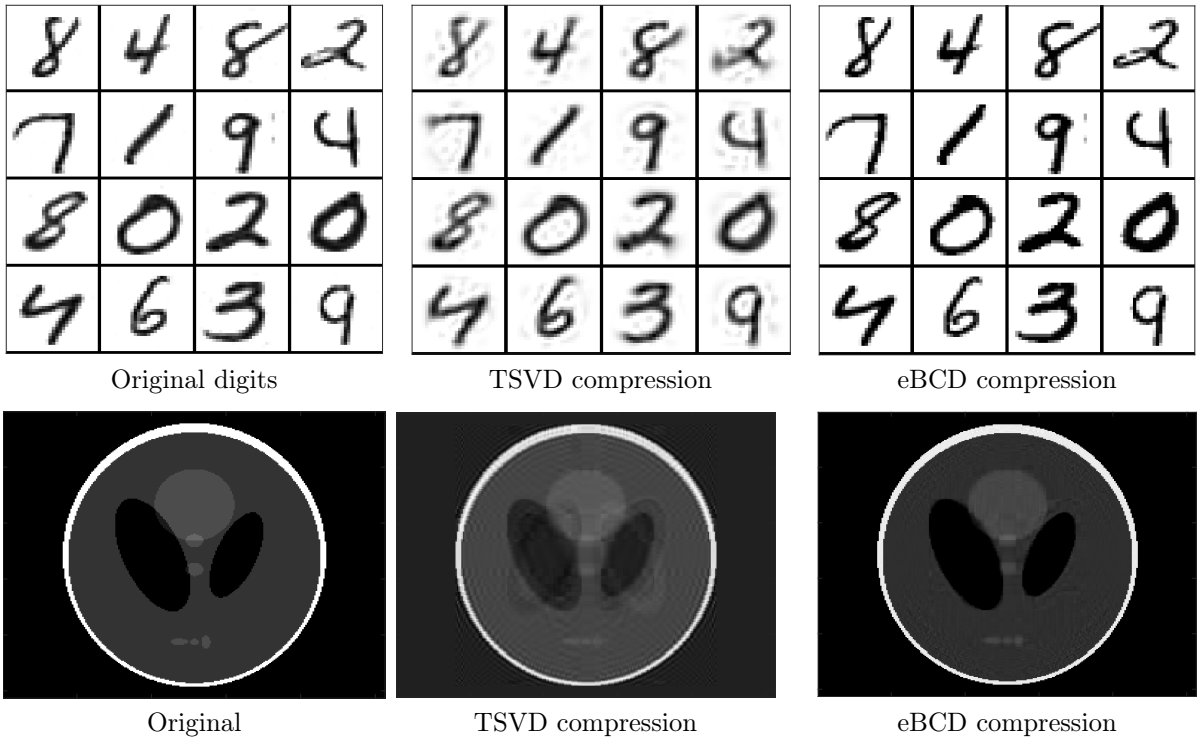

Figure 5.4: Comparison between 50% compressed images using TSVD and eBCD on MNIST and Phantom data sets.

6. Conclusion. The ReLU decomposition is a relatively recent matrix decomposition model (Saul, 2022 [33]) that finds applications in the compression of sparse data, entry-dependent matrix completion, and manifold learning. Moreover, we showed, for the first time, how it can be used for the completion of Euclidean distance matrices when only the smallest (or largest) entries are observed. Our first main contribution is to establish the non-equivalence of f the two main models used to solve the ReLU problem, namely ReLU-NMD and Latent-ReLU-NMD (Corollary 3.3). Furthermore, we derive an explicit connection between the optimal function values of the two, demonstrating that the solution of Latent-ReLU-NMD is in the worst case twice larger than that of ReLU-NMD (Theorem 3.4). Our second main contribution is to prove convergence of the BCD-NMD scheme for solving 3B-ReLU-NMD by demonstrating that it falls into the general framework of exact BCD, where one of the subproblems is strictly quasi-convex (Theorem 4.1). Our third and most important contribution is a novel algorithm, eBCD-NMD, which uses extrapolation to accelerate eBCD-NMD. eBCD-NMD extends the well-known LMaFit algorithm from [43] for matrix completion. We proved subsequence convergence of the eBCD-NMD scheme to a stationary point under mild assumptions (Theorem 4.7). Finally, we provided extensive numerical results for 4 applications on synthetic and real data sets, showing that eBCD-NMD performs favorably compared to the to the state of the art, while having convergence guarantees which other algorithms lack.

Future work includes investigating the well-posedness of the standard ReLU-NMD in (1.1), designing randomized algorithms to tackle large-scale problems, and exploring more advanced ReLU decompositions such as $X \approx W_1H_1 + \max(0, W_2H_2) - \max(0, W_3H_3)$ that could handle mixed-signed and non-sparse data.

Acknowledgments. G.S. and N.G. acknowledge the support by the European Union (ERC consolidator, eLinoR, no 101085607). The work of M.P. and G.S. was partially supported by INdAM-GNCS under the project CUP E53C23001670001. The research of M.P. was partially granted by the Italian Ministry of University and Research (MUR) through the PRIN 2022 "MOLE: Manifold constrained Optimization and LEarning", code: 2022ZK5ME7 MUR D.D. financing decree n. 20428 of November 6th, 2024 (CUP B53C24006410006).

7. Appendix.

7.1. Proof of Lemma 4.3.

Proof. Let $(W(\alpha), H(\alpha))$ be generated by (4.7), and let $S = \mathcal{P}_\Omega(X - WH) - \mathcal{P}_{\Omega^C}(\max(0, WH))$ be the residual with respect to the k-th iterate (W, H). Recall that $Z_\alpha = WH + \alpha S$, $P = H^\top(HH^\top)^{-1}H$, and $E(\alpha) = W(\alpha)W(\alpha)^T$. We denote $Q(\alpha)R(\alpha)\Pi = Z_\alpha H^T$, the QR factorization of $Z_\alpha H^T$ with column pivoting, where Π is a permutation matrix. Note that Π is needed only if $Z_\alpha H^T$ is rank-deficient; otherwise, one can consider Π as the identity matrix. We have

$$W(\alpha)H(\alpha) - WH = \left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1} - W\right] H + W(\alpha)\left[H(\alpha) - R(\alpha)\Pi(HH^T)^{-1}H\right]. \tag{7.1}$$

The first term in (7.1) can be written as

$$\begin{aligned}\left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1} - W\right] H =& W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H - WH \\ =& Z_\alpha H^T(HH^T)^{-1}H - W(HH^T)(HH^T)^{-1}H = Z_\alpha P - WHP \\ =& (Z_\alpha - WH)P = \alpha SP.\end{aligned} \tag{7.2}$$

Similarly, we want to write the second term of the summation in terms of the residual. We use (7.2) to get

$$\begin{aligned}H(\alpha) =& W(\alpha)^T Z_\alpha = W(\alpha)^T(WH + \alpha S) = \\ =& W(\alpha)^T\left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H - (W(\alpha)R(\alpha)\Pi(HH^T)^{-1} - W)H + \alpha S\right] \\ =& W(\alpha)^T\left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H + \alpha S(I - P)\right].\end{aligned}$$

We multiply both sides by $W(\alpha)$ to obtain

$$\begin{aligned}W(\alpha)H(\alpha) =& W(\alpha)W(\alpha)^T\left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H + \alpha S(I - P)\right] \\ =& W(\alpha)R(\alpha)\Pi(HH^T)^{-1}H + \alpha E(\alpha)S(I - P).\end{aligned}$$

Rearranging the terms, we have

$$W(\alpha)\left[H(\alpha) - R(\alpha)\Pi(HH^T)^{-1}H\right] = \alpha E(\alpha)S(I - P). \tag{7.3}$$

Finally, combining (7.1), (7.2), and (7.3) we obtain

$$\begin{aligned}W(\alpha)H(\alpha) - WH =& \left[W(\alpha)R(\alpha)\Pi(HH^T)^{-1} - W\right] H + W(\alpha)\left[H(\alpha) - R(\alpha)\Pi(HH^T)^{-1}H\right] \\ =& \alpha SP + \alpha Q(\alpha)S(I - P) \\ =& \alpha(I - Q(\alpha))SP + \alpha Q(\alpha)S.\end{aligned} \tag{7.4}$$

From the properties of the orthogonal projection, (7.4) implies

$$\|W(\alpha)H(\alpha) - WH\|_F^2 = \alpha^2\|(I - Q(\alpha))SP\|_F^2 + \alpha^2\|Q(\alpha)S\|_F^2, \tag{7.5}$$

which proves the second inequality in (4.13).

Let us now obtain the first inequality. From (7.4) and the properties of orthogonal projection,

$$\begin{aligned}\langle \alpha S, W(\alpha)H(\alpha) - WH\rangle =& \alpha^2\langle S, (I-E(\alpha))SP\rangle + \alpha^2\langle S, E(\alpha)S\rangle \\ =& \alpha^2\|(I-E(\alpha))SP\|_F^2 + \alpha^2\|E(\alpha)S\|_F^2 \\ =& \|W(\alpha)H(\alpha) - WH\|_F^2,\end{aligned} \tag{7.6}$$

From the observation (4.9) on Z_α, the definition of $\sigma_{WH}(\alpha)$, and from (7.6), we get

$$\begin{aligned}\sigma_{WH}(\alpha) =& \|S\|_F^2 - \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - Z_\alpha\|_F^2 = \|S\|_F^2 - \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - WH - \alpha S\|_F^2 \\ =& \|S\|_F^2 - \frac{1}{\alpha^2}\left[\|W(\alpha)H(\alpha) - WH\|_F^2 + \alpha^2\|S\|_F^2 - 2\alpha\langle S, W(\alpha)H(\alpha) - WH\rangle\right] \\ =& \frac{1}{\alpha^2}\|W(\alpha)H(\alpha) - WH\|_F^2,\end{aligned}$$

which proves the first equality in (4.13). ∎

7.2. Proof of Lemma 4.4.

Proof. Let us first show the continuity of the function in (4.14) when $\alpha \to 1^+$. Recall that $Z_1 = Z$ and that $W(\alpha)$ is an orthogonal basis of $\mathcal{R}(Z_\alpha H^T)$. Therefore, the orthogonal projection operator into $\mathcal{R}(Z_\alpha H^T)$ can be written as $W(\alpha)W(\alpha)^T = (Z_\alpha H^T)(Z_\alpha H^T)^\dagger$. If $\operatorname{rank}(Z_\alpha H^T) = \operatorname{rank}(ZH^T)$ in some interval $[1, \widehat{\alpha}]$, then by the continuity of the Moore-Penrose inverse [37], we have

$$\lim_{\alpha\to 1^+} W(\alpha)W(\alpha)^T = \lim_{\alpha\to 1^+}(Z_\alpha H^T)(Z_\alpha H^T)^\dagger = (ZH^T)(ZH^T)^\dagger = W(1)W(1)^T. \tag{7.7}$$

From (7.7) and the continuity of the product,

$$\lim_{\alpha\to 1^+} W(\alpha)H(\alpha) = \lim_{\alpha\to 1^+} W(\alpha)W(\alpha)^T Z_\alpha = W(1)W(1)^T Z = W(1)H(1), \tag{7.8}$$

which leads to the continuity of (4.14) when $\alpha \to 1^+$. Let us recall that

$$Z_\alpha = WH + \alpha S,\ P_{\Omega^C}(S) = -P_{\Omega^C}(\max(0, WH)),\ P_{\Omega^C}(Z(\alpha)) = P_{\Omega^C}(\min(0, (W(\alpha)H(\alpha))).$$

We have

$$\begin{aligned}&\frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_{\Omega^C}(S(\alpha))\|_F^2 \\ =&\frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_{\Omega^C}(W(\alpha)H(\alpha) - Z(\alpha))\|_F^2 \\ =&\frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - WH - \alpha S)\|_F^2 - \|P_{\Omega^C}(W(\alpha)H(\alpha) - \min(0, (W(\alpha)H(\alpha)))\|_F^2 \\ =&\frac{1}{\alpha^2}\|P_{\Omega^C}(W(\alpha)H(\alpha) - \min(0, WH) + (\alpha-1)\max(0, WH))\|_F^2 - \\ &- \|P_{\Omega^C}(W(\alpha)H(\alpha) - \min(0, (W(\alpha)H(\alpha)))\|_F^2.\end{aligned} \tag{7.9}$$

Since (4.14) is continuous for $\alpha \to 1^+$, we just need to show that it is larger than or equal to zero at $\alpha = 1$. Using (7.8) and (7.9),

$$\begin{aligned}&\|P_{\Omega^C}(W(1)H(1) - \min(0, WH))\|_F^2 - \|P_{\Omega^C}(\max(0, (W(1)H(1)))\|_F^2 \\ =&\|P_{\Omega^C}(W(1)H(1))\|_F^2 + \|P_{\Omega^C}(\min(0, WH))\|_F^2 - 2\langle P_{\Omega^C}(W(1)H(1)), P_{\Omega^C}(\min(0, WH))\rangle - \\ &- \|P_{\Omega^C}(\max(0, (W(1)H(1)))\|_F^2 \\ =&\|P_{\Omega^C}(\min(0, W(1)H(1)))\|_F^2 + \|P_{\Omega^C}(\min(0, WH))\|_F^2 - \\ &- 2\langle P_{\Omega^C}(\min(0, W(1)H(1))), P_{\Omega^C}(\min(0, WH))\rangle - 2\langle P_{\Omega^C}(\max(0, W(1)H(1))), P_{\Omega^C}(\min(0, WH))\rangle \\ =&\|P_{\Omega^C}(\min(0, W(1)H(1)) - (\min(0, WH))\|_F^2 - 2\langle P_{\Omega^C}(\max(0, W(1)H(1))), P_{\Omega^C}(\min(0, WH))\rangle \geq 0,\end{aligned}$$

which concludes the proof. ∎

7.3. Proof of Lemma 4.5.

Proof. By an analogous argument to the one used in Lemma 4.5, we get the continuity of (4.15) as $\alpha \to 1^+$. Next, we show that (4.15) is zero at $\alpha = 1$. By the definition of the residual in (4.8),

$$\begin{aligned}
&\frac{1}{\alpha^2}\|P_\Omega(W(\alpha)H(\alpha) - Z_\alpha)\|_F^2 - \|P_\Omega(S(\alpha))\|_F^2 \\
=&\|P_\Omega(W(\alpha)H(\alpha) - X + X - WH - \alpha S)\|_F^2 - \|P_\Omega(S(\alpha))\|_F^2 \\
=&\|-P_\Omega(S(\alpha)) + (1-\alpha)P_\Omega(S))\|_F^2 - \|P_\Omega(S(\alpha))\|_F^2,
\end{aligned}$$

which is equal to zero for $\alpha = 1$. ∎

REFERENCES

[1] Alfakih, A.Y., Khandani, A., Wolkowicz, H.: Solving Euclidean distance matrix completion problems via semidefinite programming. Computational optimization and applications **12**, 13–30 (1999)

[2] Ang, A.M.S., Gillis, N.: Accelerating nonnegative matrix factorization algorithms using extrapolation. Neural computation **31**(2), 417–439 (2019)

[3] Attouch, H., Bolte, J., Redont, P., Soubeyran, A.: Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-łojasiewicz inequality. Mathematics of operations research **35**(2), 438–457 (2010)

[4] Awari, A., Nguyen, H., Wertz, S., Vandaele, A., Gillis, N.: Coordinate descent algorithm for nonlinear matrix decomposition with the ReLU function. In: 2024 32nd European Signal Processing Conference (EUSIPCO), pp. 2622–2626 (2024). DOI 10.23919/EUSIPCO63174.2024.10715025

[5] Balzano, L., Nowak, R., Recht, B.: Online identification and tracking of subspaces from highly incomplete information. In: 2010 48th Annual allerton conference on communication, control, and computing (Allerton), pp. 704–711. IEEE (2010)

[6] Bhattacharya, S., Chatterjee, S.: Matrix completion with data-dependent missingness probabilities. IEEE Transactions on Information Theory **68**(10), 6762–6773 (2022)

[7] Bi, Y., Lavaei, J.: On the absence of spurious local minima in nonlinear low-rank matrix recovery problems. In: International Conference on Artificial Intelligence and Statistics, pp. 379–387. PMLR (2021)

[8] Boumal, N., Absil, P.A.: Low-rank matrix completion via preconditioned optimization on the Grassmann manifold. Linear Algebra and its Applications **475**, 200–239 (2015)

[9] Ciaperoni, M., Gionis, A., Mannila, H.: The Hadamard decomposition problem. Data Mining and Knowledge Discovery pp. 1–42 (2024)

[10] Dai, W., Kerman, E., Milenkovic, O.: A geometric approach to low-rank matrix completion. IEEE Transactions on Information Theory **58**(1), 237–247 (2012)

[11] Delsarte, P., Kamp, Y.: Low rank matrices with a given sign pattern. SIAM Journal on Discrete Mathematics **2**(1), 51–63 (1989)

[12] Fang, H.r., O'Leary, D.P.: Euclidean distance matrix completion problems. Optimization Methods and Software **27**(4-5), 695–717 (2012)

[13] Ganti, R.S., Balzano, L., Willett, R.: Matrix completion under monotonic single index models. Advances in Neural Information Processing Systems **28** (2015)

[14] Golub, G.H., Van Loan, C.F.: Matrix computations. JHU press (2013)

[15] Goyens, F., Cartis, C., Eftekhari, A.: Nonlinear matrix recovery using optimization on the Grassmann manifold. Applied and Computational Harmonic Analysis **62**, 498–542 (2023)

[16] Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear Gauss-Seidel method under convex constraints. Operations research letters **26**(3), 127–136 (2000)

[17] Grippo, L., Sciandrone, M.: Introduction to methods for nonlinear optimization, vol. 152. Springer Nature (2023)

[18] Hyeon-Woo, N., Ye-Bin, M., Oh, T.H.: Fedpara: Low-rank Hadamard product for communication-efficient federated learning. arXiv preprint arXiv:2108.06098 (2021)

[19] Jain, P., Netrapalli, P., Sanghavi, S.: Low-rank matrix completion using alternating minimization. In: Proceedings of the forty-fifth annual ACM symposium on Theory of computing, pp. 665–674 (2013)

[20] Keshavan, R.H., Montanari, A., Oh, S.: Matrix completion from a few entries. IEEE transactions on information theory **56**(6), 2980–2998 (2010)

[21] Keshavan, R.H., Oh, S.: A gradient descent algorithm on the Grassman manifold for matrix completion. tech. rep., Dept. of Electrical Engineering, Stanford University ((2009)

[22] Krislock, N., Wolkowicz, H.: Euclidean distance matrices and applications. Springer (2012)

[23] Lee, D., Sompolinsky, H.: Learning a continuous hidden variable model for binary data. Advances in Neural Information Processing Systems **11** (1998)
[24] Lee, J.A., Verleysen, M.: Nonlinear dimensionality reduction. Springer Science & Business Media (2007)
[25] Li, X.P., Huang, L., So, H.C., Zhao, B.: A survey on matrix completion: perspective of signal processing. arXiv preprint arXiv:1901.10885 (2019)
[26] Liu, H., Wang, P., Huang, L., Qu, Q., Balzano, L.: Symmetric matrix completion with ReLU sampling. arXiv preprint arXiv:2406.05822 (2024)
[27] Loconte, L., Sladek, A.M., Mengel, S., Trapp, M., Solin, A., Gillis, N., Vergari, A.: Subtractive mixture models via squaring: Representation and learning. In: The Twelfth International Conference on Learning Representations (2024)
[28] Mazumdar, A., Rawat, A.S.: Learning and recovery in the ReLU model. In: 2019 57th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 108–115 (2019). DOI 10.1109/ALLERTON.2019.8919900
[29] Naik, R., Trivedi, N., Tarzanagh, D.A., Balzano, L.: Truncated matrix completion an empirical study. In: 2022 30th European Signal Processing Conference (EUSIPCO), pp. 847–851. IEEE (2022)
[30] Nguyen, L.T., Kim, J., Shim, B.: Low-rank matrix completion: A contemporary survey. IEEE Access **7**, 94,215–94,237 (2019)
[31] Patriksson, M.: Decomposition methods for differentiable optimization problems over Cartesian product sets. Computational optimization and applications **9**, 5–42 (1998)
[32] Powell, M.J.: On search directions for minimization algorithms. Mathematical programming **4**, 193–201 (1973)
[33] Saul, L.K.: A geometrical connection between sparse and low-rank matrices and its application to manifold learning. Transactions on Machine Learning Research (2022)
[34] Saul, L.K.: A nonlinear matrix decomposition for mining the zeros of sparse data. SIAM Journal on Mathematics of Data Science **4**(2), 431–463 (2022)
[35] Seraghiti, G., Awari, A., Vandaele, A., Porcelli, M., Gillis, N.: Accelerated algorithms for nonlinear matrix decomposition with the ReLU function. In: 2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP), pp. 1–6. IEEE (2023)
[36] Smaragdis, P., Venkataramani, S.: A neural network alternative to non-negative audio models. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 86–90. IEEE (2017)
[37] Stewart, G.: On the continuity of the generalized inverse. SIAM Journal on Applied Mathematics **17**(1), 33–45 (1969)
[38] Tanner, J., Wei, K.: Low-rank matrix completion by alternating steepest descent methods. Applied and Computational Harmonic Analysis **40**(2), 417–429 (2016)
[39] Tasissa, A., Lai, R.: Exact reconstruction of Euclidean distance geometry problem using low-rank matrix completion. IEEE Transactions on Information Theory **65**(5), 3124–3144 (2018)
[40] Tseng, P.: Decomposition algorithm for convex differentiable minimization. Journal of Optimization Theory and Applications **70**(1), 109–135 (1991)
[41] Wang, Q., Cui, C., Han, D.: A momentum accelerated algorithm for ReLU-based nonlinear matrix decomposition. IEEE Signal Processing Letters **31**, 2865–2869 (2024). DOI 10.1109/LSP.2024.3475910
[42] Wang, Q., Qu, Y., Cui, C., Han, D.: An accelerated alternating partial bregman algorithm for relu-based matrix decomposition. arXiv preprint arXiv:2503.02386 (2025)
[43] Wen, Z., Yin, W., Zhang, Y.: Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. Mathematical Programming Computation **4**(4), 333–361 (2012)
[44] Zhong, S., Ghosh, J.: Generative model-based document clustering: a comparative study. Knowledge and Information Systems **8**, 374–384 (2005)