A DETERMINISTIC OPTIMAL SOLUTION TO NONNEGATIVE MATRIX FACTORIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper presents a new approach to nonnegative matrix factorization (NMF) that directly focuses on the subspace structure of the data instead of the specific samples. The main idea is to find the borders of the data's principal subspace with the nonnegative orthant, which we call nonnegative subspace edges (NoSEs), and construct the factorization according to these NoSEs. We introduce a deterministic algorithm to find NoSEs in linear time, and straight-forward techniques to obtain the desired NMF from these NoSEs. We show that this approach defines a deterministic, optimal, unique, and well-posed solution to NMF. To understand the importance of this result, consider the Moore-Penrose pseudo-inverse, which determines an optimal (minimum-norm) solution to ill-posed linear systems. Analogously, NoSEs provide an optimal (widest-cone) solution to the ill-posed problem of NMF.

1 INTRODUCTION

Nonnegative matrix factorization (NMF) aims to approximate a nonnegative data matrix $\mathbf{X}$ as the product of two nonnegative factors $\mathbf{U}$ and $\mathbf{V}$. Due to is broad applicability in critical applications, a myriad of NMF methods have emerged over the last decades (see surveys [1–9]). However, despite this extensive progress, NMF remains inherently ill-posed, admitting infinitely many feasible solutions. In fact, different algorithms generally produce very different factorizations of the same data. For instance, each of the NMF methods in Figure 1-a yields a significantly different decomposition — an average angle of 49.5° between matching vectors in $\mathbf{U}$. For reference, nonnegative gaussian vectors in the same subspaces have an approximate cosine similarity of 17°. These significantly different solutions inform about key features and relationships in critical applications. For example, they are used to identify key genes in single-cell sequencing [10] or compounds in drug discovery [11], and it is generally unclear which among the many available solutions should be favored.

The source of this ill-posedness has a geometric interpretation: NMF is equivalent to finding one of the *simplest* polyhedral cones that *encapsulates* the given data [7, 9, 12, 13]. By *simplest* we mean that the cone is spanned by as few nonnegative vertex rays as possible. By *encapsulate* we mean that the columns of $\mathbf{X}$ can be written as conic combinations (linear combinations with nonnegative coefficients) of the vertex rays spanning the cone. The vertex rays correspond to the columns of $\mathbf{U}$. Once $\mathbf{U}$ is known, $\mathbf{V} = (\mathbf{U}^\mathsf{T}\mathbf{U})^{-1}\mathbf{U}^\mathsf{T}\mathbf{X}$ is trivially given by the coefficients of $\mathbf{X}$ with respect to $\mathbf{U}$. The challenge is that in general, there exist infinitely many feasible cones (Figure 1-b), each corresponding to a different factorization of $\mathbf{X}$. Hence, it is not surprising that different methods produce different results — all correct in that their factors approximate $\mathbf{X}$, but inconsistent in that they produce different coefficients and representations.

Existing approaches mitigate this type of ill-posedness by enforcing constraints that result in unique solutions (Figure 1-c). Examples include ONMF, sparse NMF, and many more [1–5, 7, 14–18]. However, it has been shown that the conditions for uniqueness are very strong [19, 20]. Alternatively, other unique solutions, like the minimal-cone, have been explored [21–23]. However, these solutions are heavily dependent on the particular sample, and are generally unstable in the sense that small perturbations to the data produce significantly different solutions (Figure 1-d).

This paper delivers two main contributions: (i) we introduce a new solution to NMF, defined by the principal subspace $\mathcal{W}$ containing the data and its one-dimensional intersections with the nonnegative orthant, which we call *nonnegative subspace edges* (NoSEs). In contrast to existing

solutions, ours is well-defined, deterministic, unique, and optimal in the *maximal-coverage* sense — that is, it encapsulates the entire nonnegative portion of $\mathcal{W}$. To give some perspective, this is analogous to the Moore-Penrose pseudo-inverse, which selects the optimal (minimum-norm) solution to ill-posed least-squares problems. Our solution applies to any subspace that intersects non-trivially with the nonnegative orthant — that is, any subspace with a nonnegative component, which is a direct consequence of the nonnegativity condition. Moreover, our solution is stable in that it is identical for any sample in the nonnegative portion of $\mathcal{W}$. (ii) We present a deterministic algorithm to find such solution in linear time. This algorithm can be understood as a purely geometric version of the simplex method [24] and the Avis-Fukuda algorithm [25] with no objective function nor slack variables, that traverses the boundary of our polyhedral cone directly on V-representation instead of H-representation. The main idea is to trace faces of decreasing dimension until finding one NoSE, and then recursively traverse its adjacent faces in order to find all remaining NoSEs.

The rest of the paper is organized as follows. In Section 2 we formally define our new solution and its properties. In Section 3 we describe our algorithm to identify such solution, and derive its theoretical guarantees. In Section 4 we discuss some practical considerations on how to extend our approach to noise, outliers, and more. Section 4 presents a discussion of our approach in the context of existing work. Finally, Section 5 demonstrates the effectiveness of our approach on an extensive series of experiments on synthetic and real data.

2 NoSEs and the Maximal-Coverage Solution

The key feature that distinguishes our approach is that we do *not* focus on the specific samples that we observe (columns in $\mathbf{X}$). Instead, we focus on the low-dimensional structure of such samples, that is, their principal subspace $\mathcal{W}$, which can be trivially computed using a singular value decomposition (SVD). Our solution is given by the vectors in $\mathcal{W}$ lying at the boundary of the nonnegative orthant (see Figure 1-b for some intuition). More formally,

Definition 1. *[Nonnegative subspace edges (NoSEs)] Given an* $\mathrm{m} \times \mathrm{n}$ *data matrix* $\mathbf{X}$*, let* $\mathcal{W}$ *denote its principal subspace of dimension* r*. We define the NoSEs of* $\mathbf{X}$*, or equivalently, the NoSEs of* $\mathcal{W}$*, as the vertex rays of the polyhedral cone obtained by intersecting* $\mathcal{W}$ *with the nonnegative orthant.*

Letting $\mathbf{W}$ denote an orthonormal basis of $\mathcal{W}$, it is clear that the polyhedral cone in Definition 1 is the set $\mathcal{C}_{\mathcal{W}} := \{\mathbf{w} \,:\, \mathbf{w} = \mathbf{W}\boldsymbol{\theta} \geq \mathbf{0},\ \boldsymbol{\theta} \in \mathbb{R}^{\mathrm{r}}\}$. To characterize its vertex rays, we will assume without loss of generality that $\mathrm{r} > 1$; otherwise NMF is a trivial problem that requires no attention, as all columns in $\mathbf{X}$ are co-linear, so any column can be written as a conic combination of any other. We will also assume that both $\mathcal{W}$ and the samples in $\mathbf{X}$ are in general position. More precisely, we will assume that

A1. $\mathcal{W}$ is drawn according to an absolutely continuous distribution with respect to the Lebesgue measure over the subset of the Grassmannian containig all subspaces that cross the nonnegative orthant.

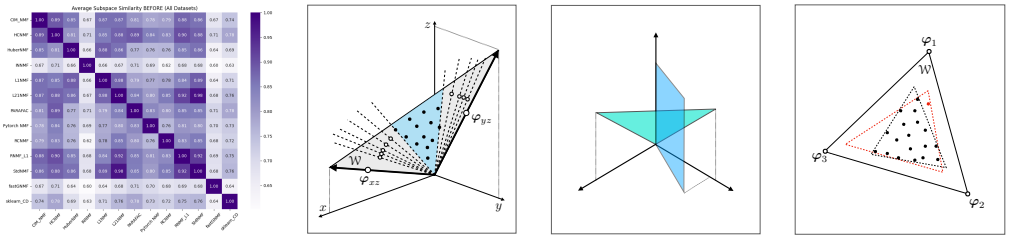
A2. The columns of $\mathbf{X}$ are drawn independently according to an absolutely continuous distribution with respect to the Lebesgue measure over the nonnegative portion of $\mathcal{W}$, that is, the intersection of $\mathcal{W}$ with the nonnegative orthant.

A1-**A2** are standard genericity assumptions [26]. In words, **A1** simply requires that $\mathcal{W}$ is in a random position on the nonnegative orthant, and similarly **A2** requires that the columns in $\mathbf{X}$ are distributed randomly over $\mathcal{W}$. Most nonnegative continuous distributions, for example nonnegative gaussian data, will satisfy these weak assumptions. All our analysis holds with probability 1 under the measures in **A1** and **A2**.

Under these assumptions, it is easy to see that the vertex rays in Definition 1 are the elements of $\mathcal{C}_{\mathcal{W}}$ that satisfy $\mathbf{W}\boldsymbol{\theta} \geq \mathbf{0}$ with exactly r equalities.

Intuitively, NoSEs are simply the 1-dimensional intersections of $\mathcal{W}$ with the canonical faces of the nonnegative orthant (Figure 1-b). To simplify our analysis, we will assume that:

From this definition, we directly obtain the following



(a) (b) (c) (d)

Figure 1: (a) Averaged cosine similarity of different factorizations over 18 datasets. Different NMF algorithms produce different factorizations. **(b)** NMF is equivalent to finding *one* of the lowest-dimensional cones that encapsulate the data (e.g., the blue cone). There exist infinitely many solutions. We propose the maximal-coverage solution (in gray) formed by NoSEs ($\boldsymbol{\varphi}_{xz}, \boldsymbol{\varphi}_{yz}$), which we define as the intersections of the faces of the nonnegative orthant with the principal subspace of the data, $\mathcal{W}$. **(c)** Existing alternatives to mitigate ill-posedness result in overly restrictive conditions. For example, ONMF [34] only allows subspaces perfectly aligned with the canonical axes (e.g., blue), which is a set of measure zero with respect to the Lebesgue measure over the Grassmannian that excludes all subspaces in general position (e.g., teal). In contrast, our work applies to all subspaces with a nonnonnegative component. **(d)** Alternative unique solutions, like the minimal-cone [21–23], are heavily dependent on the sample, and small perturbations can result in entirely different solutions. In this illustration, adding a single sample results in an entirely different minimal-cone solution. In contrast, our maximal-coverage solution is identical for any nonnegative sample in $\mathcal{W}$.

we define the NoSEs of $\mathbf{X}$ as the vertex rays of the polyhedral cone obtained by intersecting $\mathcal{W}$ with the nonnegative orthant. Equivalently, the NoSEs of $\mathbf{X}$ are the 1-dimensional intersections of $\mathcal{W}$ with the canonical faces of the nonnegative orthant. The main idea is that since NoSEs lie at the corners of the nonnegative orthant, their polyhedral cone is *maximally wide* (see Figure 1 for some intuition). Moreover, given $\mathbf{X}$, its set of NoSEs is unique, finite, and deterministic. In this sense, NoSEs single out the *optimal* (widest-cone) solution among the infinitely many available, thus resolving the ill-posedness problem of NMF. To give some perspective, this is analogous to the Moore-Penrose pseudo-inverse, which selects the optimal (minimum-norm) solution to ill-posed least-squares problems. Moreover, since NoSEs lie at the border of the nonnegative orthant, they cannot be represented as conic combinations of any other nonnegative vectors in $\mathcal{W}$. Hence, NoSEs can be used to bound the nonnegative rank. Beyond theory, one practical advantage of our approach is that we can leverage subspace estimation techniques developed over the years to handle a wide range of challenging conditions, including noise [27], outliers [28], missing data [29], sparsity [30], high dimensionality [31], mixtures [32], and more [33]. Our methodology allows us to directly use these techniques to estimate $\mathcal{W}$ robustly, after which its NoSEs and NMF can be computed without further adaptations.

As we will see, finding NoSEs is tantamount to finding the faces of the nonnegative orthant that intersect with $\mathcal{W}$. For example, in Figure 1, the (x, z) and (y, z) faces intersect with $\mathcal{W}$, but the (x, y) face does not. In higher dimensions, there is a combinatorial number of feasible faces, most of which will *not* intersect $\mathcal{W}$. To identify the *good* faces efficiently we propose an explicit, entirely deterministic geometric algorithm that can be understood as a purely geometric version of the simplex method [24] and the Avis-Fukuda algorithm [25] with no objective function nor slack variables, that traverses the boundary of our polyhedral cone directly on V-representation instead of H-representation. The main idea is to trace faces of decreasing dimension until finding one NoSE, and then recursively traverse its adjacent faces in order to find all remaining NoSEs. Our algorithm is guaranteed to identify all NoSEs in linear time.

In general, the number of NoSEs of a given subspace $\mathcal{W}$ depends not only on m and its dimension r, but also on its orientation. That is, two subspaces of $\mathbb{R}^{\mathrm{m}}$ of the same dimension may have a different number of NoSEs. This number will typically be larger than r, and may even be larger than m. However, some NoSEs form fairly obtuse angles and are only required to cover *corners* of the orthant that have small relative volume and hence are unlikely to contain any data (see Figure 1). In general, a few cleverly selected NoSEs will cover large portions of the nonnegative area of their subspace, and will suffice to represent most matrices $\mathbf{X}$, revealing the desired NMF. Given all the NoSEs of $\mathbf{X}$, identifying the few that encapsulate all data can be seen as the column selection problem of identifying the best vectors to form a basis for the data [35–37]. Fortunately, this problem can be

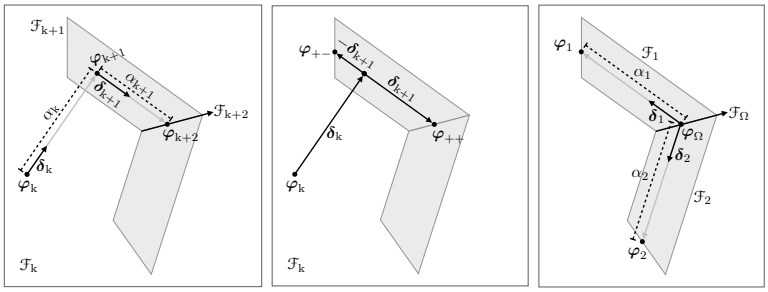


Figure 2: Left: Procedure to find the first NoSE; see Section 3-(i). **Center:** Positive and negative directions yield different NoSEs; see Section 4-(d). **Right:** Procedure to find all remaining NoSEs; see Section 3-(ii).

solved with simple methods [38–40]. We complement our analysis with a discussion on practical considerations that highlight the advantages of our approach, and we present a comprehensive list of experiments that demonstrate its effectiveness on synthetic and real data.

3 Nonnegative Subspace Edges and Where to Find Them

Given a nonnegative $\mathrm{m} \times \mathrm{n}$ data matrix $\mathbf{X}$ of approximate rank r, we define its *nonnegative subspace edges* (NoSEs) as the vertex rays of the polyhedral cone obtained by intersecting $\mathcal{W}$, the r-dimensional principal subspace of $\mathbf{X}$, with the nonnegative orthant.

To characterize NoSEs, let the *canonical face* $\mathcal{F}_\Omega$ be the span of the canonical vectors *not* in $\Omega \subset \{1, \dots, \mathrm{m}\}$, and let $\mathbf{P}_\mathcal{W}$ and $\mathbf{P}_\Omega$ denote the projection operators onto $\mathcal{W}$ and $\mathcal{F}_\Omega$. Given $\mathcal{W}$ and Ω, we say that $\boldsymbol{\varphi}_\Omega \in \mathcal{W} \cap \mathcal{F}_\Omega$ is a NoSE if $\dim(\mathcal{W} \cap \mathcal{F}_\Omega) = 1$ and all the entries in $\boldsymbol{\varphi}_\Omega$ are nonnegative. From this definition, it is clear that whether $\boldsymbol{\varphi}_\Omega$ is a NoSE or not depends entirely on the subspace $\mathcal{W}$ and the subset of coordinates Ω defining the face. It is easy to see from the rank-nulity theorem that for r-dimensional subspaces in general position, $\dim(\mathcal{W} \cap \mathcal{F}_\Omega) = 1$ if and only if $|\Omega| = \mathrm{r} - 1$, or equivalently, if $\dim(\mathcal{F}_\Omega) = \mathrm{m} - (\mathrm{r} - 1)$. In this case, $\boldsymbol{\varphi}_\Omega$ can be trivially computed as the solution to the linear system $(\mathbf{I} - \mathbf{P}_\Omega \mathbf{P}_\mathcal{W})\boldsymbol{\varphi}_\Omega = \mathbf{0}$.

From this characterization, it is clear that one way to identify NoSEs is to simply solve $(\mathbf{I} - \mathbf{P}_\Omega \mathbf{P}_\mathcal{W})\boldsymbol{\varphi}_\Omega = \mathbf{0}$ for every Ω of size $\mathrm{r} - 1$, and see if the solution $\boldsymbol{\varphi}_\Omega$ has no negative entries (in which case $\boldsymbol{\varphi}_\Omega$ is a NoSE). In fact, this strategy may be useful for small problems like the example in Figure 1, where $\mathrm{m} = 3$ and $\mathrm{r} = 2$, resulting in only 3 sets Ω of size $2 - 1 = 1$ that need to be tested. The catch is that in general there are $\binom{\mathrm{m}}{\mathrm{r}-1} = \mathcal{O}(\mathrm{m}^{\mathrm{r}-1})$ such sets Ω, which makes this onerous strategy computationally prohibitive even for problems of moderate size. In other words, the problem is that in general, there are too many faces.

To overcome this problem we introduce the following procedure, which has two main steps: (i) tracing faces of decreasing dimension until we find one NoSE, and (ii) recursively traversing such NoSE's adjacent faces in order to find all remaining NoSEs. We formalize these ideas next, for which we will use the following notations: given a subspace, matrix, or vector that is compatible with a set of indices Ω, we use the superscript Ω to indicate its restriction to the coordinates/rows in Ω. For example, $\mathbf{X}^\Omega$ denotes the $|\Omega| \times \mathrm{n}$ matrix that is equal to $\mathbf{X}$ in the rows indexed in $\Omega \subset \{1, \dots, \mathrm{m}\}$. Finally, $\ominus$ denotes the Hadamard (pointwise) division. We invite the reader to consult Figure 2 as we develop our construction in order to build some geometric intuition.

(i) Finding the first NoSE. The main idea is to start at a *pivot* point in the nonnegative orthant, and move in $\mathrm{r} - 1$ directions of $\mathcal{W}$. Each of these directions will move us to a new pivot point in a lower-dimensional face until we reach a pivot point in a face of dimension $\mathrm{m} - (\mathrm{r} - 1)$. Such point will be the desired NoSE.

We start with our first pivot point $\boldsymbol{\varphi}_0$ equal to the leading left singular vector of $\mathbf{X}$, which will always be in the nonnegative orthant, as $\mathbf{X} \geq \mathbf{0}$ by assumption. Since $\mathcal{W}$ is in general position, $\boldsymbol{\varphi}_0$ has no zero entries, so it lies in the $(\mathrm{m} - 0)$-dimensional face $\mathcal{F}_0$ given by the entire vector space, and $\Omega_0 = \emptyset$. We will now proceed by induction on $\mathrm{k} \in \{0, \dots, \mathrm{r} - 2\}$. Let our pivot $\boldsymbol{\varphi}_\mathrm{k}$ be a

nonnegative vector in $\mathcal{W}$ with k zeros in the entries indexed by $\Omega_\mathrm{k} \subset \{1, \dots, \mathrm{m}\}$, so that $\boldsymbol{\varphi}_\mathrm{k}$ lies in the $(\mathrm{m}-\mathrm{k})$-dimensional face $\mathcal{F}_\mathrm{k}$.

Our goal is to move from $\boldsymbol{\varphi}_\mathrm{k}$ along the face $\mathcal{F}_\mathrm{k}$ in a new direction of $\mathcal{W}$ orthogonal to $\boldsymbol{\varphi}_\mathrm{k}$ that leads to a smaller face $\mathcal{F}_{\mathrm{k}+1}$ of dimension $\mathrm{m}-(\mathrm{k}+1)$ that lies inside $\mathcal{F}_\mathrm{k}$ and contains a nonnegative vector $\boldsymbol{\varphi}_{\mathrm{k}+1}$. To identify such direction $\boldsymbol{\delta}_\mathrm{k}$, let $\{\boldsymbol{\varphi}_\mathrm{k}, \mathbf{w}_1, \dots, \mathbf{w}_{\mathrm{r}+1}\}$ be an orthogonal basis of $\mathcal{W}$, and let $\mathbf{W}_\mathrm{k} := [\mathbf{w}_1, \dots, \mathbf{w}_{\mathrm{r}-1}]$. Recall that $\mathbf{W}_\mathrm{k}^{\Omega_\mathrm{k}}$ denotes the $\mathrm{k} \times (\mathrm{r}-1)$ matrix with the rows of $\mathbf{W}_\mathrm{k}$ indexed in Ω_k. Since $\mathrm{k} < \mathrm{r}-1$ and $\mathcal{W}$ is in general position, the kernel of $\mathbf{W}_\mathrm{k}^{\Omega_\mathrm{k}}$ is non-trivial. Let $\boldsymbol{\gamma}_\mathrm{k} \in \mathbb{R}^{\mathrm{r}-1}$ be any vector in such kernel, and define $\boldsymbol{\delta}_\mathrm{k} := \mathbf{W}_\mathrm{k}\boldsymbol{\gamma}_\mathrm{k}$. By construction, $\boldsymbol{\delta}_\mathrm{k} \in \mathcal{W}$, it is orthogonal to $\boldsymbol{\varphi}_\mathrm{k}$, and lies in $\mathcal{F}_\mathrm{k}$, as $\boldsymbol{\delta}_\mathrm{k}^{\Omega_\mathrm{k}} = \mathbf{W}_\mathrm{k}^{\Omega_\mathrm{k}}\boldsymbol{\gamma}_\mathrm{k} = \mathbf{0}$.

We will move in the direction of $\boldsymbol{\delta}_\mathrm{k}$ to find the smaller face inside $\mathcal{F}_\mathrm{k}$ that we are looking for. More precisely, we will find the $(\mathrm{m}-(\mathrm{k}+1))$-dimensional face $\mathcal{F}_{\mathrm{k}+1}$ that is closest to $\boldsymbol{\varphi}_\mathrm{k}$ in the direction of $\boldsymbol{\delta}_\mathrm{k}$. To identify the distance that we need to move in the direction of $\boldsymbol{\delta}_\mathrm{k}$, let α_k be the smallest, positive, well-defined entry in $\boldsymbol{\alpha}_\mathrm{k} := -\boldsymbol{\varphi}_\mathrm{k} \ominus \boldsymbol{\delta}_\mathrm{k}$, let $\mathrm{i}_{\mathrm{k}+1}$ index its location, and let $\Omega_{\mathrm{k}+1} := \Omega_\mathrm{k} \cup \mathrm{i}_{\mathrm{k}+1}$. Since $\boldsymbol{\delta}_\mathrm{k}$ is orthogonal to $\boldsymbol{\varphi}_\mathrm{k}$, and all the entries of $\boldsymbol{\varphi}_\mathrm{k}$ are nonnegative, $\boldsymbol{\delta}_\mathrm{k}$ must have at least one negative entry, which implies $\boldsymbol{\alpha}_\mathrm{k}$ will always have at least one positive entry. Furthermore, since $\mathcal{W}$ is in general position, all well-defined entries in $\boldsymbol{\alpha}_\mathrm{k}$ are different. This implies that the smallest entry in $\boldsymbol{\alpha}_\mathrm{k}$ is unique, so α_k is well-defined. In words, the i^{th} entry in $\boldsymbol{\alpha}_\ell$ denotes the distance from $\boldsymbol{\varphi}_\Omega$ that we would need to move in the direction of $\boldsymbol{\delta}_\ell$ in order to touch the border of the nonnegative orthant in the i^{th} coordinate. Our next pivot point $\boldsymbol{\varphi}_{\mathrm{k}+1}$ is obtained by moving the smallest such distance α_k in the direction of $\boldsymbol{\delta}_\mathrm{k}$, so that we touch the border of the nonnegative orthant on the $\mathrm{i}_\mathrm{k}^{\mathrm{th}}$ coordinate but do not cross it towards the negative side on any other coordinate. By construction, $\boldsymbol{\varphi}_{\mathrm{k}+1} := \boldsymbol{\varphi}_\mathrm{k} + \alpha_\mathrm{k}\boldsymbol{\delta}_\mathrm{k}$ lies in $\mathcal{W}$, all its entries are nonnegative, and it has $\mathrm{k}+1$ zeros in the entries of $\Omega_{\mathrm{k}+1}$, whence it lies in the $(\mathrm{m}-(\mathrm{k}+1))$-dimensional face $\mathcal{F}_{\mathrm{k}+1}$.

Taking this induction up to $\mathrm{k} = \mathrm{r}-2$, we can obtain a vector $\boldsymbol{\varphi}_{\mathrm{k}+1} = \boldsymbol{\varphi}_{\mathrm{r}-1}$ that is nonnegative, lies in $\mathcal{W}$, and has $\mathrm{r}-1$ zeros, whence lies in a face of dimension $\mathrm{m}-(\mathrm{r}-1)$. This implies $\boldsymbol{\varphi}_{\mathrm{r}-1}$ is a NoSE, as desired. This procedure can be seen as a purely geometric version of the simplex method [24] with no objective function nor slack variables that traverses the boundary of the feasible region directly on V-representation instead of H-representation.

(ii) Finding all remaining NoSEs. We start from a NoSE $\boldsymbol{\varphi}_\Omega$ lying on an $(\mathrm{m}-(\mathrm{r}-1))$-dimensional face $\mathcal{F}_\Omega$ with $\mathrm{r}-1$ zeros indexed by Ω. The main idea is to move out of the face $\mathcal{F}_\Omega$ along all its adjacent higher-dimensional faces $\{\mathcal{F}_\ell\}$ in order to find *all* other NoSEs neighboring $\boldsymbol{\varphi}_\Omega$. It may be useful to think of $\mathcal{F}_\Omega$ as the "corner" where $\boldsymbol{\varphi}_\Omega$ is located, and $\{\mathcal{F}_\ell\}$ as the "walls" coming out of such corner that will in turn lead to other corners. More formally, let $\mathcal{F}_\ell$ denote the $(\mathrm{m}-(\mathrm{r}-2))$-dimensional face spanned by the same vectors as $\mathcal{F}_\Omega$ in addition to the canonical vector corresponding to the ℓ^{th} element in Ω. Our plan is to move from $\boldsymbol{\varphi}_\Omega$ along $\mathcal{F}_\ell$ in all directions of $\mathcal{W}$ orthogonal to $\boldsymbol{\varphi}_\Omega$ to find all its neighboring NoSEs.

To identify these directions, let $\{\boldsymbol{\varphi}_\Omega, \mathbf{w}_1, \dots, \mathbf{w}_{\mathrm{r}-1}\}$ be an orthonormal basis of $\mathcal{W}$, and let $\boldsymbol{\Delta}_\Omega = [\boldsymbol{\delta}_1, \dots, \boldsymbol{\delta}_{\mathrm{r}-1}]$ be a basis of the subspace spanned by $\{\mathbf{w}_1, \dots, \mathbf{w}_{\mathrm{r}-1}\}$ that is in reduced column-echelon form, with the identity block located in the rows of Ω. For example, if $\Omega = \{1, 2, 3\}$ and $\mathrm{m} = 5$, then

$$\mathcal{F}_\Omega = \operatorname{span}\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad \boldsymbol{\Delta}_\Omega = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \# & \# & \# \\ \# & \# & \# \end{bmatrix},$$

where $\#$ denotes a non-zero value. Since $\mathcal{W}$ is in general position, we can always construct such a basis $\boldsymbol{\Delta}_\Omega$. Let Ω_ℓ index the set of $\mathrm{r}-2$ canonical vectors that are *not* in the span of $\mathcal{F}_\ell$. For instance, in our example, $\Omega_1 = \{2, 3\}$.

Notice that $\boldsymbol{\delta}_\ell$ is the only vector in $\boldsymbol{\Delta}_\Omega$ that has zeros in Ω_ℓ, and that adding any combination of other vectors in $\boldsymbol{\Delta}_\Omega$ would induce nonzeros in Ω_ℓ. It follows that $\boldsymbol{\delta}_\ell$ is the only direction of $\mathcal{W}$ orthogonal to $\boldsymbol{\varphi}_\Omega$ along $\mathcal{F}_\ell$, so it is the only direction in $\mathcal{F}_\ell$ that leads to a NoSE. To determine the distance that we must move in this direction to obtain such NoSE, let α_ℓ be the smallest, positive, well-defined entry in $\boldsymbol{\alpha}_\ell := -\boldsymbol{\varphi}_\Omega \ominus \boldsymbol{\delta}_\ell$, let i_ℓ index its location, and let $\Omega'_\ell := \Omega_\ell \cup \mathrm{i}_\ell$. Since $\boldsymbol{\delta}_\ell$ is orthogonal to $\boldsymbol{\varphi}_\Omega$, and all the entries of $\boldsymbol{\varphi}_\Omega$ are nonnegative, $\boldsymbol{\delta}_\ell$ must have at least one negative entry, which implies

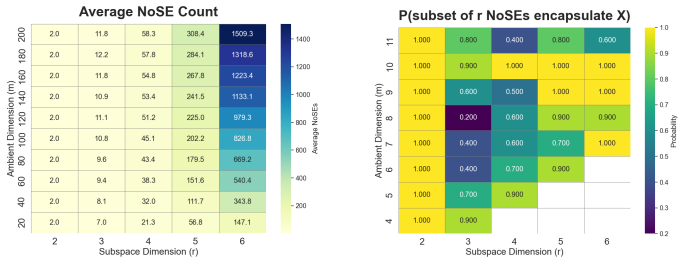


Figure 3: Left: The number of NoSEs depends on m, r, and the orientation of the subspace. Average number of NoSEs (over 20 trials) as a function of m and r. **Right:** Fraction of cases (over 10 trials) where at least one subset of r NoSEs encapsulates $\mathbf{X}$. In these experiments, $\mathbf{U}$ and $\mathbf{V}$ were populated with the absolute value of i.i.d. standard normal entries.

$\boldsymbol{\alpha}_\ell$ will always have at least one positive entry. Furthermore, since $\mathcal{W}$ is in general position, all well-defined entries in $\boldsymbol{\alpha}_\ell$ are different. This implies that the smallest entry in $\boldsymbol{\alpha}_\ell$ is unique, so $\boldsymbol{\alpha}_\ell$ is well-defined. In words, the i^{th} entry in $\boldsymbol{\alpha}_\ell$ denotes the distance from $\boldsymbol{\varphi}_\Omega$ that we would need to move in the direction of $\boldsymbol{\delta}_\ell$ in order to touch the border of the nonnegative orthant in the i^{th} coordinate. The only NoSE in $\mathcal{F}_\ell$ adjacent to $\boldsymbol{\varphi}_\Omega$ is thus obtained by moving the smallest such distance α_ℓ in the direction of $\boldsymbol{\delta}_\ell$, i.e., $\boldsymbol{\varphi}_\ell := \boldsymbol{\varphi}_\Omega + \alpha_\ell \boldsymbol{\delta}_\ell$. The point is to touch the border of the nonnegative orthant on the $\mathrm{i}_\ell^{\mathrm{th}}$ coordinate but to not cross it towards the negative side on any other coordinate.

With this procedure we obtain all the $\mathrm{r}-1$ NoSEs neighboring $\boldsymbol{\varphi}_\Omega$. This immediately implies that all r-dimensional subspaces in general position have *at least* r NoSEs. Since $\mathcal{W}$ is in general position, there is a connecting path of faces from any vertex ray in the cone to any other. hence, repeating this procedure recursively for every newly discovered NoSE we obtain all NoSEs, as desired. Notice that this procedure is linear in the number of NoSEs. We have thus proved the following:

Theorem 1. *Suppose* $\mathrm{r} > 1$. *Under* ***A1*****-*****A2****,* $\mathbf{X}$ *has a finite and deterministic set of* $\mathrm{N} \geq \mathrm{r}$ *NoSEs, which the procedure above is guaranteed to identify in linear time.*

Notice that the procedure above is easily parallelizable implementing simple subscript management techniques like Bland's rule [41] to track recursion efficiently. This procedure can be seen as a purely geometric, simplex-free version of the Avis-Fukuda algorithm [25] that traverses vertex rays directly on V-representation instead of H-representation.

4 From NoSEs to NMF

Let $\boldsymbol{\Phi}$ denote the $\mathrm{m} \times \mathrm{N}$ matrix containing all the NoSEs of $\mathbf{X}$ as columns. The procedure above shows that $\mathrm{N} \geq \mathrm{r} := \mathrm{rank}(\mathbf{X})$. That is, any rank-r matrix will always have at least r NoSEs. However, if $\mathrm{r} > 2$, N is generally much larger than r (see Figure 3). Since NoSEs are the vertex rays of a polyhedral cone, they cannot be represented as conic combinations of any other points in the cone. This means that if $\mathbf{X}$ contains $\mathrm{N}' \leq \mathrm{N}$ NoSEs, its nonnegative rank is the minimum among m, n, and N'. Nonetheless, if the columns of $\mathbf{X}$ are in general position (i.e., drawn according to an absolutely continuous distribution with respect to the Lebesgue measure) over the nonnegative portion of $\mathcal{W}$, the probability that $\mathbf{X}$ contains a NoSE (according to such measure) is zero. In such case, it is possible that there exists a small subset of $\mathrm{N}' < \min(\mathrm{m}, \mathrm{n})$ NoSEs $\boldsymbol{\Phi}'$ that encapsulate $\mathbf{X}$. In such case we can directly upper bound the nonnegative rank by N', and we also obtain an NMF using $\boldsymbol{\Phi}'$ as the first factor. It turns out that for Gaussian data, there typically exist such small subsets of NoSEs that cover $\mathbf{X}$ (see Figure 3). Moreover, given $\boldsymbol{\Phi}$, selecting such subset of N' NoSEs can be seen as an instance of a column subset selection problem [35–37], which can be solved with various alternatives, for example:

a) Clustering. Partition all NoSEs into N' clusters (e.g., using k-means clustering [42], spectral clustering [43, 44], or subspace clustering [45]), and select one representative NoSE from each cluster.

b) QR. Select the most representative columns using a QR decomposition [38–40].

c) Group-lasso. Solve the following optimization:

$$\min_{\boldsymbol{\Theta}\in\mathbb{R}^{\mathrm{N}\times\mathrm{n}}} \|\mathbf{X}-\boldsymbol{\Phi}\boldsymbol{\Theta}\|_{\mathrm{F}} \;+\; \lambda\|\boldsymbol{\Theta}\|_{2,1},\ \text{s.t. } \boldsymbol{\Theta}\geq\mathbf{0}. \tag{1}$$

Here $\|\cdot\|_{\mathrm{F}}$ denotes the Frobenius norm, given by the square root of the sum of squared entries, $\|\cdot\|_{2,1}$ denotes the $\ell_{2,1}$ norm, given by the sum of the ℓ_2 norms of the rows, and $\lambda\geq 0$ is a regularization parameter. The idea behind equation 1 is to find a representation of $\mathbf{X}$ that only uses a few of the NoSEs in $\boldsymbol{\Phi}$. Such NoSEs will be the columns in $\boldsymbol{\Phi}$ corresponding to the nonzero rows of the solution to equation 1, which in turn will be sparse thanks to the regularization term, provided that $\lambda\geq\sqrt{\mathrm{n}/\mathrm{m}}+\sqrt{\log\mathrm{N}/\mathrm{m}}$ [46]. The higher λ, the sparser $\boldsymbol{\Theta}$, so we could tune this parameter to obtain a solution with N' nonzero rows. In our implementation we simply set λ to this bound, and select the rows of $\boldsymbol{\Theta}$ with the N' largest norms.

d) Principal NoSEs. Select the N' NoSEs that cross the nonnegative orthant in the positive and negative principal directions of $\mathbf{X}$. This can be achieved by modifying the method that we described above to find the first NoSE. The only difference is that instead of always moving in the direction of $\boldsymbol{\delta}_{\mathrm{k}}$, we also move in the direction of $-\boldsymbol{\delta}_{\mathrm{k}}$. It is easy to see that different combinations of positive and negative directions will yield different NoSEs (see Figure 2).

The main advantage of (a) is its simplicity and intuitiveness, and that it allows for numerous clustering methods, modifications, and generalizations. Its main caveat is that since it relies on clustering, it is sensitive to initialization. In contrast, (b) is deterministic, and hence produces a unique solution. Similarly, (c) is a convex formulation with a unique global minimizer that standard convex solvers are guaranteed to find. Moreover, since (c) is a standard group-lasso optimization, under the well studied assumptions of this formulation we obtain consistency guarantees [47] and tight error bounds that carry through directly (see for example Theorem 3.1 in [48] or Theorem 2 in [49]). In other words, under standard regularity conditions, (c) is guaranteed to reveal the smallest subset of NoSEs that encapsulate $\mathbf{X}$. In contrast, the greedy version in (d) offers no guarantees, but since it does not require to compute all NoSEs, it can be an attractive option in applications where efficiency is paramount, or when the data is so high-dimensional that computing all NoSEs is infeasible or impractical.

Let $\boldsymbol{\Phi}'$ be the $\mathrm{m}\times\mathrm{N}'$ matrix containing a subset of NoSEs selected from $\boldsymbol{\Phi}$ using any criteria. This matrix can be used as the first factor in our NMF. To obtain the corresponding second factor it suffices to compute the nonnegative coefficients of $\mathbf{X}$ with respect to the basis $\boldsymbol{\Phi}'$. This can be trivially done by solving the following convex linear program, which can be seen as a standard refinement step of equation 1:

$$\boldsymbol{\Theta}' \;:=\; \operatorname*{arg\,min}_{\boldsymbol{\Theta}\in\mathbb{R}^{\mathrm{N}'\times\mathrm{n}}} \|\mathbf{X}-\boldsymbol{\Phi}'\boldsymbol{\Theta}\|_{\mathrm{F}},\ \text{s.t. } \boldsymbol{\Theta}\geq\mathbf{0}. \tag{2}$$

To summarize, the first factor $\mathbf{U}$ in our NMF is given by $\boldsymbol{\Phi}'$, which can be obtained by selecting a subset of N' columns in $\boldsymbol{\Phi}$ using any criteria, for example methods (a)-(d). Our second factor $\mathbf{V}$ is given by $\boldsymbol{\Theta}'$, which can be obtained by solving equation 2. If $\boldsymbol{\Phi}'$ is selected with the QR method, $\boldsymbol{\Phi}'$ and $\boldsymbol{\Theta}'$ are guaranteed to be unique and well-posed, because the QR decomposition is deterministic and unique for generic matrices. Moreover, this solution is optimal in the sense that it reveals the widest cone (lying at the border of the nonnegative orthant) containing $\mathbf{X}$. The same is true with the group-lasso method, because equation 1 is convex.

Practical Considerations: noise, outliers, sparsity, high-dimensionality, and more

One of the main strengths of our approach lies in its focus on the principal subspace of $\mathbf{X}$. Over the past few decades, extensive research has been devoted to estimating this type of subspaces under a wide range of challenging conditions, including noise [27], outliers [28], missing data [29], sparsity [30], high dimensionality [31], mixtures [32], and more [33]. We can directly leverage these advances to estimate $\mathcal{W}$, and subsequently compute its NoSEs and the corresponding NMF without requiring any further adaptations. For example, if $\mathbf{X}$ has outliers, $\mathcal{W}$ can be accurately estimated with any Robust PCA method [28]. Similarly, if $\mathbf{X}$ is incomplete, $\mathcal{W}$ can be estimated with a low-rank matrix completion method [50, 51]. Furthermore, if $\mathbf{X}$ contains a mixture of data lying in a union of subspaces, such union can be estimated using any subspace clustering algorithm [52]. In any of

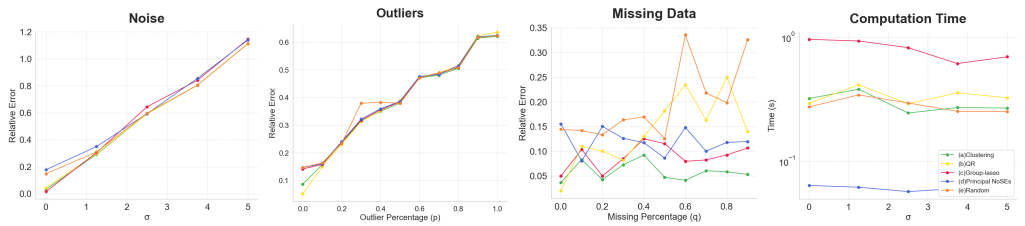


Figure 4: Factorization error and computation time of NoSEs NMF methods (a)-(e) under noise, outliers, and missing data, which our approach can handle seamlessly. The reconstruction error remains comparable across all methods, including a random selection of NoSEs. This suggests that the cones formed by NoSEs are wide enough to encapsulate most data, so most combinations will produce a reasonable approximation.

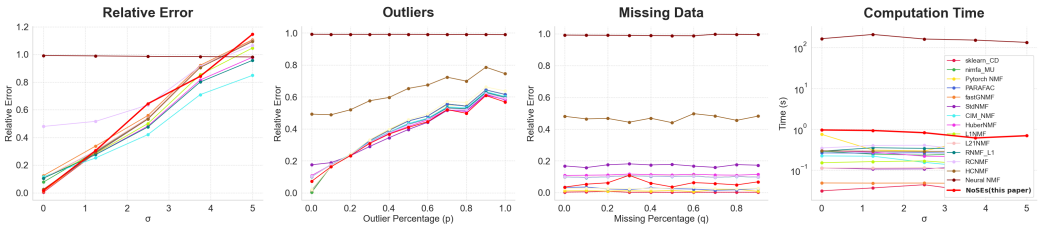


Figure 5: Performance of our NoSEs method and other algorithms on synthetic data. The accuracy of our method is comparable to other state-of-the-art algorithms. Its main advantages are the theoretical guarantees and the deterministic well-posed solution it offers, which do not require strong assumptions like orthogonality [55–57] or that the data satisfy special conditions, like the extreme data property [12]. We also point out that, as seen in Figure 4, our group-lasso version in this plot is the slowest of our methods, so it can be seen an upper bound on the computation time of our other strategies.

these cases, the NoSEs of each subspace (and the corresponding NMF) can be computed without any modifications.

We illustrate this with numerical experiments on noisy data, incomplete data, and data contaminated with outliers, all of which our approach can handle seamlessly. In these experiments we fixed $\mathrm{m} = \mathrm{n} = 200$, $\mathrm{r} = 5$, and populated the matrices $\mathbf{U}$ and $\mathbf{V}$ with the absolute value of i.i.d. standard normal entries. Then we generated a noise matrix $\mathbf{Z}$ with the absolute value of i.i.d. normal entries with variance σ^2, and constructed $\mathbf{X} = \mathbf{UV} + \mathbf{Z}$. To simulate missing data, we independently removed each entry in $\mathbf{X}$ with probability p. Similarly, we induced outliers by independently replacing entries in $\mathbf{X}$ with standard normal values. Next we computed $\mathcal{W}$ either (i) using a singular value decomposition truncating noise at the elbow point, (ii) using a robust PCA algorithm [53], or (iii) using a low-rank matrix completion algorithm [54]. After estimating $\mathcal{W}$, we calculated $\mathbf{\Phi}'$ and $\mathbf{\Theta}'$ using the methods above with $\mathrm{N}' = \mathrm{r}$. Finally, we constructed our factorization $\hat{\mathbf{X}} := \mathbf{\Phi}'\mathbf{\Theta}'$ and measured the normalized approximation error $\|\mathbf{X} - \hat{\mathbf{X}}\|_{\mathrm{F}}/\|\mathbf{X}\|_{\mathrm{F}}$. Figure 4 shows that the reconstruction error remains proportional to the range of noise, outliers, and missing data. Interestingly, all methods have a comparable performance, including a random selection of NoSEs. This suggests that the cones formed by NoSEs are wide enough to encapsulate most data, and even though there is a unique optimal combination of NoSEs, most combinations will produce a reasonable approximation. Figure 4 also shows a speed comparison.

5 Experiments

There exists such a plethora of NMF algorithms (see surveys [1–9] for a yet incomplete list) that it is impossible to compare them all. So, in our experiments we compare our approach against 14 well-established methods from the literature, covering a wide range of approaches, spanning classical methods to state-of-the-art (listed in Tables 1 and 2). To measure accuracy we use the normalized approximation error $\|\mathbf{X} - \hat{\mathbf{X}}\|_{\mathrm{F}}/\|\mathbf{X}\|_{\mathrm{F}}$, where $\hat{\mathbf{X}} = \hat{\mathbf{U}}\hat{\mathbf{V}}$ denotes the factorization obtained by each method. To measure efficiency we report the computation time of each method. All experiments were conducted on a computer with an AMD Ryzen 7 5800H CPU, 16 GB RAM, and an NVIDIA RTX 3060 Ti GPU (8 GB).

Table 1: Reconstruction Errors across Datasets.

Method	Heart Disease	Iris	Seeds	Housing	news20	Olivetti	ORL	Jaffe	COIL100	COIL20	EYaleB	Flowers	Oxford Pet	Reuters	Semeion	USPS	CIFAR10	MNIST
sklearn_CD[76]	0.070	**0.040**	**0.042**	0.047	**0.901**	**0.103**	0.193	0.142	0.392	**0.202**	0.399	0.442	0.389	**0.795**	**0.439**	**0.225**	0.348	0.451
Pytorch NMF[77]	0.070	0.041	0.043	0.045	0.907	0.136	**0.112**	**0.122**	**0.215**	0.243	**0.171**	**0.244**	**0.169**	0.819	0.470	0.264	**0.171**	**0.393**
fastGNMF[78]	0.085	0.053	0.078	0.058	0.902	0.168	0.168	0.172	0.243	0.242	0.213	0.282	0.222	0.798	0.449	0.254	0.226	0.487
StdNMF[79]	0.084	0.061	0.049	0.130	0.908	0.135	0.136	0.151	0.264	0.268	0.222	0.305	0.236	0.812	0.540	0.321	0.226	0.497
CIM_NMF[80]	0.137	0.091	0.079	0.281	1.005	0.205	0.161	0.212	0.376	0.505	0.763	0.392	0.320	0.959	1.240	0.579	0.298	0.804
HuberNMF[81]	0.105	0.066	0.064	0.248	0.970	0.142	0.143	0.162	0.287	0.297	0.243	0.317	0.249	0.910	0.674	0.344	0.238	0.657
L1NMF[82]	0.084	0.061	0.049	0.130	0.971	0.168	0.136	0.151	0.264	0.358	0.222	0.305	0.236	0.918	0.710	0.422	0.227	0.497
L21NMF[83]	0.085	0.061	0.050	0.140	0.909	0.135	0.137	0.152	0.266	0.273	0.223	0.305	0.237	0.812	0.540	0.322	0.227	0.500
RNMF_L1[81]	0.105	0.062	0.057	0.229	0.909	0.135	0.186	0.220	0.432	0.271	0.448	0.549	0.508	0.813	0.575	0.322	0.420	0.816
RCNMF[84]	0.384	0.113	0.279	0.547	0.915	0.480	0.479	0.553	0.521	0.532	0.480	0.548	0.475	0.835	0.659	0.514	0.476	0.622
HCNMF[84]	0.069	0.067	0.044	0.041	0.937	0.138	0.103	0.107	0.197	0.253	0.141	0.216	0.137	0.852	0.525	0.259	0.146	0.325
nimfa_MU[85]	0.084	0.061	0.049	0.048	**0.901**	0.132	0.134	0.139	0.244	0.233	0.199	0.284	0.217	0.796	0.455	0.260	0.213	0.405
PARAFAC[86]	0.072	0.041	0.046	**0.044**	**0.901**	0.111	0.113	0.126	0.224	0.213	0.177	0.259	0.188	0.796	0.445	0.241	0.189	0.398
NoSEs (this paper)	**0.069**	**0.040**	**0.042**	0.058	0.967	0.155	0.154	0.191	0.323	0.389	0.397	0.407	0.310	0.909	0.683	0.492	0.315	0.427
Neural NMF[87]	*timed out*																	

Table 2: Running Time (seconds) across Datasets.

Method	Heart Disease	Iris	Seeds	Housing	news20	Olivetti	ORL	Jaffe	COIL100	COIL20	EYaleB	Flowers	Oxford Pet	Reuters	Semeion	USPS	CIFAR10	MNIST
sklearn_CD[76]	**0.004**	0.005	**0.006**	**0.005**	18.014	4.456	**0.061**	**0.022**	**0.352**	10.900	**0.120**	**2.407**	**2.148**	11.586	1.302	3.097	**4.283**	**0.887**
Pytorch NMF[77]	0.215	0.223	0.408	0.651	**0.656**	**0.366**	1.222	0.682	0.841	**0.294**	0.831	3.960	4.732	**0.441**	0.378	**0.391**	9.370	1.976
fastGNMF[78]	0.043	0.018	0.025	0.099	148.746	2.357	0.857	0.404	39.969	2.832	4.922	114.984	103.932	170.170	1.426	16.961	442.388	158.718
StdNMF[79]	0.013	0.014	0.014	0.015	25.343	5.758	1.975	1.159	33.587	5.758	9.524	137.088	132.719	27.774	1.610	6.591	324.566	72.910
CIM_NMF[80]	0.023	0.024	0.035	0.026	5.401	0.402	2.538	0.830	32.456	1.315	20.758	125.983	152.819	4.849	3.157	0.493	453.814	33.286
HuberNMF[81]	0.058	0.046	0.044	0.057	175.834	13.930	4.857	2.032	66.827	12.105	23.861	279.761	265.155	161.812	3.690	15.906	842.488	142.193
L1NMF[82]	0.016	0.020	0.025	0.024	39.524	11.230	4.234	1.754	59.001	10.592	20.048	243.010	232.003	50.451	2.929	13.025	900.738	140.035
L21NMF[83]	0.025	0.029	0.037	0.020	36.817	9.734	2.196	1.442	49.670	9.556	16.581	228.411	215.554	39.578	2.612	10.963	468.635	94.002
RNMF_L1[81]	0.049	0.017	0.051	0.054	25.131	6.217	3.613	1.407	42.146	6.334	15.281	187.131	175.378	31.549	2.563	8.069	580.251	95.601
RCNMF[84]	0.178	0.044	0.087	0.380	554.563	9.389	4.376	1.578	252.260	14.611	23.307	864.188	721.167	502.722	6.184	97.429	4767.069	1481.410
HCNMF[84]	0.066	0.044	0.048	0.043	269.927	21.151	8.297	2.380	63.571	18.641	29.559	400.267	396.735	252.588	5.174	24.251	1064.065	116.737
nimfa_MU[85]	0.031	0.018	0.025	0.095	151.401	11.219	4.535	1.715	59.447	12.973	19.339	314.665	305.291	155.837	3.287	13.473	696.579	132.182
PARAFAC[86]	0.048	0.025	0.083	0.044	12.387	2.380	1.128	0.595	13.163	3.288	5.269	72.613	76.975	16.685	**1.396**	2.535	121.054	21.215
NoSEs (this paper)	0.006	**0.002**	**0.006**	0.009	74.919	4.894	0.975	0.251	23.690	3.539	4.019	198.264	467.520	78.945	2.184	9.546	355.480	137.835
Neural NMF[87]	*timed out*																	

Synthetic data. In these experiments we generate $\mathbf{X}$ the same way as we described in Section 4, and use each of the methods above to obtain an NMF. The results are in Figure 5, together with a discussion. To avoid clutter, we compare only against our group-lasso version (c), but the curious reader can scrutinize the performance of the rest of our strategies (a)-(d) in Figure 4. Missing data and outliers were artificially incorporated the same way as in our simulations. In the spirit of fairness, we used the same low-rank matrix completion [54] or robust PCA algorithm [53] to pre-process the data before using each NMF algorithm.

Real data. We evaluate our method on 18 real datasets related to image processing, text analysis, document classification, and more. The datasets are Heart Disease [58], Iris [59], Seeds [60], Housing [61], news20 [62], Olivetti [63], ORL [64], Jaffe [65, 66], COIL100 and COIL20 [67], EYaleB [68], Flowers [69], Oxford Pet [70], Reuters [71], Semeion [72], USPS [73], CIFAR10 [74], and MNIST [75]. More details on these datasets can be found in Appendix B. In each case, the subspace dimension r was identified as the elbow point in a singular value decomposition. Tables 1 and 2 show the results, with copies in larger font in Tables 3 and 4 in Appendix B. Unfortunately, Neural NMF timed out in these larger datasets, in agreement to the high computational complexity it exhibited in our simulations in Figure 5, and the factors obtained by HCNMF contained negative values. Nonetheless we include these methods for completeness. It is worth noticing that while no method dominates across all datasets, our approach outperforms the state-of-the-art in the first three, and performs nearly as well (within a negligible margin of at most 0.04) on datasets 4 through 8.

In light of these comparable results, a skeptical reader might ask: *why should one prefer our solution?* A related question provides some perspective: why is the Moore-Penrose pseudo-inverse often favored when solving an underdetermined linear system? The reasoning in both cases is similar. The Moore-Penrose pseudo-inverse identifies a well-posed (unique), deterministic, and optimal (minimum-norm) solution to ill-posed linear systems with multiple solutions. Like these underdetermined systems, NMF is also an ill-posed problem with infinitely many solutions. In fact, each of the NMF methods tested in our real-data experiments yields a significantly different factorization — an average angle of 49.5° between matching vectors in $\mathbf{U}$; see Figures 6 and 7 in Appendix C for details. For reference, nonnegative gaussian vectors in the same subspaces have an approximate cosine similarity of 17°. These significantly different factorizations inform about key features and relationships in critical applications. For example, they are used to identify key genes in single-cell sequencing [10] or compounds in drug discovery [11], and it is generally unclear which among the many available solutions should be favored. Like the Moore-Penrose pseudo-inverse, our solution to NMF has the desirable properties of being well-posed (unique), deterministic, and optimal (widest-cone), without requiring strong assumptions like orthogonality [55–57] or that the data satisfy special conditions, like the extreme data property [12].

REFERENCES

[1] L. Li and Y.-j. ZHANG, "A survey on algorithms of non-negative matrix factorization," *ACTA ELECTONICA SINICA*, vol. 36, no. 4, p. 737, 2008.

[2] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on knowledge and data engineering*, vol. 25, no. 6, pp. 1336–1353, 2012.

[3] Z. Huang, A. Zhou, and G. Zhang, "Non-negative matrix factorization: a short survey on methods and applications," in *Computational Intelligence and Intelligent Systems: 6th International Symposium, ISICA 2012, Wuhan, China, October 27-28, 2012. Proceedings*. Springer, 2012, pp. 331–340.

[4] T. Li and C.-c. Ding, "Nonnegative matrix factorizations for clustering: A survey," *Data Clustering*, pp. 149–176, 2018.

[5] J. Gan, T. Liu, L. Li, and J. Zhang, "Non-negative matrix factorization: A survey," *The Computer Journal*, vol. 64, no. 7, pp. 1080–1092, 2021.

[6] S. Fathi Hafshejani and Z. Moaberfard, "Initialization for non-negative matrix factorization: a comprehensive review," *International Journal of Data Science and Analytics*, pp. 1–16, 2022.

[7] W.-S. Chen, Q. Zeng, and B. Pan, "A survey of deep nonnegative matrix factorization," *Neurocomputing*, vol. 491, pp. 305–320, 2022.

[8] F. Saberi-Movahed, K. Berahman, R. Sheikhpour, Y. Li, and S. Pan, "Nonnegative matrix factorization in dimensionality reduction: A survey," *arXiv preprint arXiv:2405.03615*, 2024.

[9] Y.-T. Guo, Q.-Q. Li, and C.-S. Liang, "The rise of nonnegative matrix factorization: Algorithms and applications," *Information Systems*, p. 102379, 2024.

[10] J. A. Johnson, A. P. Tsang, J. T. Mitchell, D. L. Zhou, J. Bowden, E. Davis-Marcisak, T. Sherman, T. Liefeld, M. Loth, L. A. Goff *et al.*, "Inferring cellular and molecular processes in single-cell data with non-negative matrix factorization using python, r and genepattern notebook implementations of cogaps," *Nature protocols*, pp. 1–42, 2023.

[11] Y. Zhong, C. Seoighe, and H. Yang, "Non-negative matrix factorization combined with kernel regression for the prediction of adverse drug reaction profiles," *Bioinformatics Advances*, vol. 4, no. 1, p. vbae009, 2024.

[12] B. Klingenberg, J. Curry, and A. Dougherty, "Non-negative matrix factorization: Ill-posedness and a geometric algorithm," *Pattern Recognition*, vol. 42, no. 5, pp. 918–928, 2009.

[13] S. Yang, X. Zhang, Y. Yao, S. Cheng, and L. Jiao, "Geometric nonnegative matrix factorization (gnmf) for hyperspectral unmixing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2696–2703, 2015.

[14] N. Del Buono and G. Pio, "Non-negative matrix tri-factorization for co-clustering: an analysis of the block matrix," *Information Sciences*, vol. 301, pp. 13–26, 2015.

[15] A. Čopar, B. Zupan, and M. Zitnik, "Fast optimization of non-negative matrix tri-factorization," *PloS one*, vol. 14, no. 6, p. e0217994, 2019.

[16] A. M. S. Ang and N. Gillis, "Algorithms and comparisons of nonnegative matrix factorizations with volume regularization for hyperspectral unmixing," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 12, pp. 4843–4853, 2019.

[17] D.-I. Lee and S. Roy, "Grinch: simultaneous smoothing and detection of topological units of genome organization from sparse chromatin contact count matrices with matrix factorization," *Genome biology*, vol. 22, no. 1, p. 164, 2021.

[18] B. Baur, D.-I. Lee, J. Haag, D. Chasman, M. Gould, and S. Roy, "Deciphering the role of 3d genome organization in breast cancer susceptibility," *Frontiers in Genetics*, vol. 12, p. 2804, 2022.

[19] D. Donoho and V. Stodden, "When does non-negative matrix factorization give a correct decomposition into parts?" *Advances in neural information processing systems*, vol. 16, 2003.

[20] R. C. Henry, "Current factor analysis receptor models are ill-posed," *Atmospheric Environment (1967)*, vol. 21, no. 8, pp. 1815–1820, 1987.

[21] Y. Chen, J. Mairal, and Z. Harchaoui, "Fast and robust archetypal analysis for representation learning," 2014. [Online]. Available: https://arxiv.org/abs/1405.6472

[22] H. Javadi and A. Montanari, "Non-negative matrix factorization via archetypal analysis," 2017. [Online]. Available: https://arxiv.org/abs/1705.02994

[23] G. Barbarino and N. Gillis, "On the robustness of the successive projection algorithm," 2025. [Online]. Available: https://arxiv.org/abs/2411.16195

[24] G. B. Dantzig, "Linear programming and extensions," 2016.

[25] D. Avis and K. Fukuda, "A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra," in *Proceedings of the seventh annual symposium on Computational geometry*, 1991, pp. 98–104.

[26] D. L. Pimentel-Alarcón, N. Boston, and R. D. Nowak, "A characterization of deterministic sampling patterns for low-rank matrix completion," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 4, pp. 623–636, 2016.

[27] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and L. Zhang, "Robust principal component analysis with complex noise," in *International Conference on Machine Learning*, 2014, pp. 55–63.

[28] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[29] E. J. Candès and T. Tao, "The power of convex relaxation: Near-optimal matrix completion," *IEEE Transactions on Information Theory*, vol. 56, no. 5, pp. 2053–2080, 2010.

[30] T. T. Cai, Z. Ma, and Y. Wu, "Sparse pca: Optimal rates and adaptive estimation," *The Annals of Statistics*, vol. 41, no. 6, pp. 3074–3110, 2013.

[31] I. M. Johnstone and D. Paul, "Pca in high dimensions: An orientation," *Proceedings of the IEEE*, vol. 106, no. 8, pp. 1277–1292, 2018.

[32] R. Vidal, Y. Ma, and S. Sastry, "Generalized principal component analysis (gpca)," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 12, pp. 1945–1959, 2005.

[33] M. Greenacre, P. J. Groenen, T. Hastie, A. I. d'Enza, A. Markos, and E. Tuzhilina, "Principal component analysis," *Nature Reviews Methods Primers*, vol. 2, no. 1, p. 100, 2022.

[34] J. P. Nkurunziza, F. Nahayo, and N. Gillis, "Orthogonal nonnegative matrix factorization with the kullback-leibler divergence," 2024. [Online]. Available: https://arxiv.org/abs/2410.07786

[35] J. A. Tropp, "Column subset selection, matrix factorization, and eigenvalue optimization," in *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2009, pp. 978–986.

[36] C. Boutsidis, M. W. Mahoney, and P. Drineas, "An improved approximation algorithm for the column subset selection problem," in *Proceedings of the twentieth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2009, pp. 968–977.

[37] A. Sood and T. Hastie, "A statistical view of column subset selection," *arXiv preprint arXiv:2307.12892*, 2023.

[38] T. F. Chan, "Rank revealing qr factorizations," *Linear algebra and its applications*, vol. 88, pp. 67–82, 1987.

[39] M. Gu and S. C. Eisenstat, "Efficient algorithms for computing a strong rank-revealing qr factorization," *SIAM Journal on Scientific Computing*, vol. 17, no. 4, pp. 848–869, 1996.

[40] A. Matakos, B. Ordozgoiti, and S. Thejaswi, "Fair column subset selection," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 2189–2199.

[41] R. G. Bland, "New finite pivoting rules for the simplex method," *Mathematics of operations Research*, vol. 2, no. 2, pp. 103–107, 1977.

[42] D. Arthur and S. Vassilvitskii, "K-means++ the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007, pp. 1027–1035.

[43] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[44] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[45] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[46] S. N. Negahban, P. Ravikumar, M. J. Wainwright, and B. Yu, "A Unified Framework for High-Dimensional Analysis of M-Estimators with Decomposable Regularizers," *Statistical Science*, vol. 27, no. 4, pp. 538 – 557, 2012. [Online]. Available: https://doi.org/10.1214/12-STS400

[47] F. R. Bach, "Consistency of the group lasso and multiple kernel learning." *Journal of Machine Learning Research*, vol. 9, no. 6, 2008.

[48] K. Lounici, M. Pontil, S. Van De Geer, and A. B. Tsybakov, "Oracle inequalities and optimal inference under group sparsity," 2011.

[49] A. Dedieu, "An error bound for lasso and group lasso in high dimensions," *arXiv preprint arXiv:1912.11398*, 2019.

[50] L. T. Nguyen, J. Kim, and B. Shim, "Low-rank matrix completion: A contemporary survey," *IEEE Access*, vol. 7, pp. 94 215–94 237, 2019.

[51] K.-L. Du, M. Swamy, Z.-Q. Wang, and W. H. Mow, "Matrix factorization techniques in machine learning, signal processing, and statistics," *Mathematics*, vol. 11, no. 12, p. 2674, 2023.

[52] W. Qu, X. Xiu, H. Chen, and L. Kong, "A survey on high-dimensional subspace clustering," *Mathematics*, vol. 11, no. 2, p. 436, 2023.

[53] H. Cai, J.-F. Cai, and K. Wei, "Accelerated alternating projections for robust principal component analysis," 2019. [Online]. Available: https://arxiv.org/abs/1711.05519

[54] P. Zilber and B. Nadler, "Gnmr: A provable one-line algorithm for low rank matrix recovery," *SIAM Journal on Mathematics of Data Science*, vol. 4, no. 2, pp. 909–934, 2022.

[55] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2006, pp. 126–135.

[56] S. Choi, "Algorithms for orthogonal nonnegative matrix factorization," in *2008 ieee international joint conference on neural networks (ieee world congress on computational intelligence)*. IEEE, 2008, pp. 1828–1832.

[57] F. Pompili, N. Gillis, P.-A. Absil, and F. Glineur, "Two algorithms for orthogonal nonnegative matrix factorization with application to clustering," *Neurocomputing*, vol. 141, pp. 15–25, 2014.

[58] S. W. P. M. Janosi, Andras and R. Detrano, "Heart Disease," UCI Machine Learning Repository, 1988, DOI: https://doi.org/10.24432/C52P4X.

[59] R. A. Fisher, "Iris," UCI Machine Learning Repository, 1936, DOI: https://doi.org/10.24432/C56C76.

[60] N. J. K. P. K. P. Charytanowicz, Magorzata and S. Lukasik, "Seeds," UCI Machine Learning Repository, 2010, DOI: https://doi.org/10.24432/C5H30K.

[61] D. Harrison and D. L. Rubinfeld, "Housing data set," http://lib.stat.cmu.edu/datasets/boston, 1996, accessed: 2024-05-20.

[62] T. Mitchell, "Twenty Newsgroups," UCI Machine Learning Repository, 1997, DOI: https://doi.org/10.24432/C5C323.

[63] AT&T Laboratories Cambridge, "The olivetti face database," https://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html, accessed: 2025-05-13.

[64] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in *Proceedings of 1994 IEEE workshop on applications of computer vision*. IEEE, 1994, pp. 138–142.

[65] M. J. Lyons, ""excavating ai" re-excavated: Debunking a fallacious account of the jaffe dataset," 2021. [Online]. Available: https://arxiv.org/abs/2107.13998

[66] M. J. Lyons, M. Kamachi, and J. Gyoba, "Coding facial expressions with gabor wavelets (ivc special issue)," 2020. [Online]. Available: https://zenodo.org/record/4029679

[67] S. A. Nene, S. K. Nayar, H. Murase *et al.*, "Columbia object image library (coil-20)," 1996.

[68] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[69] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[70] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar, "Cats and dogs," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[71] D. Lewis, "Reuters-21578 Text Categorization Collection," UCI Machine Learning Repository, 1987, DOI: https://doi.org/10.24432/C52G6M.

[72] "Semeion Handwritten Digit," UCI Machine Learning Repository, 1998, DOI: https://doi.org/10.24432/C5SC8V.

[73] J. Hull, "A database for handwritten text recognition research," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 5, pp. 550–554, 1994.

[74] A. Krizhevsky, "Learning multiple layers of features from tiny images," University of Toronto, Toronto, ON, Canada, Technical Report, 2009, computer Science Department.

[75] L. Deng, "The mnist database of handwritten digit images for machine learning research," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[76] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *the Journal of machine Learning research*, vol. 12, pp. 2825–2830, 2011.

[77] yoyolicoris, "pytorch-NMF: Nonnegative matrix factorization in pytorch," https://github.com/yoyolicoris/pytorch-NMF, 2019.

[78] M. Zitnik and K. Sucipto, "fastGNMF: Fast graph-regularized nmf," https://github.com/mims-harvard/fastGNMF, 2020.

[79] D. Lee and H. Seung, "Algorithms for non-negative matrix factorization," *Adv. Neural Inform. Process. Syst.*, vol. 13, 02 2001.

[80] A. Hamza and D. Brady, "Reconstruction of reflectance spectra using robust nonnegative matrix factorization," *IEEE Transactions on Signal Processing*, vol. 54, no. 9, pp. 3637–3642, 2006.

[81] E. Y. Lam, "Non-negative matrix factorization for images with laplacian noise," in *APCCAS 2008 - 2008 IEEE Asia Pacific Conference on Circuits and Systems*, 2008, pp. 798–801.

[82] H. Gao, F. Nie, W. Cai, and H. Huang, "Robust capped norm nonnegative matrix factorization: Capped norm nmf," ser. CIKM '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 871–880. [Online]. Available: https://doi.org/10.1145/2806416.2806568

[83] L. Zhang, Z. Chen, M. Zheng, and X. He, "Robust non-negative matrix factorization," *Frontiers of Electrical and Electronic Engineering in China*, vol. 6, no. 2, pp. 192–200, 2011.

[84] L. Du, X. Li, and Y.-D. Shen, "Robust nonnegative matrix factorization via half-quadratic minimization," in *2012 IEEE 12th International Conference on Data Mining*, 2012, pp. 201–210.

[85] M. Zitnik and B. Zupan, "Nimfa: A python library for nonnegative matrix factorization," *Journal of Machine Learning Research*, vol. 13, pp. 849–853, 2012.

[86] J. Kossaifi, Y. Panagakis, A. Anandkumar, and M. Pantic, "Tensorly: Tensor learning in python," 2018. [Online]. Available: https://arxiv.org/abs/1610.09555

[87] T. Will, R. Zhang, E. Sadovnik, M. Gao, J. Vendrow, J. Haddock, D. Molitor, and D. Needell, "Neural nonnegative matrix factorization for hierarchical multilayer topic modeling," 2023. [Online]. Available: https://arxiv.org/abs/2303.00058

[88] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[89] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[90] R. Peharz and F. Pernkopf, "Sparse nonnegative matrix factorization with ℓ_0-constraints," *Neurocomputing*, vol. 80, no. 1, pp. 38–46, Mar. 2012.

[91] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, "A deep semi-nmf model for learning hidden representations," in *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ser. ICML'14. JMLR.org, 2014, p. II–1692–II–1700.

[92] C. Fevotte and N. Dobigeon, "Nonlinear hyperspectral unmixing with robust nonnegative matrix factorization," *IEEE Transactions on Image Processing*, vol. 24, no. 12, p. 4810–4819, Dec. 2015. [Online]. Available: http://dx.doi.org/10.1109/TIP.2015.2468177

[93] A. Marmoret and J. Cohen, "nn_fac: Nonnegative factorization techniques toolbox," https://gitlab.inria.fr/amarmore/nonnegative-factorization, 2020, bSD 3-Clause "New" or "Revised" License.

[94] D. I. Lee and S. Roy, "Grinch: simultaneous smoothing and detection of topological units of genome organization from sparse chromatin contact count matrices with matrix factorization," *Genome Biology*, vol. 22, p. 164, 2021. [Online]. Available: https://doi.org/10.1186/s13059-021-02378-z

[95] B. Baur, D. I. Lee, J. Haag, D. Chasman, M. Gould, and S. Roy, "Deciphering the role of 3d genome organization in breast cancer susceptibility," *Frontiers in Genetics*, vol. 12, p. 788318, Jan. 2022.

[96] C. Zeng, J. Tian, and Y. Xu, "Analyze the robustness of three nmf algorithms (robust nmf with l1 norm, l2-1 norm nmf, l2 nmf)," 2023. [Online]. Available: https://arxiv.org/abs/2312.01357

[97] K. Abe and T. Shimamura, "Unmf: a unified nonnegative matrix factorization for multi-dimensional omics data," *Briefings in Bioinformatics*, vol. 24, no. 5, p. bbad253, 07 2023. [Online]. Available: https://doi.org/10.1093/bib/bbad253

[98] Y. Li, J. Chen, C. Chen, L. Yang, and Z. Zheng, "Contrastive deep nonnegative matrix factorization for community detection," 2024. [Online]. Available: https://arxiv.org/abs/2311.02357

[99] Z. Wang and W. Min, "Graph regularized nmf with l20-norm for unsupervised feature learning," 2024. [Online]. Available: https://arxiv.org/abs/2403.10910

[100] F. Saberi-Movahed, B. Biswas, P. Tiwari, J. Lehmann, and S. Vahdati, "Deep nonnegative matrix factorization with joint global and local structure preservation," *Expert Systems with Applications*, vol. 249, p. 123645, 2024. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417424005116

[101] L. L. Z. T. G. E. K. D. O. B. M. A. W. J. R. N. K. W. C. Q. D. Ran Gu, Yevgeny Rakita and S. J. L. Billinge, "Stretched non-negative matrix factorization," *NPJ Computational Materials*, vol. 10, p. 45, 2024.

[102] A. Zito and J. W. Miller, "Compressive bayesian non-negative matrix factorization for mutational signatures analysis," 2025. [Online]. Available: https://arxiv.org/abs/2404.10974

[103] L. Kassab, E. George, D. Needell, H. Geng, N. J. Nia, and A. Li, "Towards a fairer non-negative matrix factorization," 2024. [Online]. Available: https://arxiv.org/abs/2411.09847

[104] K. Subramani, P. Smaragdis, T. Higuchi, and M. Souden, "Rethinking non-negative matrix factorization with implicit neural representations," 2024. [Online]. Available: https://arxiv.org/abs/2404.04439

[105] H. Chen, L. Liu, X. Xiu, and W. Liu, "Adaptive multi-order graph regularized nmf with dual sparsity for hyperspectral unmixing," 2025. [Online]. Available: https://arxiv.org/abs/2503.19258

[106] M. G. . Y. D. Yi Ru, "Robust self-supervised symmetric nonnegative matrix factorization for graph clustering," *Scientific Reports*, vol. 15, p. 7350, 2025. [Online]. Available: https://doi.org/10.1038/s41598-025-92564-x

[107] W. Xiong, Y. Ma, C. Zhang, and S. Liu, "Dual graph-regularized sparse robust adaptive non-negative matrix factorization," *Expert Systems with Applications*, vol. 281, p. 127594, 04 2025.

[108] Y. Torabi, S. Shirani, and J. P. Reilly, "Large language model-based nonnegative matrix factorization for cardiorespiratory sound separation," 2025. [Online]. Available: https://arxiv.org/abs/2502.05757

[109] R. Haba, M. Ohzeki, and K. Tanaka, "Relaxation-assisted reverse annealing on nonnegative/binary matrix factorization," 2025. [Online]. Available: https://arxiv.org/abs/2501.02114

[110] J. Chen, L. Huang, and Y. Wei, "Coseparable nonnegative tensor factorization with t-cur decomposition," 2025. [Online]. Available: https://arxiv.org/abs/2401.16836

[111] J. Chew, W. Diepeveen, and D. Needell, "Curvature corrected nonnegative manifold data factorization," 2025. [Online]. Available: https://arxiv.org/abs/2502.15124

[112] D.-I. Lee and S. Roy, "Examining dynamics of three-dimensional genome organization with multi-task matrix factorization," *bioRxiv*, 2023. [Online]. Available: https://www.biorxiv.org/content/early/2023/08/27/2023.08.25.554883

A Related Work

Non-negative matrix factorization (NMF) has evolved markedly since its inception, reflecting growing demands for robustness, structure, and scalability. Early work revealed that non-negativity constraints yield interpretable, parts-based image decompositions, sparking widespread adoption [88]. Over the following decade, researchers incorporated domain priors and computational refinements: graph-regularized NMF preserved local manifold geometry [89], open-source libraries unified dozens of update rules [85], and exact sparsity was enforced through ℓ_0-constrained formulations [90]. Multi-layer "Deep Semi-NMF" architectures were introduced to capture hierarchical semantics [91], while robust formulations embedded β-divergence and $\ell_{2,1}$ penalties to mitigate outliers in hyperspectral unmixing [92].

As GPU acceleration and toolkits matured, the NMF ecosystem diversified further. High-performance implementations enabled efficient divergence minimization on modern hardware [77], and large-scale factorization methods supported a variety of loss functions [93]. Simultaneously, domain-specific pipelines emerged: graph-regularized clustering frameworks were applied to Hi-C chromatin data [94], and multitask models incorporated network enhancement layers [95].

Between 2023 and 2025, novel NMF variants proliferated, driven by the community's pursuit of adaptability and performance. A comprehensive survey by [9] cataloged over a hundred algorithmic variants and categorized modern NMF developments into structured, constrained, and generalized classes, highlighting applications across 130 fields and calling for case-specific formulations to address performance limitations. Building on this taxonomy, hierarchical layers were learned using neural modules in end-to-end architectures [87], and comparative studies evaluated ℓ_1, ℓ_2, and $\ell_{2,1}$-norm variants under different corruption regimes, revealing no universally superior divergence [96]. Unified NMF frameworks integrated multi-omics datasets through shared latent factors [97], and contrastive learning techniques aligned topological and attribute embeddings for community detection [98].

In 2024, further specialization emerged: graph-regularized NMF was enhanced with $\ell_{2,0}$ row-sparsity and provably convergent solvers [99]; deep architectures incorporated both global and local regularization [100]; per-component stretch parameters were introduced for interpretable material factors [101]; Bayesian priors were used to prune redundant components [102]; fairness-aware formulations equalized reconstruction errors across demographic groups [103]; and NMF layers were reinterpreted as implicit neural operators [104].

By 2025, adaptive graph learning and robust self-supervision took center stage. New models jointly learned multi-hop graphs and dual sparsity [105], embedded robust PCA within symmetric NMF [106], and combined dual graph structures with correntropy losses for handling heterogeneous noise [107]. Hybrid pipelines matured, incorporating large language models for guided sound separation [108] and leveraging quantum annealing for accelerating non-negative or binary factorization [109]. Tensor and geometric generalizations also emerged, including coseparable tensor factorization [110] and curvature-corrected models [111]. Tree-guided approaches extended NMF to capture hierarchical structure in time-series data [112].

Despite these advances, NMF remains fundamentally ill-posed. Factorizations are non-unique, highly sensitive to initialization and model choices, and lack a systematic method for verifying correctness. This unresolved identifiability problem is one of the main motivation of our approach.

B Real Datasets

Below we give a brief description of all the real datasets used in our experiments.

- ORL [64]: A facial recognition dataset containing 400 images with 40 classes(individuals), where $\mathrm{m}, \mathrm{n} = 1024, 400$ and $\mathrm{r} = 30$.
- COIL20 [67]: Columbia object image library with 1440 images of 20 objects photographed at different angles, where $\mathrm{m}, \mathrm{n} = 1024, 1440$ and $\mathrm{r} = 45$.
- Extended Yale-B [68]: A facial image dataset with 38 classes(individuals) under varying lighting condition with 64 images per class,where $\mathrm{m}, \mathrm{n} = 2016, 2432$ and $\mathrm{r} = 40$.

- Flowers [69]: A image classificaion dataset with 102 flower categories with 40 to 258 images per class, where $\mathrm{m}, \mathrm{n} = 3072, 8189$ and $\mathrm{r} = 125$.
- Oxford Pets [70]: Image based dataset of pet animals under various lighting and poses with 37 classes of 200 sample images each, where $\mathrm{m}, \mathrm{n} = 3072, 7349$ and $\mathrm{r} = 160$.
- 20 Newsgroups [62]: A popular text dataset of 20k documents across 20 categories primarily used for document classification, where $\mathrm{m}, \mathrm{n} = 1000, 18846$ and $\mathrm{r} = 50$.
- Reuters-21578 [71]: A text categorization benchmark dataset consisting of Reuters newswire articles of 90 classes, where $\mathrm{m}, \mathrm{n} = 1000, 19043$ and $\mathrm{r} = 60$.
- Semeion [72]: A handwritten digit dataset of 1593 binary images, where $\mathrm{m}, \mathrm{n} = 256, 1593$ and $\mathrm{r} = 50$.
- USPS [73]: THe U.S. Postal Service digit recognition dataset containing 9298 images of digits, where $\mathrm{m}, \mathrm{n} = 256, 7291$ and $\mathrm{r} = 28$.
- Heart Disease [58]: A heart disease prediction dataset with 303 samples(patient records) and 13 clinical features, where $\mathrm{m}, \mathrm{n} = 13, 302$ and $\mathrm{r} = 2$.
- IRIS [59]: A classification and clustering dataset comprising of 150 samples from three iris flower species, where $\mathrm{m}, \mathrm{n} = 4, 150$ and $\mathrm{r} = 2$.
- Olivetti [63]: A face recognition dataset with 400 images of 40 individuals under different posing and lightning conditions, where $\mathrm{m}, \mathrm{n} = 4096, 400$ and $\mathrm{r} = 34$.
- Boston Housing [61]: A non-image regression dataset with 506 samples and 4 features used to predict housing prices, where $\mathrm{m}, \mathrm{n} = 14, 505$ and $\mathrm{r} = 3$.
- Seeds [60]: A dataset containing 210 samples describing the measurements of geometrical properties of kernels from three different varieties of wheat, where $\mathrm{m}, \mathrm{n} = 7, 210$, and $\mathrm{r} = 2$.
- JAFFE [65, 66]: A Japanese female facial expression dataset containing 213 images of 10 individuals, where $\mathrm{m}, \mathrm{n} = 1024, 213$ and $\mathrm{r} = 13$.
- COIL100 [67]: An extension of COIL20 with 100 objects, where $\mathrm{m}, \mathrm{n} = 1024, 7200$ and $\mathrm{r} = 52$.
- CIFAR-10 [74]: A popular image classification dataset with 60k, 32x32 pixel images with 10 classes. We randomly selected 20k samples for evaluation. $\mathrm{m}, \mathrm{n} = 3072, 20000$ and $\mathrm{r} = 100$.
- and MNIST [75]. An extremely popular benchmark dateset of handwritten digit dataset with 70k 28x28 images. We use a subset of 20k randomly selected samples. $\mathrm{m}, \mathrm{n} = 784, 20000$ and $\mathrm{r} = 50$.

Tables 3 and 4 below contain the same results as Tables 1 and 2 in larger font.

C Solutions Disparity

Figures 6 and 7 show the cosine similarity of the solutions of multiple NMF algorithms. Given the solutions $\mathbf{U}$ and $\mathbf{U}'$ of two methods, we found the best match between the columns in $\mathbf{U}$ and $\mathbf{U}'$, and computed the average cosine similarity between each matching column. Notice that a simple principal angle distance does not apply in the NMF setting, because it would be zero if the two bases span the same subspace (which is often the case), even if the two bases $\mathbf{U}$ and $\mathbf{U}'$ are entirely different.

Table 3: Reconstruction Errors across Datasets.

Method	Heart Disease	Iris	Seeds	Housing	news20	Olivetti	ORL	Jaffe	COIL100	COIL20	EYaleB	Flowers	Oxford Pet	Reuters	Semeion	USPS	CIFAR10	MNIST
sklearn_CD[76]	0.070	**0.040**	**0.042**	0.047	**0.901**	**0.103**	0.193	0.142	0.392	**0.202**	0.399	0.442	0.389	**0.795**	**0.439**	**0.225**	0.348	0.451
Pytorch NMF[77]	0.070	0.041	0.043	0.045	0.907	0.136	**0.112**	**0.122**	**0.215**	0.243	**0.171**	**0.244**	**0.169**	0.819	0.470	0.264	**0.171**	**0.393**
fastGNMF[78]	0.085	0.053	0.078	0.058	0.902	0.168	0.168	0.172	0.243	0.242	0.213	0.282	0.222	0.798	0.449	0.254	0.226	0.487
StdNMF[79]	0.084	0.061	0.049	0.130	0.908	0.135	0.136	0.151	0.264	0.268	0.222	0.305	0.236	0.812	0.540	0.321	0.226	0.497
CIM_NMF[80]	0.137	0.091	0.079	0.281	1.005	0.205	0.161	0.212	0.376	0.505	0.763	0.392	0.320	0.959	1.240	0.579	0.298	0.804
HuberNMF[81]	0.105	0.066	0.064	0.248	0.970	0.142	0.143	0.162	0.287	0.297	0.243	0.317	0.249	0.910	0.674	0.344	0.238	0.657
L1NMF[82]	0.084	0.061	0.049	0.130	0.971	0.168	0.136	0.151	0.264	0.358	0.222	0.305	0.236	0.918	0.710	0.422	0.227	0.497
L21NMF[83]	0.085	0.061	0.050	0.140	0.909	0.135	0.137	0.152	0.266	0.273	0.223	0.305	0.237	0.812	0.540	0.322	0.227	0.500
RNMF_L1[81]	0.105	0.062	0.057	0.229	0.909	0.135	0.186	0.220	0.432	0.271	0.448	0.549	0.508	0.813	0.575	0.322	0.420	0.816
RCNMF[84]	0.384	0.113	0.279	0.547	0.915	0.480	0.479	0.553	0.521	0.532	0.480	0.548	0.475	0.835	0.659	0.514	0.476	0.622
HCNMF[84]	0.069	0.067	0.044	0.041	0.937	0.138	0.103	0.107	0.197	0.253	0.141	0.216	0.137	0.852	0.525	0.259	0.146	0.325
nimfa_MU[85]	0.084	0.061	0.049	0.048	**0.901**	0.132	0.134	0.139	0.244	0.233	0.199	0.284	0.217	0.796	0.455	0.260	0.213	0.405
PARAFAC[86]	0.072	0.041	0.046	**0.044**	**0.901**	0.111	0.113	0.126	0.224	0.213	0.177	0.259	0.188	0.796	0.445	0.241	0.189	0.398
NoSEs (this paper)	**0.069**	**0.040**	**0.042**	0.058	0.967	0.155	0.154	0.191	0.323	0.389	0.397	0.407	0.310	0.909	0.683	0.492	0.315	0.427
Neural NMF[87]	*timed out*																	

Table 4: Running Time (seconds) across Datasets.

Method	Heart Disease	Iris	Seeds	Housing	news20	Olivetti	ORL	Jaffe	COIL100	COIL20	EYaleB	Flowers	Oxford Pet	Reuters	Semeion	USPS	CIFAR10	MNIST
sklearn_CD[76]	**0.004**	0.005	**0.006**	**0.005**	18.014	4.456	**0.061**	**0.022**	**0.352**	10.900	**0.120**	**2.407**	**2.148**	11.586	1.302	3.097	**4.283**	**0.887**
Pytorch NMF[77]	0.215	0.223	0.408	0.651	**0.656**	**0.366**	1.222	0.682	0.841	**0.294**	0.831	3.960	4.732	**0.441**	0.378	**0.391**	9.370	1.976
fastGNMF[78]	0.043	0.018	0.025	0.099	148.746	2.357	0.857	0.404	39.969	2.832	4.922	114.984	103.932	170.170	1.426	16.961	442.388	158.718
StdNMF[79]	0.013	0.014	0.014	0.015	25.343	5.758	1.975	1.159	33.587	5.758	9.524	137.088	132.719	27.774	1.610	6.591	324.566	72.910
CIM_NMF[80]	0.023	0.024	0.035	0.026	5.401	0.402	2.538	0.830	32.456	1.315	20.758	125.983	152.819	4.849	3.157	0.493	453.814	33.286
HuberNMF[81]	0.058	0.046	0.044	0.057	175.834	13.930	4.857	2.032	66.827	12.105	23.861	279.761	265.155	161.812	3.690	15.906	842.488	142.193
L1NMF[82]	0.016	0.020	0.025	0.024	39.524	11.230	4.234	1.754	59.001	10.592	20.048	243.010	232.003	50.451	2.929	13.025	900.738	140.035
L21NMF[83]	0.025	0.029	0.037	0.020	36.817	9.734	2.196	1.442	49.670	9.556	16.581	228.411	215.554	39.578	2.612	10.963	468.635	94.002
RNMF_L1[81]	0.049	0.017	0.051	0.054	25.131	6.217	3.613	1.407	42.146	6.334	15.281	187.131	175.378	31.549	2.563	8.069	580.251	95.601
RCNMF[84]	0.178	0.044	0.087	0.380	554.563	9.389	4.376	1.578	252.260	14.611	23.307	864.188	721.167	502.722	6.184	97.429	4767.069	1481.410
HCNMF[84]	0.066	0.044	0.048	0.043	269.927	21.151	8.297	2.380	63.571	18.641	29.559	400.267	396.735	252.588	5.174	24.251	1064.065	116.737
nimfa_MU[85]	0.031	0.018	0.025	0.095	151.401	11.219	4.535	1.715	59.447	12.973	19.339	314.665	305.291	155.837	3.287	13.473	696.579	132.182
PARAFAC[86]	0.048	0.025	0.083	0.044	12.387	2.380	1.128	0.595	13.163	3.288	5.269	72.613	76.975	16.685	**1.396**	2.535	121.054	21.215
NoSEs (this paper)	0.006	**0.002**	**0.006**	0.009	74.919	4.894	0.975	0.251	23.690	3.539	4.019	198.264	467.520	78.945	2.184	9.546	355.480	137.835
Neural NMF[87]	*timed out*																	

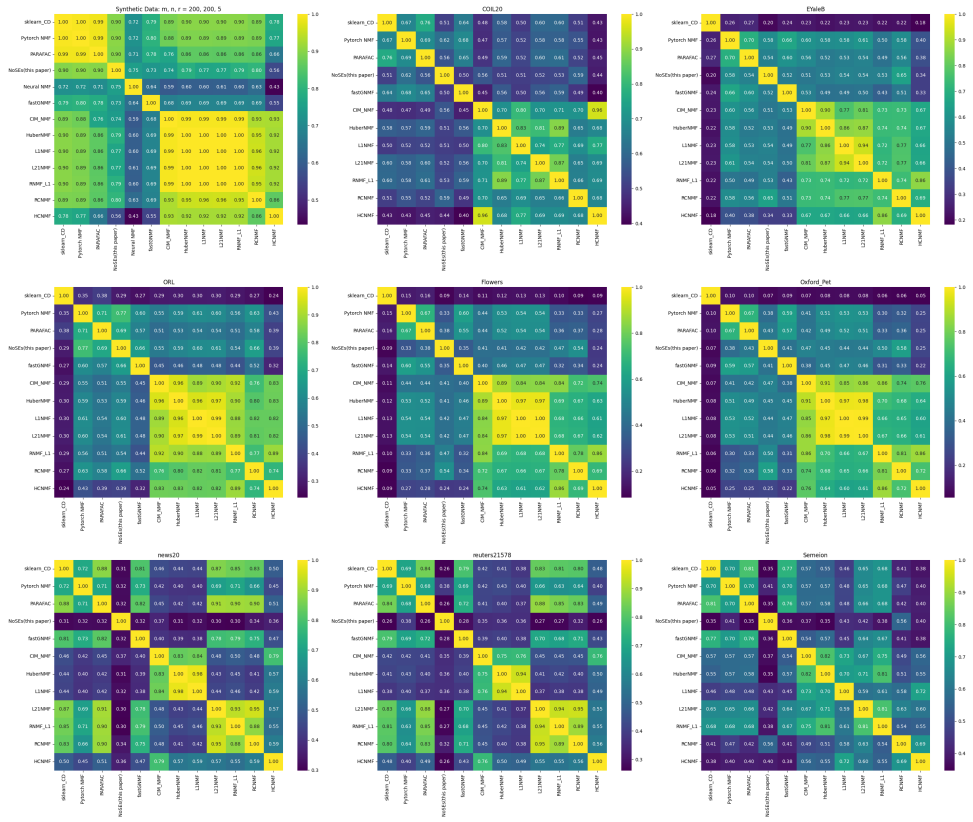


Figure 6: Cosine similarity between NMF solutions of different methods. The overall average across all methods and all datasets is, 0.6496, which is equivalent to a 49.5° angle.

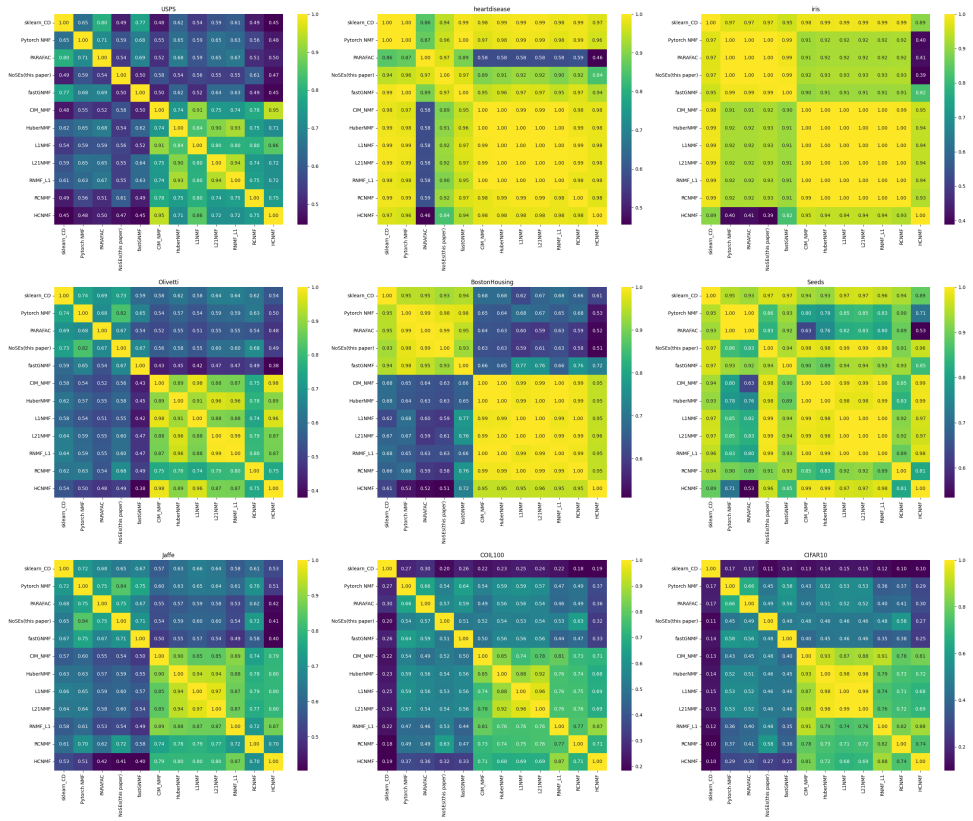


Figure 7: Cosine similarity between NMF solutions of different methods. The overall average across all methods and all datasets is, 0.6496, which is equivalent to a 49.5° angle.