

Deep Nonnegative Matrix Factorization with Beta Divergences

Valentin Leplat¹ Le Thi Khanh Hien² Akwum Onwunta³ Nicolas Gillis²

¹ Center for Artificial Intelligence Technology, Skoltech, Moscow, Russia*

² Department of Mathematics and Operational Research, University of Mons, Mons, Belgium[†]

³ Department of Industrial and Systems Engineering, Lehigh University, Bethlehem, USA[‡]

Abstract

Deep Nonnegative Matrix Factorization (deep NMF) has recently emerged as a valuable technique for extracting multiple layers of features across different scales. However, all existing deep NMF models and algorithms have primarily centered their evaluation on the least squares error, which may not be the most appropriate metric for assessing the quality of approximations on diverse datasets. For instance, when dealing with data types such as audio signals and documents, it is widely acknowledged that β -divergences offer a more suitable alternative. In this paper, we develop new models and algorithms for deep NMF using β -divergences. Subsequently, we apply these techniques to the extraction of facial features, the identification of topics within document collections, and the identification of materials within hyperspectral images.

Keywords. deep nonnegative matrix factorization, β -divergences, minimum-volume regularization, identifiability, multi-block nonconvex optimization, facial feature extraction, topic modeling, hyperspectral unmixing

1 Introduction

Deep NMF seeks to approximate an input data matrix $X \in \mathbb{R}_+^{m \times n}$, as follows:

$$X \approx W_1 H_1, \quad W_1 \approx W_2 H_2, \quad \dots \quad W_{L-1} \approx W_L H_L, \quad (1)$$

where $W_\ell \in \mathbb{R}_+^{m \times r_\ell}$ and $H_\ell \in \mathbb{R}_+^{r_\ell \times r_{\ell-1}}$ for $\ell = 1, 2, \dots, L$, and with $r_0 = n$. This approach yields a total of L layers of decompositions for X :

$$X \approx W_1 H_1, \quad X \approx W_2 H_2 H_1, \quad \dots, \quad X \approx W_L H_L H_{L-1} \dots H_1. \quad (2)$$

For example, let $X \in \mathbb{R}_+^{m \times n}$ represent a hyperspectral image, with m spectral bands and n pixels, where $X(:, j)$ is the spectral signature of the j th pixel and $X(i, :)$ is the vectorized image corresponding to the i th wavelength. Then the first layer of the factorization, $X \approx W_1 H_1$, is such that $W_1 \in \mathbb{R}_+^{m \times r_1}$ contains the spectral signatures of r_1 materials, while $H_1 \in \mathbb{R}_+^{r_1 \times n}$ contains the so-called abundance

*Email: V.Leplat@skoltech.ru. VL acknowledges the support by Ministry of Science and Higher Education grant No. 075-10-2021-068.

[†]Email: {nicolas.gillis,ThiKhanhHien.LE}@umons.ac.be. NG and LTKH acknowledge the support by the European Union (ERC consolidator, eLinoR, no 101085607), by the Fonds de la Recherche Scientifique - FNRS and the Fonds Wetenschappelijk Onderzoek - Vlanderen (FWO) (EOS project no O005318F-RG47), and by the Francqui Foundation.

[‡]Email: ako221@lehigh.edu.

maps that indicate which material is present in which pixel and in which proportion. Figure 1 (a) provides an example where the extracted materials include grass, trees, roof tops, and roads. This is the same interpretation as for NMF [29]. At the second layer, $W_1 \approx W_2 H_2$ so that $W_2 \in \mathbb{R}_+^{m \times r_2}$ contains the spectral signatures of higher-level materials; for example vegetation vs. non-vegetation in Figure 1 (b). In other words, W_2 will merge similar materials in a single material (e.g., grass and trees in the vegetation). The factor $H_2 \in \mathbb{R}^{r_2 \times r_1}$ indicates how the low-level materials are combined into high-level materials.

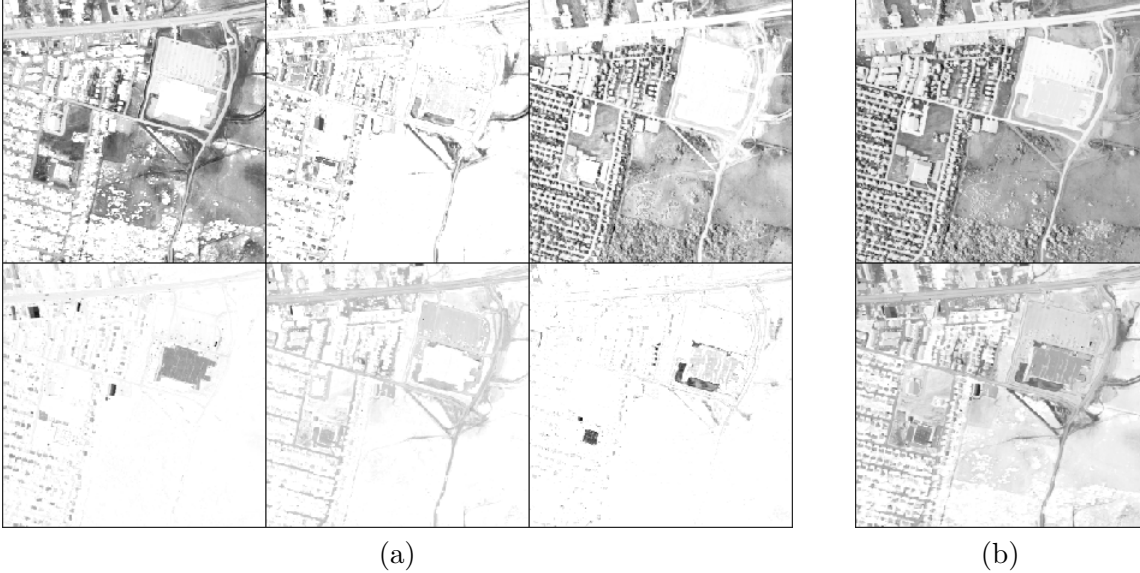


Figure 1: Deep NMF applied on the Urban hyperspectral image, which is an aerial image of a Walmart in Copperas Cove, Texas. We can for example easily identify the roof top and the parking lot of the store; see the fourth and fifth image in (a), respectively. Using Deep NMF with two layers, we obtain the following: (a) Layer 1 with $r_1 = 6$ contains the abundance maps H_1 corresponding to the spectral signatures in W_1 , and (b) Layer 2 with $r_2 = 2$ contains the abundance maps $H_2 H_1$ corresponding to the spectral signatures in W_2 . As the factorization unfolds, deep NMF generates denser abundance maps which are combinations of abundance maps from previous layers. Here, the first level extracts 6 materials (including grass, roof tops and dirt, trees, other roof tops, road and dirt), which are merged into vegetation vs. non-vegetation at the second layer.

We will make the assumption that the ranks decrease as the factorization proceeds, specifically, that $r_{\ell+1} < r_\ell$ for all ℓ . This rank reduction is the most natural and common scenario. It is important to note that employing $r_{\ell+1} > r_\ell$ leads to overparametrization, which can have its merits in certain contexts, such as cases involving implicit regularization, as discussed in [2]. However, our primary objective in this paper is not to pursue overparametrization.

There has been a recent surge of research on deep NMF. It began with the pioneering work of Cichocki and Zdunek [7, 8], which focused on multilayer NMF techniques that sequentially decompose the input matrix X . Subsequently, Trigeorgis et al. [36, 37] introduced deep NMF, presenting a model closely related to the formulation in (2). Deep NMF has found applications across a diverse range of fields, including recommender systems [31], community detection [40], and topic modeling [39]. For more comprehensive insights and surveys on deep matrix factorizations, readers can refer to [10] and [5].

These surveys offer up-to-date overviews of the field and its recent advancements. To the best of our knowledge, it is noteworthy that all existing deep NMF models employ the Frobenius norm, which corresponds to the least squares error, as the standard metric to assess the reconstruction error at each layer. Furthermore, a significant portion of the literature has tended to overlook the modeling aspects inherent to deep NMF. Consequently, many studies have adopted inconsistent models throughout the different layers of the factorization process, as highlighted in [9].

Contribution and outline of the paper In this paper, we first focus on the modeling aspect in Section 2. We explain how to use meaningful regularizations and why a layer-centric loss function is more appropriate than a data-centric one when it comes to identifiability. This was observed experimentally in [9] but not justified from a theoretical viewpoint. Then, in Section 3, we propose new regularized models for deep NMF based on β -divergences, consistent across the layers, and design algorithms for solving the proposed deep NMF models, with a focus on the Kullback-Leibler divergence ($\beta = 1$). As a by-product, for the first time, we will provide Multiplicative Update (MU) for a problem of the type

$$\min_{W \geq 0} D_\beta(X, WH) + \lambda D_\beta(W, \bar{W}),$$

where X , H and \bar{W} are fixed, λ is a positive penalty parameter, and D_β is a β -divergence (see Section 3 for its definition). For the KL divergence, this will require the use of the Lambert \mathcal{W} -function within the MU. Finally, in Section 4, we use the newly proposed models and algorithms for facial feature extraction, topic modeling, and the identification of materials within hyperspectral images.

2 What Deep NMF model to use?

De Hanschutter et al. [9] introduced two distinct loss functions specifically designed for deep NMF:

1. A data-centric loss function (DCLF) defined as

$$\mathcal{L}(H_1, H_2, \dots, H_L; W_1, W_2, \dots, W_L) = \sum_{\ell=1}^L \lambda_\ell D\left(X, W_\ell \prod_{i=0}^{\ell-1} H_{\ell-i}\right), \quad (3)$$

where $D(A, B)$ is a measure of distance between A and B , and the λ_ℓ 's are positive penalty parameters. This loss function minimizes a weighted sum of the errors in the decompositions of X at each of the L levels, as given in (2).

2. A layer-centric loss function (LCLF) defined as

$$\mathcal{L}(H_1, H_2, \dots, H_L; W_1, W_2, \dots, W_L) = \sum_{\ell=1}^L \lambda_\ell D(W_{\ell-1}, W_\ell H_\ell), \quad (4)$$

where $W_0 = X$. This loss function minimizes a weighted sum of the errors at each layer, as given in (1).

In the majority of previous works, the loss function minimized at each layer was not consistent across layers. As a result, their algorithms generally did not converge effectively and often exhibited poorer performance compared to the loss functions introduced in [9].

Layer centric vs. data centric It was empirically observed in [9] that LCLF (as defined in (4)) outperformed DCLF (as defined in (3)) and the state of the art. This superior performance was observed across various synthetic and real datasets, and it extended to the recovery of ground truth factors. Interestingly, the reason behind this performance is not a matter of chance but rather has a theoretical explanation, as indicated in the subsequent sections. This theoretical insight helps provide a deeper understanding of the observed empirical results and offers a rationale for the preference of LCLF in the context of deep NMF.

2.1 Identifiability of NMF

One of the primary reasons why LCLF is found to be more efficient than DCLF lies in the fact that LCLF possesses better identifiability properties compared to DCLF (see Section 2.2). To better understand this distinction, it is beneficial to revisit some of the key NMF identifiability results.

The sufficiently scattered condition A nonnegative matrix $H \in \mathbb{R}_+^{r \times n}$ satisfies the sufficiently scattered condition (SSC)¹ if

$$\mathcal{C} = \{x \in \mathbb{R}_+^r \mid \|x\|_1 \geq q\|x\|_2\} \subseteq \text{cone}(H) = \{x \mid x = Hy, y \geq 0\},$$

for some $q < \sqrt{r-1}$. The set \mathcal{C} is the intersection of the nonnegative orthant with a second-order cone. The SSC implies that H is sufficiently sparse, in particular it requires H to have at least $r-1$ zeros per row; see the discussions in [14] and [16, Chap. 4], and the references therein. Based on the SSC, we have the following identifiability result for NMF.

Theorem 1 ([20]). *Let $X = W^\dagger H^\dagger$ be a rank- r NMF of X , where W^\dagger and H^\dagger satisfy the SSC. Then any other rank- r NMF of X , $X = WH$, corresponds to (W^\dagger, H^\dagger) , up to permutation and scaling of the columns of W^\dagger and rows of H^\dagger .*

Imposing the SSC on both factors, W and H , can sometimes be overly restrictive. For instance, in hyperspectral imaging, it makes sense for H to have this constraint because its rows often represent sparse abundance maps. However, assuming the SSC to W is typically not appropriate since it is expected to be dense in many cases. To address this limitation and provide more flexibility, researchers have introduced regularized NMF models. Among these, the minimum-volume NMF is one of the most effective approaches, both from a theoretical perspective and in practical applications.

Minimum-volume NMF Minimizing the volume of the columns of W is a popular and powerful NMF regularization technique. The most prevalent form of this regularization is achieved by utilizing $\log\det(W^\top W)$ under normalization constraints on either W or H , see e.g., [14] and [16, Chap. 4]. This leads to identifiability/uniqueness of NMF, as stated in Theorem 2. In practice, we use $\log\det(W^\top W + \delta I)$ (with the addition of a small parameter δ) for numerical stability; see the discussion in [23].

Theorem 2 ([15, 13, 24]). *Let $X = W^\dagger H^\dagger$ be a rank- r NMF of X , where $\text{rank}(W^\dagger) = r$ and H^\dagger satisfies the SSC. Then any optimal solution of the following problem*

$$\begin{aligned} \min_W \det(W^\top W) \quad \text{such that } X = WH \quad \text{and} \\ H^\top e = e \text{ [15]} \quad \text{or} \quad He = e \text{ [13]} \quad \text{or} \quad W^\top e = e \text{ [24]}, \end{aligned}$$

corresponds to (W^\dagger, H^\dagger) , up to permutation and scaling of the columns of W^\dagger and rows of H^\dagger .

¹There are several definitions of the SSC, see the discussion in [16, Chap. 4], and we choose here the simplest from [28].

Furthermore, minimizing the volume of the columns of W also has several important implications:

- By encouraging the columns of W to be closer to the data points, this regularization enhances the interpretability of the features represented by these columns.
- The regularization leads to sparser factors H . When the columns of W are close to the data points, it implies that more data points are located near the faces of the cone generated by these columns. Consequently, this results in a sparser representation of the data in the factor H , where many elements are driven towards zero.
- In scenarios where the factorization rank has been overestimated, min-vol NMF can perform automatic rank detection by effectively setting some of the rank-one factors to zero [23].

2.2 Discussion on identifiability of regularized deep NMF

Regularization plays a crucial role in enhancing the interpretability and identifiability of deep NMF, similar to its importance in standard NMF, as illustrated by Theorem 2. When designing a deep NMF model, careful consideration should be given to which factor should be regularized and how it should be done. In the context of LCLF, where W_ℓ is factorized at each layer, it is important to note that overly sparse W_ℓ matrices can be challenging to approximate with NMF. For instance, the identity matrix, which is very sparse, has a unique NMF representation of maximum size ($I = I \cdot I$). Therefore, it makes sense to focus on minimizing the volume of W_ℓ matrices and/or maximizing the sparsity of H_ℓ matrices since this will generate denser W_ℓ matrices.

By the aforementioned reasons, adopting the min-vol NMF approach is a reasonable choice to establish a baseline regularization for deep NMF. In the context of LCLF, at each layer, min-vol LCLF aims to find the solution with the minimum volume for the corresponding W_ℓ . However, applying Theorem 2 to each layer individually is not possible because it would require the W_ℓ matrices to have full rank, which is precluded by construction due to the hierarchical structure where $W_{\ell-1} = W_\ell H_\ell$ and the assumption $r_\ell < r_{\ell-1}$. Fortunately, empirical observations suggest that min-vol NMF can recover W even when it is rank-deficient, provided that H is sufficiently sparse, as demonstrated in [23]. Additionally, the literature includes sparse NMF models, such as those discussed in [1], which offer identifiability even in the rank-deficient case. These observations underscore the adaptability and effectiveness of min-vol regularization in various settings within deep NMF, despite rank-deficiency challenges. We therefore have the following intuition: *when the sparsity of H_ℓ is sufficient, min-vol deep NMF employing the LCLF should exhibit identifiability, provided that each layer, $W_{\ell-1} = W_\ell H_\ell$, is identifiable.*

In the context of DCLF, achieving identifiability necessitates that the products $\prod_{\ell=1}^p H_\ell$ are sufficiently sparse for each layer, where p ranges from 1 to L . However, it is crucial to recognize that the product of sparse nonnegative matrices tends to be denser than the individual factors. Consequently, it becomes significantly less likely for the product of these matrices to exhibit sparsity, which is essential for DCLF to be identifiable. On the other hand, a necessary condition for the product $H = H_1 H_2$ with $H_1 \geq 0$ and $H_2 \geq 0$ to satisfy the SSC is that H_2 satisfies the SSC, because $\text{cone}(H) \subseteq \text{cone}(H_2)$. Remarkably, even when both H_1 and H_2 individually satisfy the SSC, it remains rather unlikely for their product $H_2 H_1$ to satisfy the SSC. In fact, for $H_2 H_1$ to satisfy the SSC, H_1 and H_2 need to be extremely sparse, since $H_2 H_1$ is typically much denser than any of the two. Let us illustrate this observation on a simple example.

Example 1 (The product of matrices satisfying the SSC typically does not satisfy the SSC). Let $r_2 = 3$, $r_1 = 6$, and

$$H_2 = \begin{pmatrix} \omega & 1 & 1 & \omega & 0 & 0 \\ 1 & \omega & 0 & 0 & \omega & 1 \\ 0 & 0 & \omega & 1 & 1 & \omega \end{pmatrix}.$$

It can be shown that H_2 satisfies the SSC if and only if $\omega < 0.5$ [20]. The matrix H_2 is the sparsest non-trivial case for a rank-three matrix that satisfies the SSC, since having columns with two zero entries correspond a stronger condition, referred to as separability, which makes NMF much easier to handle, because columns of W are present in the data set, up to scaling [3]. Now, for any matrix $H_1 \in \mathbb{R}^{6 \times n}$ having 3 nonzeros entries per column, $H_2 H_1$ will be dense, for any $\omega > 0$ (since the sum of any 3 columns of H_2 is dense). Hence $H_2 H_1$ cannot satisfy the SSC for any matrix H_2 with 3-sparse columns, and hence the factorization $X = W_2(H_2 H_1)$ will not be unique. In fact, a necessary condition for the uniqueness of the NMF $X = WH$ is that the supports of the rows of H are not contained in one another [20].

On the other hand, matrices H_1 with three non-zeros per column, and sufficiently many columns, are likely to satisfy the SSC. For example, we have generated randomly 1000 matrices with 100 columns and with 3 non-zeros entries per column, where the position of the non-zero entries are selected at random, and the non-zero entries are generated with the uniform distribution in the interval $[0, 1]$. Among these 1000 sparse matrices, all of them satisfied the SSC².

In summary, the consideration of the product of the H_ℓ matrices within the factorizations in DCLF renders it less likely to achieve identifiability compared to LCLF. This is primarily due to the inherent tendency of these products to become increasingly denser as the factorization unfolds.

3 Deep β -NMF: models and algorithms

In this section, we propose two new deep β -NMF models, describe the algorithms for solving them, and consider the convergence guarantee for the algorithms. The β -divergence between two matrices A and B is defined as follows

$$D_\beta(A, B) = \sum_{i,j} d_\beta(A_{ij}, B_{ij}),$$

where, for scalars x and y ,

$$d_\beta(x, y) = \begin{cases} \frac{x}{y} - \log \frac{x}{y} - 1 & \text{for } \beta = 0, \\ x \log \frac{x}{y} - x + y & \text{for } \beta = 1, \\ \frac{1}{\beta(\beta-1)} (x^\beta + (\beta-1)y^\beta - \beta xy^{\beta-1}) & \text{for } \beta \neq 0, 1. \end{cases} \quad (5)$$

When $\beta = 2$, this corresponds to the least-squares measurement, whereas for $\beta = 1$, it corresponds to the Kullback-Leibler (KL) divergence. With convention that $a \times \log 0 = -\infty$ for $a > 0$ and $0 \times \log 0 = 0$, the KL-divergence is well-defined.

²Although it is NP-hard to check the SSC [20], it is possible to do it for medium-scale matrices by solving a non-convex quadratic optimization problem with Gurobi. We thank Robert Luce, from Gurobi, to help us write down and solve this optimization problem.

3.1 The two proposed deep NMF models

As elucidated in Section 2.2, it is more likely for LCLF to be identifiable compared to DCLF. Hence we employ LCLF as the basis, and introduce the following two novel deep β -NMF model:

1. Deep β -NMF without regularization:

$$\min_{\{W_\ell \geq 0, H_\ell \geq 0\}_{\ell=1}^L} \sum_{\ell=1}^L \lambda_\ell D(W_{\ell-1}, W_\ell H_\ell) \quad \text{subject to } H_\ell e = e \text{ for } \ell = 1, 2, \dots, L, \quad (6)$$

where $W_0 = X$, W_ℓ has r_ℓ columns with $r_0 = n$, and the λ_ℓ 's are positive penalty parameters.

Why the normalization $H_\ell e = e$? Let us explain why we choose the normalization constraints $H_\ell e = e$ in our deep β -NMF model without regularization. The LCLF (4) is not fully consistent because of the scaling degree of freedom in NMF, this was not pointed out in [9]. In fact, except at the first layer, all errors $D(W_{\ell-1}, W_\ell H_\ell)$ for $\ell = 2, 3, \dots, L$ can be made arbitrarily small by using the scaling degree of freedom: multiply W_1 by an arbitrarily small positive constant and divide H_1 by the same constant. This does not change $D(X, W_1 H_1)$ while W_1 is arbitrarily close to zero making $D(W_1, W_2 H_2)$ arbitrarily small (for any norm which is not scaled invariant, e.g., all β -divergences for $\beta > 0$). Therefore, for (3) to make sense, it is crucial to add a normalization constraints on the W_ℓ 's or the H_ℓ 's. Many options are possible, and depend on the application at hand. For example, in hyperspectral imaging, we might impose $H_\ell^\top e = e$ for all ℓ which is known as the sum-to-one constraint of the abundances, and, in topic modeling, where the columns of W_ℓ correspond to topics, we might impose $W_\ell^\top e = e$ as the entries in each column of W_ℓ correspond to probabilities of the words to belong to the corresponding topic. In this paper, we choose $H_\ell e = e$, that is, the sum of the entries in each row of H_ℓ sums to one, as in [13]. The main reason is that this normalization can be made w.l.o.g. by the scaling degree of freedom. Moreover, we do not constraint W_ℓ because it would make the design of closed-form MU for W_ℓ much more difficult, if possible (see Section 3.2.2 for such a case).

2. Minimum-volume deep β -NMF:

$$\min_{W_\ell \geq 0, H_\ell \geq 0} \sum_{\ell=1}^L \lambda_\ell D_\beta(W_{\ell-1}, W_\ell H_\ell) + \alpha_\ell \log \det (W_\ell^\top W_\ell + \delta I) \quad \text{such that } W_\ell^\top e = e, \quad (7)$$

where $W_0 = X$, W_ℓ has r_ℓ columns with $r_0 = n$, δ is small positive scalar that prevents the log det to go to $-\infty$, and the α_ℓ 's are positive penalty parameters. The choice to normalize the columns of W_ℓ , rather than the rows of H_ℓ , is from the fact that it significantly improves the conditioning of the min-vol NMF problem. When rows of H_ℓ are normalized, it can lead to highly ill-conditioned of W_ℓ , especially when certain columns of W_ℓ exhibit substantially larger norms compared to others. This issue is further elaborated in discussions and examples provided in [24] and [16, Chapter 4.3.3.5].

3.2 Algorithms for solving the proposed deep β -NMF models

Obtaining a global solution for deep NMF is a computationally challenging problem, as it generalizes NMF which is NP-hard [38]. Moreover, the objective functions presented in (6) and (7) are jointly non-convex for the variable $(W_1, \dots, W_L, H_1, \dots, H_L)$. Consequently, updating all factors simultaneously

can be prohibitively expensive. Therefore, most algorithms designed to address (deep) NMF rely on block coordinate methods. These methods update one factor at a time while keeping the others fixed. In this paper, we also adopt this strategy to efficiently address the optimization challenges associated with deep NMF. Specifically, we will employ the block majorization minimization (BMM) framework, which was designed to solve the following multi-block nonconvex optimization problem:

$$\min_{x:=(x_1, x_2, \dots, x_s) \in \mathcal{X}} F(x) := f(x) + \sum_{i=1}^s g_i(x_i), \quad (8)$$

where f is continuous on \mathcal{X} , g_i 's are proper and lower-semicontinuous functions (possibly with extended values), and $\mathcal{X} = \mathcal{X}_1 \times \mathcal{X}_2 \times \dots \times \mathcal{X}_s$ with \mathcal{X}_i ($i = 1, 2, \dots, s$) being closed convex sets. At iteration k , BMM fixes the latest values of block $j \neq i$ and updates block x_i by

$$x_i^k \in \arg \min_{x_i \in \mathcal{X}_i} \left\{ u_i(x_i, x_1^k, \dots, x_{i-1}^k, x_i^{k-1}, x_{i+1}^{k-1}, \dots, x_s^{k-1}) + g_i(x_i) \right\}, \quad (9)$$

where $u_i : \mathcal{X}_i \times \mathcal{X} \rightarrow \mathbb{R}$ is a block majorizer of $f(x)$, that is, u_i satisfies the following conditions

$$\begin{aligned} u_i(x_i, x) &= f(x), \forall x \in \mathcal{X}, \\ u_i(y_i, x) &\geq f(x_1, \dots, x_{i-1}, y_i, x_{i+1}, \dots, x_s), \forall y_i \in \mathcal{X}_i, x \in \mathcal{X}. \end{aligned} \quad (10)$$

From the definition of u_i , we have

$$\begin{aligned} F(x^k) &= u_1(x_1^k, x_1^k, \dots, x_s^k) + \sum_i g_i(x_i^k) \geq u_1(x_1^{k+1}, x_1^k, \dots, x_s^k) + g_1(x_1^{k+1}) + \sum_{i \neq 1} g_i(x_i^k) \\ &\geq f(x_1^{k+1}, x_2^k, \dots, x_s^k) + g_1(x_1^{k+1}) + \sum_{i \neq 1} g_i(x_i^k) \\ &= u_2(x_2^k, x_1^{k+1}, x_2^k, \dots, x_s^k) + g_1(x_1^{k+1}) + \sum_{i \neq 1} g_i(x_i^k) \\ &\geq u_2(x_2^{k+1}, x_1^{k+1}, x_2^k, \dots, x_s^k) + g_1(x_1^{k+1}) + g_2(x_2^{k+1}) + \sum_{i \notin \{1,2\}} g_i(x_i^k) \\ &\geq f(x_1^{k+1}, x_2^{k+1}, \dots, x_{s-1}^k, x_s^k) + g_1(x_1^{k+1}) + g_2(x_2^{k+1}) + \sum_{i \notin \{1,2\}} g_i(x_i^k) \\ &\geq \dots \geq F(x^{k+1}). \end{aligned}$$

In other words, BMM produces a non-increasing sequence $\{F(x^k)\}$. We refer the readers to [18, 30, 33, 34] for examples of majorizer functions. BMM was introduced in [33] (with the name BSUM - block successive upper-bound minimization) to solve nonconvex problem (8) with $g_i = 0$ for $i = 1, \dots, s$. It is proved in [33, Theorem 2] that if the following conditions are satisfied then we have a convergence guarantee for BSUM:

- $u_i(y_i, x)$ is quasi-convex in y_i for $i = 1, \dots, s$,
- the subproblem (9) with $g_i = 0$ has a unique solution,
- $u'_i(y_i, x; d_i)|_{y_i=x_i} = f'(x; d) \quad \forall d = (0, \dots, d_i, \dots, 0) \text{ s.t. } x_i + d_i \in \mathcal{X}_i \quad \forall i$,
- $u_i(y_i, x)$ is continuous in (y_i, x) , for all i .

It is worth mentioning that the authors in [18] introduced TITAN, an accelerated version of BMM for solving Problem (8). TITAN enhances BMM method by including an inertial term in each block update, which significantly boosts the convergence of BMM. Leveraging the convergence outcomes established for BSUM and TITAN, numerous algorithms addressing low-rank factorization problems come with guaranteed convergence. For example, BSUM assures the convergence of a perturbed Multiplicative Update (MU) method and a block mirror descent method for KL NMF, as described in [17]; TITAN provides convergence guarantees for accelerated algorithms dealing with minimum-volume NMF [35], sparse NMF and matrix completion [18].

Although convergence guarantees have been firmly established for BSUM (and TITAN) under appropriate assumptions, which serve as valuable tools to ensure the convergence of BMM in solving deep NMF models, it is not a straightforward task to construct suitable majorizers that satisfy the required assumptions. In the next sections, we propose suitable majorizers and apply BSUM to design efficient algorithms to solve the two proposed deep β -NMF models. To that end, we need the following lemma from [12] that provides a majorizer for $h \mapsto \sum_i d_\beta(v_i, [Wh]_i)$, where vector v and matrix W are fixed.

Lemma 1 (Majorizer function for β -NMF [12]). *Denote Wh by \tilde{v} and the entries $[Wh]_i$ by \tilde{v}_i . Let \tilde{h} be such that $\tilde{v}_i > 0$ and $\tilde{h}_i > 0$. Then the following function*

$$G(h, \tilde{h}) = \sum_i \left[\sum_j \frac{w_{ij} \tilde{h}_j}{\tilde{v}_j} \check{d}(v_i, \tilde{v}_j \frac{h_j}{\tilde{h}_j}) \right] + \left[\check{d}(v_i, \tilde{v}_i) \sum_j w_{ij} (h_j - \tilde{h}_j) + \hat{d}(v_i, \tilde{v}_i) \right] + \bar{d}(v_i) \quad (11)$$

is a majorizer of the function $h \mapsto \sum_i d_\beta(v_i, [Wh]_i)$, where \check{d} is a convex function of u , \hat{d} is a concave function of u and \bar{d} is a constant of u in the following decomposition of d_β

$$d_\beta(v, u) = \check{d}(v, u) + \hat{d}(v, u) + \bar{d}(v), \quad (12)$$

see Table 1.

Table 1: Differentiable convex-concave-constant decomposition of the β -divergence under the form (12) [12].

	$\check{d}(v, u)$	$\hat{d}(v, u)$	$\bar{d}(v)$
$\beta = 0$	vu^{-1}	$\log(u)$	$u(\log(v) - 1)$
$\beta \in [1, 2]$	$d_\beta(v, u)$	0	0

Note 1. It is important to note that $G(h, \tilde{h})$ is convex in h . Furthermore, since $D_\beta(y, wH) = D_\beta(y^\top, H^\top w^\top)$, where vector y^\top and matrix H are fixed, Lemma 1 can be used to derive a similar majorizer $G(w, \tilde{w})$ for $w \mapsto D_\beta(y, wH)$. On the other hand, note that $D_\beta(Y, WH) = \sum_i D_\beta(y_i, w_i H)$, where y_i and w_i are the i -th row of Y and W respectively. This means $W \mapsto D_\beta(Y, WH)$ is separable with respect to the rows of W . Hence, we can formulate a majorizer for $W \mapsto D_\beta(Y, WH)$ by summing up the majorizers of its rows $w \mapsto D_\beta(y, wH)$. We have similar procedure for $H \mapsto D_\beta(Y, WH)$. Considering KL NMF, BSUM using the majorizers in Lemma 1 is the MU algorithm proposed in [21, 22], see [17].

3.2.1 Algorithm for solving deep β -NMF without regularization

Problem (6) has the form of (8), where x comprises W_l and H_l for $l = 1, \dots, L$, $g_i = 0$, the closed convex set \mathcal{X}_i that corresponds to W_l is $\mathbb{R}_+^{m \times r_l}$ and that corresponds to H_l is $\{H_l : H_l \in \mathbb{R}_+^{r_l \times r_{l-1}}, H_l e = e\}$.

Update of H_l We observe that H_l only appears in one term of the objective function in (6). While fixing the other factors, minimizing the objective of Problem (6) with respect to H_l is the same as in standard β -NMF. We hence employ the majorizers in Lemma 1 (see also Note 1) and the recently introduced framework in [25] that allows one to derive MU for block of variables satisfying disjoint equality constraints as well as nonnegativity constraints. See Algorithm 1 for the actual update. Note that, in the update of H_ℓ , the parameter μ appearing in the denominator correspond to the optimal vector of Lagrange multipliers allowing the new updates to satisfy both the nonnegativity and the sum-to-one constraints, see [25] for more details about the procedure for such updates.

Update of W_l for $l = 1, \dots, L - 1$. While fixing the other factors, the corresponding subproblems of (6) with respect to each of the block W_ℓ , $l = 1, \dots, L - 1$, have the same structure. Hence we can focus on building the majorizer for one representative W_ℓ . To simplify the notation, let us denote W_ℓ by W , $W_{\ell-1}$ by Y , H_ℓ by H , and $W_{\ell+1}H_{\ell+1}$ by \bar{W} . Then each subproblem is equivalent to the following problem (after removing the constants in the objective):

$$\text{find } \underset{W \geq 0}{\operatorname{argmin}} \lambda_l D_\beta(Y, WH) + \lambda_{l+1} D_\beta(W, \bar{W}), \quad (13)$$

where Y , H , \bar{W} are given and kept fixed during the update of W . Let w_i , y_i , and \bar{w}_i , $i = 1, \dots, m$, be the rows of W , Y and \bar{W} respectively. Note that the objective function of (13) equals to

$$\lambda_l \sum_i^m D_\beta(y_i, w_i H) + \lambda_{l+1} \sum_i^m D_\beta(w_i, \bar{w}_i),$$

which is separable with respect to the rows of W . Therefore, we can focus on building the majorizer for each row of W (then sum up these majorizers to formulate the majorizer for W). Problem (13) restricted to a particular row i of W is equivalent to the following problem:

$$\text{find } \underset{w \geq 0}{\operatorname{argmin}} \lambda_\ell D_\beta(y, wH) + \lambda_{\ell+1} D_\beta(w, \bar{w}), \quad (14)$$

where the subscript i has been dropped for notation succinctness. For the first term in the objective of (14), we use the majorizer $G(w, \tilde{w})$ proposed in [12], see Note 1. This implies that $\lambda_\ell G(w, \tilde{w}) + \lambda_{\ell+1} D_\beta(w, \bar{w})$ is a majorizer of $w \mapsto \lambda_\ell D_\beta(y, wH) + \lambda_{\ell+1} D_\beta(w, \bar{w})$. Consequently, the update of each row of W is

$$w \in \underset{w \geq 0}{\operatorname{argmin}} G(w, \tilde{w}) + \lambda D_\beta(w, \bar{w}), \quad (15)$$

where $\lambda = \frac{\lambda_{\ell+1}}{\lambda_\ell}$. To compute the positive minimizer of (15), it's sufficient to look for $w \in \mathbb{R}_+^{r_\ell}$ that cancels the gradient of objective function from (15). Since the objective function is separable w.r.t. each entry w_{k_ℓ} , we focus on solving:

$$\text{find } \hat{w}_{k_\ell} \geq 0 \text{ such that } \nabla_{w_{k_\ell}} [G(w|\tilde{w}) + \lambda D_\beta(w|\bar{w})]_{w=\hat{w}_{k_\ell}} = 0. \quad (16)$$

The next steps depend on the value chosen for β . A closed-form of the minimizer \hat{w} can be derived for $\beta \in \{0, 1/2, 1, 3/2, 2\}$. In the following, we detail the updates for the case $\beta = 1$, the updates obtained for $\beta \in \{0, 1/2, 3/2\}$ can be found similarly. Note that considering the case $\beta = 2$ is excluded since the objective function is L -smooth in this case, hence efficient first-order methods can be used to tackle the subproblems such as the well-known Nesterov accelerated projected gradient descent [32]. Furthermore, for $\beta \in \{0, 1, 3/2\}$, $D_\beta(w, \tilde{w})$ is strictly convex in w , which makes the majorizer $\lambda_\ell G(w, \tilde{w}) + \lambda_{\ell+1} D_\beta(w, \tilde{w})$ strictly convex in w (this convexity is used to verify the conditions for a convergence guarantee of BSUM).

Considering $\beta = 1$, from Lemma 1 and Table 1 for $\beta = 1$, one can show that solving (16) boils down to solve the following scalar equation in w_{k_ℓ} :

$$a = \frac{b}{w_{k_\ell}} - \lambda \log(w_{k_\ell}), \quad (17)$$

where $a = \sum_{k_{\ell-1}}^{r_{\ell-1}} H_{k_\ell k_{\ell-1}} - \lambda \log(\tilde{w}_{k_\ell})$ and $b = \tilde{w}_{k_\ell} \sum_{k_{\ell-1}}^{r_{\ell-1}} H_{k_\ell k_{\ell-1}} \frac{x_{k_{\ell-1}}}{[\tilde{w}H]_{k_{\ell-1}}}$. Equation (17) has the following nonnegative solution:

$$\hat{w}_{k_\ell} = \frac{b}{\lambda \mathcal{W}\left(\frac{be^{\frac{a}{\lambda}}}{\lambda}\right)}, \quad (18)$$

where $\mathcal{W}(\cdot)$ denotes the Lambert \mathcal{W} -Function. Interestingly, this update is well defined at the boundaries of the feasible set, in particular when b and a respectively tend to 0 and $+\infty$, the latter occurs when the entry \tilde{w}_{k_ℓ} tends to 0. Indeed, we have $\lim_{\substack{b \rightarrow 0, b > 0 \\ a < +\infty}} \hat{w}_{k_\ell}(b) = e^{-\frac{a}{\lambda}}$, and $\lim_{\substack{a \rightarrow +\infty \\ b \neq 0}} \hat{w}_{k_\ell}(a) = 0$. Equation

(18) can be expressed in matrix form as follows:

$$\hat{W} = \frac{[B]}{\left[\lambda \mathcal{W}\left(\frac{B \odot e^{\frac{A}{\lambda}}}{\lambda}\right) \right]}, \quad (19)$$

where $A = JH_\ell^\top - \log(\tilde{W})$ with J is a all-one matrix of size m -by- $r_{\ell-1}$, and \log is element wise, and the notation e^A is the element-wise exponential, and $B = \tilde{W} \odot \left(\frac{[X]}{[\tilde{W}H]} H^\top \right)$ with $C \odot D$ (resp. $\frac{[C]}{[D]}$) is the Hadamard product (resp. division) between C and D and $C^{(\cdot, \alpha)}$ is the element-wise α exponent of C .

Update of W_L We observe that W_L only appears in the last term of the objective in (6). Minimizing the objective of Problem (6) with respect to W_L is the same as in standard β -NMF. As noted in Note 1, BSUM step collapses to the classical MU of β -NMF.

Algorithm 1 summarizes our proposed algorithm for deep β -NMF in the case $\beta = 1$, that is, for deep KL-NMF, recalling that $W_0 = X$ and W_ℓ has r_ℓ columns with $r_0 = n$.

Convergence guarantee Although Algorithm 1 produces a non-increasing sequence $\{f^k\}$, where f^k is the objective of (6) at iteration k (since its update follows BSUM), it is not guaranteed that the generated sequence of Algorithm 1 converges (the objective in (6) is not directionally differentiable). To have some convergence for Algorithm 1, we need to impose the constraints $W_\ell \geq \varepsilon$ and $H_\ell \geq \varepsilon$, where $\varepsilon > 0$, to (6). In our implementation, we choose the MATLAB machine epsilon for ε . Using the same majorizers, we obtain a perturbed version of Algorithm 1, in which we take the element-wise

Algorithm 1 Algorithm for Deep KL-NMF

Input: Input data matrix X , number of layers L , inner ranks r_l 's, weight parameters λ_ℓ 's.

Output: An approximate solution to Problem 6

```
1: for  $k = 1, \dots$  do
2:   for  $l \in \{1, \dots, L\}$  do
3:     % Update of factors  $H_\ell$  using [25]
4:      $H_\ell^k \leftarrow H_\ell^{k-1} \odot \frac{\left[ W_\ell^{T,k-1} \left( \frac{[W_{\ell-1}^k]}{[W_\ell^{k-1} H_\ell^{k-1}]} \right) \right]}{\left[ W_\ell^{T,k-1} e_{m \times r_{\ell-1}} - \mu e_{1 \times r_{\ell-1}} \right]}$ , where  $\mu$  is the root of a univariate polynomial.
5:     % Update of factors  $W_\ell$ 
6:     if  $l < L$  then
7:        $\lambda \leftarrow \frac{\lambda_{\ell+1}}{\lambda_\ell}$ 
8:        $B \leftarrow W_\ell^{k-1} \odot \left( \frac{[W_{\ell-1}^k]}{[W_\ell^{k-1} H_\ell^k]} H_\ell^{T,k} \right)$ 
9:        $A \leftarrow J H_\ell^{T,k} - \log(W_{\ell+1}^{k-1} H_{\ell+1}^{k-1})$ 
10:       $W_\ell^k \leftarrow \frac{[B]}{\left[ \lambda W \left( \frac{B \odot e}{\lambda} \right) \right]}$ 
11:    else if  $l = L$  then
12:       $W_\ell^k \leftarrow W_L^{k-1} \odot \frac{\left( \frac{[W_{L-1}^k]}{[W_L^{k-1} H_L^k]} \right) H_L^{T,k}}{\left[ e_{m \times r_L} H_L^{T,k} \right]}$ , which are the standard MU for KL-NMF.
13:    end if
14:  end for
15: end for
16: return  $\{W_L, H_L, H_{L-1}, \dots, H_1\}$ 
```

maximum between the updates of factors $\{W_\ell, H_\ell\}_{\ell=1}^L$ (corresponding to the closed-form expression of the minimizer of BMM step built at the current iterate) and ε . With this additional constraints, the sufficient conditions for convergence of BSUM would be satisfied, leading to the convergence of the perturbed version of Algorithm 1.

It is worth noting that for the case $\beta \in \{0, 3/2\}$, we also have closed-form solution for (16) and strict convexity for the majorizer used in (15). Hence, a similar rationale can be applied to have a convergence guarantee for a perturbed version of the algorithm solving the proposed deep β -NMF, with $\beta \in \{0, 3/2\}$.

Parallelization The proposed algorithm relies on MU-based approaches, which involve computationally intensive steps in matrix products. Furthermore, the updates to all factor entries can be carried out independently. Therefore, the proposed algorithm can be effectively executed on a parallel and high performance computing platform.

Remark 1 (Usefulness of our MU in other contexts). *The MU (19) allows us to update W to minimize (13). The regularized NMF problem (13), in which \bar{W} is a given matrix to which W should be close to, could be useful in other contexts than deep β -NMF, e.g., if some \bar{W} is available via prior some knowledge, by taking $\bar{W} = 0$ to regularize W , or for symmetric NMF where $\bar{W} = H^\top$ [26].*

3.2.2 Algorithm for solving minimum-volume deep β -NMF

Problem (7) has the form of (8) in which x comprises W_l and H_l for $l = 1, \dots, L$, $g_i = 0$, the closed convex set \mathcal{X}_i that corresponds to W_ℓ is $\{W_\ell : W_\ell \in \mathbb{R}_+^{m \times r_{\ell-1}}, W_\ell^\top e = e\}$ and that corresponds to H_ℓ is $\mathbb{R}_+^{r_\ell \times r_{\ell-1}}$. Let us focus on the special case $\beta = 1$. Similar rationale can be followed for other values of the β parameter.

Update of H_ℓ Each factor H_ℓ with $1 \leq \ell \leq L$ only appears in a single term within the objective function of Problem (7). As a result, one can directly apply the classical MU updates for H_ℓ , which is a BSUM step that minimizes the convex majorizers in Lemma 1 (see also Note 1) over $\mathbb{R}_+^{r_\ell \times r_{\ell-1}}$.

Update of W_ℓ , $\ell = 1, \dots, L-1$ Since all the subproblems in W_ℓ with $1 \leq \ell < L$ have the same structure, we detail in the following the methodology for deriving the BSUM update for one specific W_ℓ . For notation succinctness, we drop the subscript ℓ and denote W_ℓ by W , $W_{\ell-1}$ by Y , H_ℓ by H , and $W_{\ell+1}H_{\ell+1}$ by \bar{W} . The subproblem in W is as follows:

$$\begin{aligned} \text{find } \arg \min_W \quad & \lambda_\ell D(Y, WH) + \lambda_{\ell+1} D(W, \bar{W}) + \alpha_\ell \log \det (W^\top W + \delta I) \\ \text{such that } & W^\top e = e, W \geq 0. \end{aligned}$$

For a given matrix \tilde{W} , denote $A = (\tilde{W}^\top \tilde{W} + \delta I)^{-1}$, $A^+ = \max(A, 0) \geq 0$, $A^- = \max(-A, 0) \geq 0$, and $\Phi(\tilde{w}_i) = \text{Diag} \left(2 \frac{[A^+ \tilde{w}_i + A^- \tilde{w}_i]}{[\tilde{w}]}\right)$ (which is a diagonal matrix), where \tilde{w}_i is a row of \tilde{W} , and $\frac{[\cdot]}{[\cdot]}$ denotes the component-wise division. Let $l(w) = w^\top A w$ and $\Delta w_i = w_i - \tilde{w}_i$, where w_i is a row of W . From [24, Lemma 3], we have

$$\begin{aligned} \log \det (W^\top W + \delta I) &\leq \log \det (\tilde{W}^\top \tilde{W} + \delta I) + \langle A, W^\top W - \tilde{W}^\top \tilde{W} \rangle \\ &\leq g(W, \tilde{W}) := \log \det (\tilde{W}^\top \tilde{W} + \delta I) - \langle A, \tilde{W}^\top \tilde{W} \rangle + \sum_i l(\tilde{w}_i) + \langle \Delta w_i, \nabla l(\tilde{w}_i) \rangle + \frac{1}{2} \langle \Delta w_i, \Phi(\tilde{w}_i) \Delta w_i \rangle \end{aligned} \quad (20)$$

We then use the following majorizer for W :

$$\bar{G}(W, \tilde{W}) = \lambda_\ell G(W, \tilde{W}) + \lambda_{\ell+1} D(W, \bar{W}) + \alpha_\ell g(W, \tilde{W}),$$

where $G(W, \tilde{W})$ is the majorizer of $W \mapsto D(Y, WH)$, which is formed by summing the majorizers proposed in Lemma 1 for each row of W as discussed in Note 1. As $G(W, \tilde{W})$ and $D(W, \bar{W})$ are convex in W and $g(W, \tilde{W})$ is strongly convex in W over $\mathbb{R}_+^{m \times r_{\ell-1}}$, we also have $\bar{G}(W, \tilde{W})$ is strongly convex in W . Hence, the following subproblem of BSUM for the update of W has the unique solution:

$$\begin{aligned} \text{find } \arg \min_W \quad & \lambda_\ell G(W, \tilde{W}) + \lambda_{\ell+1} D(W, \bar{W}) + \alpha_\ell g(W, \tilde{W}) \\ \text{such that } & W^\top e = e, W \geq 0, \end{aligned} \quad (21)$$

where \tilde{W} is the current iterate. Below we will describe an ADMM to solve Problem (21).

Update of W_L We observe that W_L only appears in the last term of the objective. We use the following majorizer for W_L

$$\bar{G}_L(W_L, \tilde{W}_L) = \lambda_L G_L(W_L, \tilde{W}_L) + \alpha_L g(W_L, \tilde{W}_L),$$

where $G_L(W_L, \tilde{W}_L)$ is the majorizer of $W_L \mapsto D(W_{L-1}, W_L H_L)$, which is derived as in Note 1, and g is defined in (20). Then BSUM step collapses to the MU update proposed in [25].

ADMM for solving Problem (21) to update W_ℓ We rewrite Problem (21) as follows:

$$\begin{aligned} \text{find } & \arg \min_{W, Z} G(W, \tilde{W}) + \lambda D(Z, \bar{W}) + \alpha g(W, \tilde{W}) + \mathcal{I}_{\mathcal{X}_W}(W) \\ \text{s.t. } & W - Z = 0, \end{aligned} \quad (22)$$

where $\mathcal{I}_{\mathcal{X}_W}(W)$ denotes the indicator function associated to the convex set $\mathcal{X}_W =: \{W \in \mathbb{R}^{m \times r_{l-1}} | W^\top e = e, W \geq 0\}$, $\alpha = \frac{\alpha_\ell}{\lambda_\ell}$, and $\lambda = \frac{\lambda_{\ell+1}}{\lambda_\ell}$. The augmented Lagrangian function associated to Problem (22) is

$$L_\rho(W, Z, U) := G(W, \tilde{W}) + \lambda D(Z, \bar{W}) + \alpha g(W, \tilde{W}) + \mathcal{I}_{\mathcal{X}_W}(W) + \frac{\rho}{2} \|W - Z + U\|_F^2, \quad (23)$$

where U denotes the *scaled* dual variables associated to constraints $W - Z = 0$, written in the so-called scaled form. Here-under, we detail the iterative procedure to compute the solution (W, Z) for Problem (22). As for classical ADMM-based methods, each iteration of our procedure performs three steps. Given the current iterates (W^i, Z^i, U^i) , the three steps are:

1. W -minimization:

$$W^{i+1} := \arg \min_{W \in \mathcal{X}_W} \left\{ G(W, \tilde{W}) + \alpha g(W, \tilde{W}) + \frac{\rho}{2} \|W - Z^i + U^i\|_F^2 \right\}, \quad (24)$$

where \tilde{W} is the current iterate of the main algorithm.

2. Z -minimization:

$$Z^{i+1} := \arg \min_Z \left\{ \lambda D(Z, \bar{W}) + \frac{\rho}{2} \|W^{i+1} + U^i - Z\|_F^2 \right\}. \quad (25)$$

3. Dual Updates:

$$U^{i+1} := U^i + W^{i+1} - Z^{i+1}. \quad (26)$$

It remains to present how we tackle Problems (24) and (25).

W-minimization The update of W is computed based on the methodology recently introduced in [25]. We obtain the following updates in matrix form:

$$\hat{W}(\mu) = \tilde{W} \odot \frac{\left[[C + e\mu^\top] \cdot^2 + S \right]^{\frac{1}{2}} - (C + e\mu^\top)}{[T]}, \quad (27)$$

where $A = (\tilde{W}^\top \tilde{W} + \delta I)^{-1}$, $A^+ = \max(A, 0) \geq 0$, $A^- = \max(-A, 0) \geq 0$, $C = e_{m, r_\ell-1} H^\top - 4\alpha(\tilde{W} A^-) - \rho(Z^i - U^i)$, $T = 4\alpha\tilde{W}(A^+ + A^-) + 2\rho e_{m, r_\ell}$, $S = (8\alpha\tilde{W}(A^+ + A^-) + 4\rho e_{m, r_\ell}) \odot \left(\frac{[Y]}{[WH]} H^\top \right)$, and $e_{m, r_\ell-1}$ and e_{m, r_ℓ} are respectively m -by- $r_\ell - 1$ and m -by- r_ℓ matrices of all ones, and $\mu \in \mathbb{R}^{r_\ell}$ denotes the vector of Lagrange multipliers associated to equality constraints on the columns of W , see [25] for further details. One can easily observe that updates defined in Equation (27) satisfy the nonnegativity constraints, given $\tilde{W} \geq 0$. Moreover, as per [25, Proposition 2], the constraints $\hat{W}(\mu)^\top e = e$ are satisfied for a unique $\mu^* \in \mathbb{R}^{r_\ell}$. The computation of μ^* is achieved by using a Newton-Raphson procedure for solving $\hat{W}(\mu)^\top e = e$. The update W^{i+1} for Problem (24) is finally performed using Equation (27) with $\mu = \mu^*$.

Z-minimization For this step, one can observe that the objective function to minimize is separable w.r.t. each entry of factor Z , hence each entry can be optimized independently. The updates are obtained by computing \hat{Z} which satisfy the first-order optimality conditions, a.k.a the KKT conditions. For $\beta = 1$, and denoting by z the (o, p) -th entry of Z of interest, we are looking for \hat{z} such that

$$\nabla_z \{ \lambda d_{KL}(z, \bar{W}_{o,p}) + \frac{\rho}{2} \|z - V_{o,p}^i\|_2^2 \}_{z=\hat{z}} = 0.$$

It boils to solve the following scalar equation in z :

$$\log(z) + b + \nu z = 0, \quad (28)$$

where $b = -\log(\bar{W}_{o,p}) - \nu V_{o,p}^i$ and $\nu = \frac{\rho}{\lambda}$. Equation (28) can be solved in closed form as follows:

$$\hat{z} = \frac{\mathcal{W}(e^{-b}\nu)}{\nu}, \quad (29)$$

where $\mathcal{W}(\cdot)$ denotes the Lambert \mathcal{W} -function. Equation (29) can be easily expressed in matrix form; the update Z^{i+1} for Problem (25) finally writes:

$$Z^{i+1} = \frac{[\mathcal{W}(e^{-B}\nu)]}{\nu} \quad (30)$$

where $B = -\log(\bar{W}) - \nu V^i$; the $\log(\cdot)$ being applied element-wise to the matrix \bar{W} .

Finally, the three steps of the ADMM-like procedure detailed above for solving Problem (21) are repeated either for a maximum number iteration i_{max} or until the stopping criterion $\|W^i - Z^i\|_F \leq \epsilon$ is reached, with ϵ a threshold defined a priori.

Algorithm 2 summarizes our proposed algorithm for minimum-volume deep β -NMF in the case $\beta = 1$, that is, for minimum-volume deep KL-NMF (recall that $W_0 = X$ and W_ℓ has r_ℓ columns with $r_0 = n$).

Convergence guarantees In fact, similar to the deep β -NMF without regularization, if we enforce the constraints $W_\ell \geq \varepsilon$ and $H_\ell \geq \varepsilon$ onto (7), given that each block update (9) of BSUM admits a closed-form solution, BSUM would indeed possess a convergence guarantee for the cases $\beta = 0$ or $\beta \geq 1$ (as the majorizer used in (21) would be strongly convex for these cases). However, it is essential to emphasize that we employ ADMM to address (21) during the update of W_ℓ . Since the update of W_ℓ is inexact, Algorithm 2 can be viewed as an inexact version of BSUM.

4 Numerical Experiments

In this section, we report the use of deep KL-NMF for three applications: facial feature extraction, topic modeling, and hyperspectral unmixing. The codes are written in MATLAB R2021a and available from <https://github.com/vleplat/deep-KL-NMF-public>, and can be used to reproduce all experiments described below.

It is important to acknowledge the challenge of quantitatively assessing the performance of our novel deep KL-NMF models. First, we are the first to introduce deep NMF models based on the KL divergence. Second, the absence of ground-truth real-world data for deep NMF models further complicates performance evaluation, as pointed out in the survey paper [10]. A promising avenue for future research lies in the development of datasets specifically designed for deep NMF models. These datasets would facilitate more robust and empirical evaluations, addressing this critical need in the field.

Algorithm 2 Algorithm for Minimum-Volume Deep KL-NMF

Input: Input data matrix X , number of layers L , inner ranks r_l 's, weight parameters λ_ℓ 's and α_ℓ 's, a scalar $\delta > 0$, a maximum of iteration i_{max} , and a threshold ϵ , and parameter $\rho > 0$ for ADMM procedure.

Output: An approximate solution to Problem 7

```

1: for  $k = 1, \dots$  do
2:   for  $l \in \{1, \dots, L\}$  do
3:     % Update of factors  $H_\ell$ 
4:      $H_\ell^k \leftarrow H_\ell^{k-1} \odot \frac{W_\ell^{T, k-1} \left( \frac{[W_{\ell-1}^k]}{[W_\ell^{k-1} H_\ell^{k-1}]} \right)}{[W_\ell^{T, k-1} e_{m \times r_{\ell-1}}]}$ 
5:     % Update of factors  $W_\ell$ 
6:     if  $l < L$  then
7:       % ADMM-procedure
8:        $\tilde{W} \leftarrow W_\ell^{k-1}$ ,  $A \leftarrow (\tilde{W}^\top \tilde{W} + \delta I)^{-1}$ ,  $A^+ \leftarrow \max(A, 0)$ ,  $A^- \leftarrow \max(-A, 0)$ ,
9:        $T \leftarrow 4 \frac{\alpha_\ell}{\lambda_\ell} \tilde{W} (A^+ + A^-) + 2\rho e_{m, r_\ell}$ ,  $S \leftarrow (8 \frac{\alpha_\ell}{\lambda_\ell} \tilde{W} (A^+ + A^-) + 4\rho e_{m, r_\ell}) \odot \left( \frac{[W_{\ell-1}^k]}{[\tilde{W} H_\ell^k]} (H_\ell^k)^\top \right)$ 
10:       $i \leftarrow 1$ 
11:       $W^0 \leftarrow W_\ell^{k-1}$ ,  $Z^0 \leftarrow W^0$ ,  $U^0 \leftarrow 0$ 
12:       $\nu \leftarrow \frac{\rho}{\lambda_{\ell+1}}$ 
13:      while  $i \leq i_{max}$  &  $\|W^i - Z^i\|_F \leq \epsilon$  do
14:        % W-minimization
15:         $V^i \leftarrow Z^i - U^i$ 
16:         $C \leftarrow e_{m, r_{\ell-1}} (H_\ell^k)^\top - 4 \frac{\alpha_\ell}{\lambda_\ell} (\tilde{W} A^-) - \rho V^i$ 
17:         $\mu \in \text{root}(\tilde{W}(\mu)^\top e = e)$  over  $\mathbb{R}^{r_\ell}$ 
18:         $W^i \leftarrow \tilde{W} \odot \frac{[[C + e\mu^\top]^2 + S]^{\frac{1}{2}} - (C + e\mu^\top)}{[T]}$ 
19:        % Z-minimization
20:         $V^i \leftarrow W^{i+1} + U^i$ ,  $B \leftarrow -\log(W_{\ell+1}^{k-1} H_{\ell+1}^{k-1}) - \nu V^i$ 
21:         $Z^{i+1} \leftarrow \frac{[W(e^{-B\nu})]}{\nu}$ 
22:        % Dual Updates
23:         $U^i \leftarrow U^{i-1} + W^i - Z^i$ 
24:         $i \leftarrow i + 1$ 
25:      end while
26:       $W_\ell^k \leftarrow W^i$ 
27:    else if  $l = L$  then
28:      % Update from [25]
29:    end if
30:  end for
31: end for
32: return  $\{W_L, H_L, H_{L-1}, \dots, H_1\}$ 

```

4.1 Facial feature extraction

Let us apply deep KL-NMF on the CBCL face data set, used in the seminal paper of Lee and Seung who introduced NMF to the machine learning community [21]. It contains 2429 faces, each with 19×19 pixels. We chose the penalty parameters λ_ℓ as in [9], that is, the λ_ℓ are chosen such that each term in the objective is equal to one another at initialization, that is, all values $\lambda_\ell D(W_{\ell-1}^{(0)}, W_\ell^{(0)} H_\ell^{(0)})$ are equal to one, where $(W_1^{(0)}, W_2^{(0)}, \dots, H_\ell^{(0)})$ is the initialization. This is the default in our implementation, with $\lambda_\ell = \frac{1}{D(W_{\ell-1}^{(0)}, W_\ell^{(0)} H_\ell^{(0)})}$.

The data matrix, $X \in \mathbb{R}^{2429 \times 361}$, contains vectorized images on its rows. As it is now well established, NMF, with $X \approx WH$, is able to extract facial features as the rows of H , such as eyes, noses and lips; see Figure 3.

Let us now apply multilayer and deep KL-NMF on this data set with three layers, and $r = [80, 40, 20]$. For each layer of multilayer KL-NMF, we run 1000 iterations of the standard MU. We initialize deep KL-NMF with 500 iterations of multilayer KL-NMF, and then run it for 500 iterations using our proposed Algorithm 1. This means that deep KL-NMF is initialized with the solution of multilayer KL-NMF obtained after only 500 iterations. We repeat this experiment 35 times, and Figure 2 reports the median of the evolution of the error of deep KL-NMF divided by the final error obtained by multilayer KL-NMF after 1000 iterations. As expected, these ratios are initially larger than 1, since deep KL-NMF is initialized with multilayer KL-NMF after only 500 iterations.

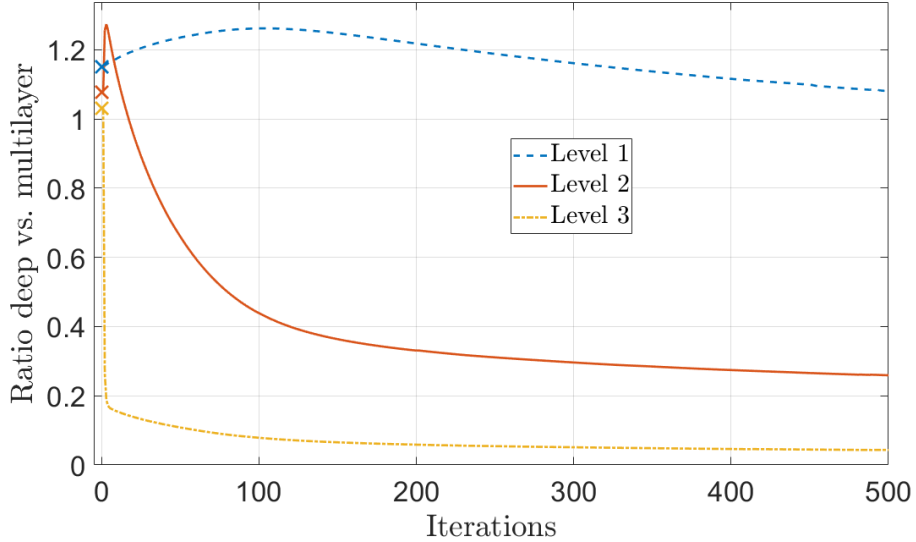


Figure 2: Evolution of the median errors at the different levels of deep KL-NMF (initialized with multilayer KL-NMF after 500 iterations) divided by the error of multilayer KL-NMF after 1000 iterations.

The main observation from Figure 2 is the following: Because deep KL-NMF needs to balance the error between each layer, the error at the first layer remains larger than that of multilayer KL-NMF, which is expected. However, the errors at the next two layers becomes quickly significantly smaller; see also the second column of Table 2. At convergence, the error at the second (resp. third) layer is on average about 4 (resp. 20) times smaller than that of the multilayer KL-NMF.

	error deep/multilayer (in %)	sparsity multilayer (in %)	sparsity deep (in %)
Layer 1	108.3 ± 1.1	58.5 ± 0.6	70.1 ± 1.2
Layer 2	26.8 ± 3.2	37.3 ± 1.0	50.6 ± 2.6
Layer 3	4.4 ± 0.3	22.1 ± 1.0	27.1 ± 2.9

Table 2: Deep vs. Multilayer KL-NMF: average and standard deviation for the error of deep KL-NMF divided by that of multilayer KL-NMF (second column), and average and standard deviation for the sparsity of the facial features of multilayer KL-NMF (second column) and deep KL-NMF (third column).

Table 2 also reports the average sparsity of the facial features at each layer, using the widely used Hoyer sparsity [19] given by

$$\text{sparsity}(x) = \frac{\sqrt{n} - \frac{\|x\|_1}{\|x\|_2}}{\sqrt{n} - 1} \in [0, 1],$$

for an n -dimensional vector x . This measure is equal to one if x has a single non-zero entry, and is equal to zero if all entries of x are equal to one another.

First, observe that for both deep and multilayer KL-NMF, the sparsity decreases as the factorization unfolds: this is unavoidable since facial features at deeper levels are nonnegative linear combinations of facial features at shallower levels. For example, the facial features at the second layer are given by $H_2 H_1$, that is, nonnegative linear combinations of the facial features at the first level, H_1 ; see also the discussion in Section 2.2. Second, we observe that deep KL-NMF produces significantly sparser facial features (namely +11.6% at layer 1, +13.2% at layer 2, +5.1% at layer 3). This makes sense because deep KL-NMF balances the error at the three layers, and sparser features at the first layer gives more degree of freedom to generate features at the next layers. In fact, a dense feature at the first layer can only generate denser ones at the next layers. This is an interesting side result of deep KL-NMF: it can be used to solve sparse NMF, without parameter tuning.

Figure 3 displays the facial features of multilayer and deep NMF at the different layers (for the last run of our experiment, since it does not make sense to average facial features). We observe that most of the facial features of the first two layers of deep and multilayer KL-NMF are similar, the main difference is that those of deep KL-NMF are sparser. However, at the third layer, where the error of deep KL-NMF is significantly smaller, most facial features are completely different (e.g., the first one), showing that deep KL-NMF produces significantly different outcomes than multilayer KL-NMF.

4.2 Topic modeling

Topic modeling aims to discover the underlying topics or themes in a collection of documents. It is a form of unsupervised learning that can help organize, summarize, and understand large textual datasets. In topic models, one typically assumes that a document is generated by a mixture of topics, each of which is a distribution over words in the vocabulary; see, e.g., [6] and the references therein. Topic modeling models aim to represent each document through the learned topics and understand the topics in the corpus through the most probable words of each topic.

NMF has been successfully used in this context, initiated by the paper of Lee and Seung [21]. If X is a word-by-document matrix, its NMF, $X \approx WH$, extract topics in W which is a word-by-topic matrix, while H allows one to classify the documents across the topics. In this context, deep NMF allows one to extract layers of topics: from lower level topics to higher level ones (e.g., tennis and



Figure 3: Example of facial features extracted by multilayer KL-NMF vs. deep KL-NMF.

football belong to sports). We will see an example below; see also [39] for a different but similar deep NMF model used for topic modeling. It has been well established that the KL divergence is more appropriate for the analysis of document data set. The reason is that such data sets are sparse (most documents only use a few words in the dictionary). In fact, the KL divergence amounts for a Poisson counting process; see, e.g., [16, Chapter 5] and the references therein.

In this section, we apply deep KL-NMF to the TDT2-top30 dataset and compare its performance with multilayer NMF. The TDT2 corpus (Nist Topic Detection and Tracking corpus) consists of data collected during the first half of 1998 and taken from 6 sources, including 2 newswires (APW, NYT), 2 radio programs (VOA, PRI) and 2 television programs (CNN, ABC). Only the largest 30 categories are kept, thus leaving us with 9394 documents in total [4].

We ran the experiment with three layers with $r = [20, 10, 5]$. Moreover, since deep NMF is computationally more intensive, we preprocess the data set to keep only the most important words. To do that, we perform a rank-20 NMF of X , and keep the 30 most important words in each topic

(that is, each column of W); we used the code from <https://gitlab.com/ngillis/nmfbook/>.

We use a similar setting as for the CBCL data set, except that we only run the algorithms once (we will focus on the analysis of the topics obtained), and we use $\lambda = [4, 2, 1]$, that is, we give more importance to the first layers. Otherwise, we observe the first layer topics were getting too similar: for example, giving too much importance to the term $\|W_1 - W_2 H_2\|_2$ will make W_1 to become rank deficient (since $r_2 < r_1$), and hence its columns will become more colinear.

Figure 4 reports the evolution of the error of deep KL-NMF divided by the final error obtained by multilayer KL-NMF, exactly as for the CBCL data set. As for CBCL, the errors of deep KL-NMF

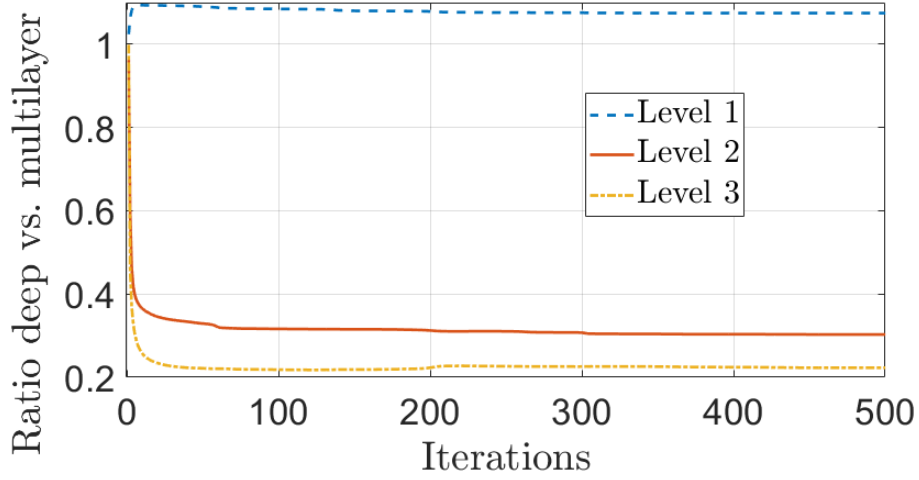


Figure 4: Evolution of the error at the different levels of deep KL-NMF divided by the error of multilayer KL-NMF.

at the first level is slightly larger (namely 107.5%), while the error at the second and third levels are significantly smaller (namely 30% and 22%, respectively) than that of multilayer NMF.

Let us try to analyse the topics extracted by deep KL-NMF, and the relationships between them. For each column of W_ℓ , which contain the topics, we sort the words in order of importance, and report them in that order. Table 3 provides the 10 most important words for the second layer, and Table 4 provides the 5 most important words for the third layer. For simplicity, we do not report the topics at layer 1 (there are 20), but a similar analysis can be made, and they are available from our code online.

What is interesting with deep NMF models, is that we can link the topics together, as in a hierarchical decomposition. For example, the topics of layers 2 and 3, within the columns of W_2 and W_3 respectively, are linked via the relation

$$W_2(:, j) \approx \sum_{k=1}^{r_3} W_3(:, k) H_3(k, j) \quad \text{for } j = 1, 2, \dots, r_2,$$

where $H_3(k, j)$ tells us the importance of the k th topic at level 3 to reconstruct the j th topic at level 2. The last line of Table 4 provides this information. Because most topics extracted at the second layer

Topic 1: Asian politics	Topic 2: Tobacco bill	Topic 3: American politics	Topic 4: Nuclear conflicts	Topic 5: Stock market	Topic 6: Media	Topic 7: Asian economy	Topic 8: Olympic games	Topic 9: Clinton-Lewinsky scandal	Topic 10: Iraq war
united states president indonesia suharto general american government indonesian jakarta	tobacco industry defense bill companies smoking military legislation health officials	president clinton house political government china administration white visit rights	nuclear india pakistan government party president tests minister indias political	percent market stock economic bank asia prices asian economy israel	spkr says reporter news game correspondent going headline voa reports	economic crisis billion financial economy asian government asia bank south	olympic city games team olympics nagano gold won women police	lewinsky president spkr clinton white house starr lawyers jury case	iraq weapons united iraqi states saddam military inspectors baghdad security

Table 3: Most important words in each topic of the second layer of deep KL-NMF.

Topic 1 Politics	Topic 2 Military conflicts	Topic 3 Economy	Topic 4 Clinton-Lewinsky	Topic 5 Mixed topics
president clinton government nuclear india house political china pakistan party	iraq united states weapons iraqi military saddam security inspectors president	percent economic crisis economy asian financial billion market bank asia	spkr lewinsky president clinton white house starr lawyers news case	tobacco olympic team city games going olympics won nagano gold
Merged topics 3 & 4 from layer 1	Merged topics 1, 6, & 10 from layer 1	Merged topics 5 & 7 from layer 1	Merged topics 6 & 9 from layer 1	Merged topics 2, 6 & 8 from layer 1

Table 4: Most important words in each topic of third layer of deep KL-NMF.

are rather different and use different words, H_3 is rather sparse (this in turn is because X is sparse, and hence the W_ℓ and H_ℓ 's are as well). For example, the topic 1 of the third layer (about politics) merges the topics 3 and 4 from the second layer (about American politics and nuclear conflicts), and the the topic 3 of the third layer (about economics) merges the topics 5 and 7 from the second layer (about stock market and the Asian economic crisis). Except for the fifth topic of the third layer which is a mixture of heterogeneous ones (mostly olympic games, but mixed with the tobacco billa and the media topics), the other ones are rather meaningful. Interestingly, the topic at level 2 about the media (with words such as reporter, correspondent, headline, etc.) is merged into three topics where the media is present (military conflicts, political scandals, Olympic games).

4.3 Hyperspectral imaging

In this section, we consider hyperspectral images to evaluate the effectiveness of the proposed minimum-volume deep KL-NMF model solved via Algorithms 2, in comparison to the multilayer KL-NMF [7, 8], and the Frobenius-norm based deep MF framework with minimum-volume penalty and data-centric loss function recently proposed in [9]. To ease the notation, the latter will be dubbed as “LC-DMF”. All the algorithms are implemented and tested on a laptop computer with Intel Core i7-11800H@2.30GHz CPU, and 16GB memory.

4.3.1 Data sets

A hyperspectral image (HSI) is an image that contains information over a wide spectrum of light instead of just assigning primary colors (red, green, and blue) to each pixel as in RGB images. The spectral range of typical airborne sensors is 380-12700 nm and 400-1400 nm for satellite sensors. For instance, the AVIRIS airborne hyperspectral imaging sensor records spectral data over 224 continuous channels. The advantage of HSI is that they provide more information on what is imaged, some of it blind to the human eye as many wavelengths belong to the invisible light spectrum. This additional information allows one to identify and characterize the constitutive materials present in a scenery. We

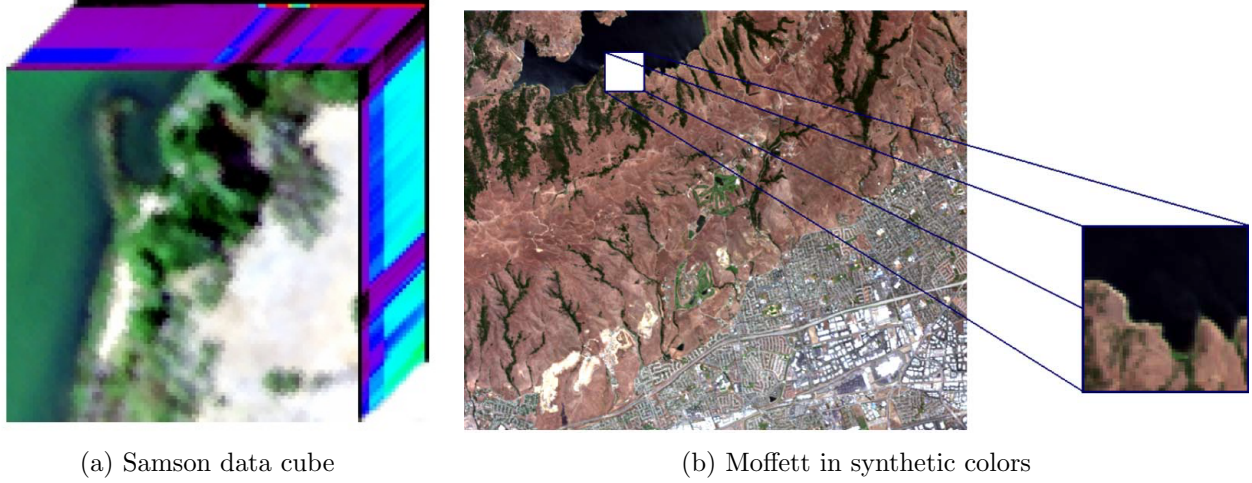


Figure 5: HSI data sets: Samson image (left) - Moffett Field image acquired by AVIRIS in 1997 and the region of interest (right) represented in synthetic colors, figure reproduced from [11].

consider the following real HSI:

- **AVIRIS Moffett Field:** this data set has been acquired with over Moffett Field (CA, USA) in 1997 by the JPL spectro-imager AVIRIS³ and consists of 512×614 pixels and 224 spectral reflectance bands in the wavelength range 400nm to 2500nm. Due to the water vapor and atmospheric effects, we remove the noisy spectral bands. After this process, there remains 159 bands. As in [11], we extract a 50×50 sub-image from this data set, see Figure 5b. It is widely acknowledged within the hyperspectral remote sensing community that this subimage consists of three distinct materials: vegetation, soil, and water. It is worth noting that the norm of the spectral signature for water is significantly smaller compared to the other two materials. For more detailed information on this dataset, please refer to [11].
- **Tumor:** 519 spectral bands with 13×11 pixels, corresponding to a simulated Magnetic resonance spectroscopic imaging (MRSI) of a glioma patient's brain [27]. This MRSI grid contains spectra from normal tissue, as well as tumor tissue and necrosis, see [27] for more details about this data set.
- **Samson:** The Samson dataset⁴ comprises 156 spectral bands and has a resolution of 95×95 pixels. It primarily contains three to four materials, namely "Soil," "Tree," and "Water." In Figure 5a, the hyperspectral cube is depicted, and upon closer examination, it becomes apparent that the "Soil" material actually consists of at least two sub-components, specifically sand and rocks.

4.3.2 Results

In this section, we present the results obtained from the benchmarked methods for each hyperspectral imaging (HSI) dataset, as described in Section 4.3.1. Specifically, we showcase the abundance maps

³https://aviris.jpl.nasa.gov/data/image_cube.html

⁴<http://lesun.weebly.com/hyperspectral-data-set.html>

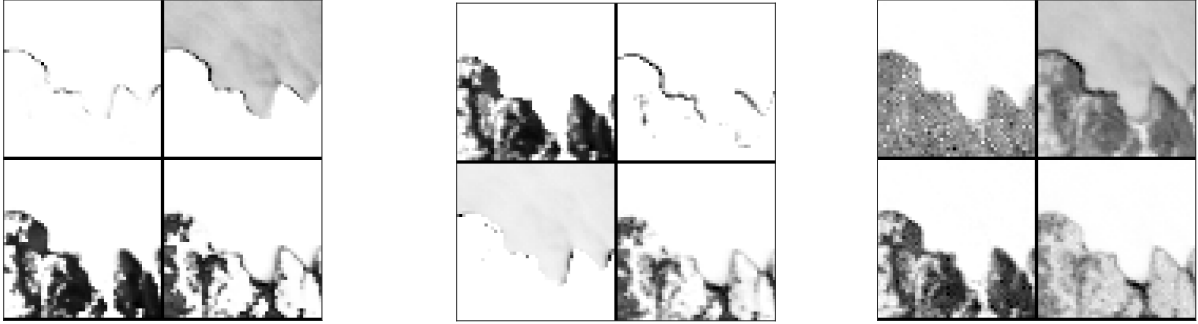
obtained for each layer of the deep models, aiming to provide a qualitative assessment of the unmixing and clustering outcomes. For all the models under consideration, we enforce a two-layer decomposition. Our objective is to achieve an accurate estimation and localization of the constitutive materials, also known as endmembers, in the first layer. The subsequent layer is expected to provide a clustering effect by merging the endmembers into more general clusters, such as vegetation vs. non-vegetation as illustrated in Figure 1, or mineral vs. organic or healthy vs non-healthy human tissues.

To ensure completeness and reproducibility of the results, we will now provide additional details regarding the parameters that were considered for each analysis.

- **AVIRIS Moffett Field:** For the AVIRIS Moffett Field dataset, we consider factorization ranks of $r_1 = 4$ and $r_2 = 2$, with a maximum number of iterations set to 300. In the deep MF framework with a minimum-volume penalty [9], we impose a sum-to-one constraint on the columns of factors W_l . The values for penalty weights of minimum-volume regularizations have been tuned and set to $[2; 1]$ to obtain the best results. For our proposed Algorithms 2, we set the parameter ρ to 100, the threshold ϵ to 10^{-6} for the ADMM procedure and the penalty weights of minimum-volume regularizations to $[4; 1]$.
- **Tumor:** For this data set, we consider factorization ranks of $r_1 = 3$ and $r_2 = 2$, with a maximum number of iterations set to 200. As done for previous data set, we impose a sum-to-one constraint on the columns of factors W_l for LC-DMF [9] with default values for min-vol penalty weights. In the case of Algorithms 2, the parameters for the ADMM procedure are the same as the ones considered above, whereas the penalty weights of minimum-volume regularizations are set to $[\frac{1}{4}10^{-4}; 10^{-2}]$.
- **Samson:** we consider factorization ranks of $r_1 = 4$ and $r_2 = 2$, with a maximum number of iterations set to 800. Again, we impose a sum-to-one constraint on the columns of factors W_l for LC-DMF [9] with values for penalty weights of minimum-volume regularizations tuned and set to $[0.15; 0.15]$ to obtain the best results. In the case of Algorithms 2, the parameters for the ADMM procedure are the same as the ones for previous data sets, whereas the penalty weights of minimum-volume regularizations are set to $[0.45; 0.45]$.

Discussion for AVIRIS Moffett Field: Figure 6 presents the abundance maps obtained for each layer of the three models. Impressively, both min-vol deep NMF models accurately detect the presence of water in the first layer, as well as a discernible "material" observed at the interface between the water and the soil. This interface showcases non-linear effects that arise from the phenomenon of double scattering of light. Both min-vol deep models effectively highlight these effects, with our proposed method demonstrating slightly superior accuracy in capturing such intricate features. It is worth noting that the estimation of water in this dataset is highly challenging, and the most successful results have been achieved by imposing sum-to-one constraints on the H factor of NMF models, as discussed in detail in [16]. Notably, the deep KL-NMF model provides the most accurate estimation of water. Moving on to Figures 7, we observe the abundance maps obtained for the final layer of the three models. Once again, both min-vol deep KL-NMF models yield more meaningful outcomes. While LC-DMF [9] distinguishes between vegetation and soil through clustering, min-vol deep KL-NMF (Alg. 2) gathers soil and vegetation, contrasting them with water.

Discussion for Tumor data set: Figure 8 illustrates the abundance maps obtained for each layer of the three models. As reported in [27], the data set comprises three endmembers: the "necrosis"



(a) min-vol deep KL-NMF (Alg. 2)

(b) LC-DMF [9]

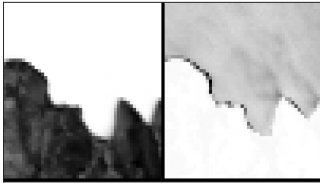
(c) Multi-layer KL-NMF [7, 8]

Figure 6: AVIRIS Moffett Field data set: From left to right abundance maps extracted from the first layer of min-vol deep KL-NMF (Alg. 2), LC-DMF [9] and multi-layer KL-NMF [7, 8].

forming a ball in the lower corner of the MRSI grid, the aureole-shaped tumor surrounding the necrosis, and the healthy tissue. Overall, all three deep models produce satisfactory results. In this analysis, it is evident that the min-vol deep NMF models more accurately extract this information, with a slightly cleaner localization achieved by min-vol deep KL-NMF (Alg. 2). Examining the second layers extracted by the models depicted in Figures 9, we observe that all three models exhibit a similar clustering of the endmembers extracted in the first layer, distinguishing non-healthy tissue (tumor + necrosis) from healthy tissue. Once again, min-vol deep KL-NMF (Alg. 2) showcases a slightly better separation in this regard.

Discussion for Samson data set: Figure 10 illustrates the abundance maps obtained for each layer of the three models. Interestingly, both min-vol deep NMF models successfully extract four materials, as explained in Section 4.3.1: water, vegetation, soil, and an additional material that likely corresponds to a second type of soil, possibly rocks. These models also capture non-linear effects at the interface between soil and water. It is worth noting that min-vol deep KL-NMF (Alg. 2) demonstrates better estimation of the localization of water and soil, while LC-DMF [9] provides slightly improved separation for vegetation, resulting in fewer residuals from the soil. Moving on to Figures 7, we observe the abundance maps obtained for the final layer of the three models. Remarkably, all three deep models combine the endmembers extracted in the first layer into clusters: minerals (soil + water) versus vegetation.

Conclusion: Our proposed approach, min-vol deep KL-NMF (Alg. 2), demonstrates promising results in hyperspectral imaging compared to two other methods. Despite its unconventional use of KL divergence instead of the Frobenius norm, deep KL-NMF exhibits superior accuracy in capturing intricate features, particularly in detecting water and the water-soil interface. It outperforms other methods in accurately estimating water, a challenging task in the AVIRIS Moffett Field dataset. Additionally, deep KL-NMF achieves satisfactory results in the Tumor dataset with improved localization and separation. In the Samson dataset, deep KL-NMF successfully extracts multiple materials and offers better localization of water and soil.



(a) min-vol deep KL-NMF (Alg. 2)

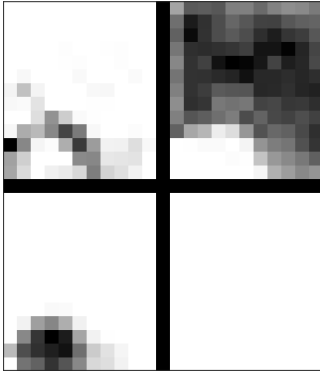


(b) LC-DMF [9]

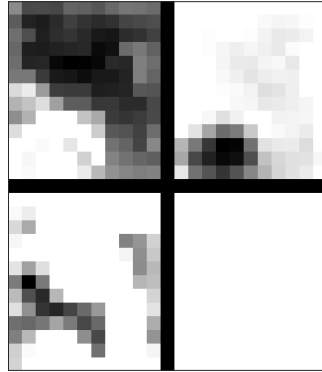


(c) Multi-layer KL-NMF [7, 8]

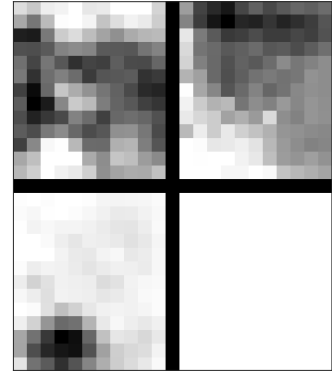
Figure 7: AVIRIS Moffett Field data set: From left to right abundance maps extracted from the second layer of min-vol deep KL-NMF (Alg. 2), LC-DMF [9] and multi-layer KL-NMF [7, 8].



(a) min-vol deep KL-NMF (Alg. 2)

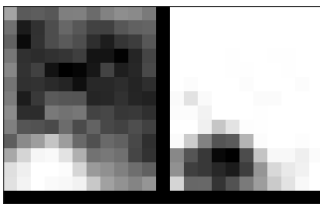


(b) LC-DMF [9]

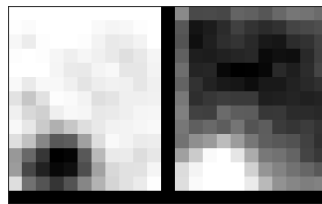


(c) Multi-layer KL-NMF [7, 8]

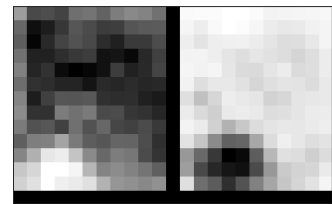
Figure 8: Tumor data set: From left to right abundance maps extracted from the first layer of min-vol deep KL-NMF (Alg. 2), LC-DMF [9] and multi-layer KL-NMF [7, 8].



(a) min-vol deep KL-NMF (Alg. 2)

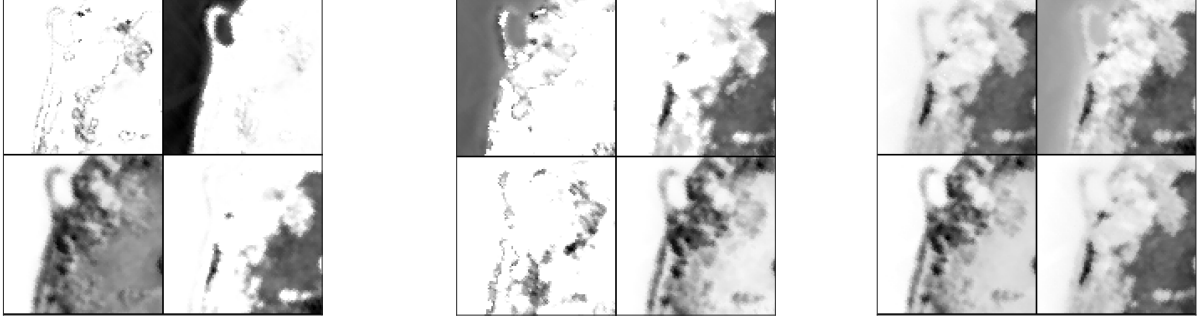


(b) LC-DMF [9]



(c) Multi-layer KL-NMF [7, 8]

Figure 9: Tumor data set: From left to right abundance maps extracted from the second layer of min-vol deep KL-NMF (Alg. 2), LC-DMF [9] and multi-layer KL-NMF [7, 8].

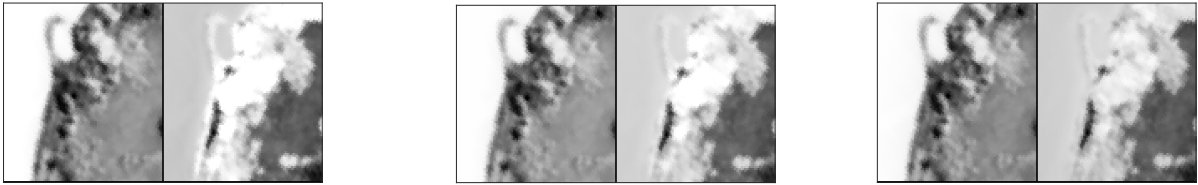


(a) min-vol deep KL-NMF (Alg. 2)

(b) LC-DMF [9]

(c) Multi-layer KL-NMF [7, 8]

Figure 10: Samson data set: From left to right abundance maps extracted from the first layer of min-vol deep KL-NMF (Alg. 2), LC-DMF [9] and multi-layer KL-NMF [7, 8].



(a) min-vol deep KL-NMF (Alg. 2)

(b) LC-DMF [9]

(c) Multi-layer KL-NMF [7, 8]

Figure 11: Samson data set: From left to right abundance maps extracted from the second layer of min-vol deep KL-NMF (Alg. 2), LC-DMF [9] and multi-layer KL-NMF [7, 8].

5 Conclusion

We have introduced two new deep NMF models based on β -divergences and the layer-centric loss function. Leveraging the BSUM method, we devised efficient algorithms to estimate the parameters of these models. Our experimental results underscored the practical efficacy of these approaches in diverse applications, including facial feature extraction, topic modeling, and hyperspectral image unmixing. Our future research direction includes exploring acceleration strategies of TITAN to further accelerate convergence of the proposed algorithms. Additionally, we aim to extend the applications of deep NMF in other domains such as source separation and gene expression analysis.

References

- [1] Abdolali, M., Gillis, N.: Simplex-structured matrix factorization: Sparsity-based identifiability and provably correct algorithms. *SIAM Journal on Mathematics of Data Science* **3**(2), 593–623 (2021)
- [2] Arora, S., Cohen, N., Hu, W., Luo, Y.: Implicit regularization in deep matrix factorization. *Advances in Neural Information Processing Systems* **32** (2019)
- [3] Arora, S., Ge, R., Kannan, R., Moitra, A.: Computing a nonnegative matrix factorization—provably. In: *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pp. 145–162 (2012)
- [4] Cai, D., Mei, Q., Han, J., Zhai, C.: Modeling hidden topics on document manifold. In: *Proceeding of the 17th ACM Conference on Information and Knowledge Management, CIKM’08*, p. 911–920 (2008)
- [5] Chen, W.S., Zeng, Q., Pan, B.: A survey of deep nonnegative matrix factorization. *Neurocomputing* **491**, 305–320 (2022)
- [6] Churchill, R., Singh, L.: The evolution of topic modeling. *ACM Computing Surveys* **54**(10s), 1–35 (2022)
- [7] Cichocki, A., Zdunek, R.: Multilayer nonnegative matrix factorisation. *Electronics Letters* **42**, 947–948 (2006)
- [8] Cichocki, A., Zdunek, R.: Multilayer nonnegative matrix factorization using projected gradient approaches. *International Journal of Neural Systems* **17**(06), 431–446 (2007)
- [9] De Handschutter, P., Gillis, N.: A consistent and flexible framework for deep matrix factorizations. *Pattern Recognition* **134**, 109,102 (2023)
- [10] De Handschutter, P., Gillis, N., Siebert, X.: A survey on deep matrix factorizations. *Computer Science Review* **42**, 100,423 (2021)
- [11] Dobigeon, N., Moussaoui, S., Coulon, M., Tournieret, J.Y., Hero, A.O.: Joint bayesian endmember extraction and linear unmixing for hyperspectral imagery. *IEEE Transactions on Signal Processing* **57**(11), 4355–4368 (2009). DOI 10.1109/TSP.2009.2025797

- [12] F  votte, C., Idier, J.: Algorithms for nonnegative matrix factorization with the β -divergence. *Neural computation* **23**(9), 2421–2456 (2011)
- [13] Fu, X., Huang, K., Sidiropoulos, N.D.: On identifiability of nonnegative matrix factorization. *IEEE Signal Processing Letters* **25**(3), 328–332 (2018)
- [14] Fu, X., Huang, K., Sidiropoulos, N.D., Ma, W.K.: Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications. *IEEE Signal Process. Mag.* **36**(2), 59–80 (2019)
- [15] Fu, X., Ma, W.K., Huang, K., Sidiropoulos, N.D.: Blind separation of quasi-stationary sources: Exploiting convex geometry in covariance domain. *IEEE Transactions on Signal Processing* **63**(9), 2306–2320 (2015)
- [16] Gillis, N.: *Nonnegative matrix factorization*. SIAM, Philadelphia (2020)
- [17] Hien, L.T.K., Gillis, N.: Algorithms for nonnegative matrix factorization with the Kullback-Leibler divergence. *Journal of Scientific Computing* (87), 93 (2021)
- [18] Hien, L.T.K., Phan, D.N., Gillis, N.: An inertial block majorization minimization framework for nonsmooth nonconvex optimization. *Journal of Machine Learning Research* **24**(18), 1–41 (2023). URL <http://jmlr.org/papers/v24/21-0571.html>
- [19] Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research* **5**(Nov), 1457–1469 (2004)
- [20] Huang, K., Sidiropoulos, N.D., Swami, A.: Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing* **62**(1), 211–224 (2013)
- [21] Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. *Nature* **401**(6755), 788–791 (1999)
- [22] Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: *Advances in neural information processing systems*, pp. 556–562 (2001)
- [23] Leplat, V., Ang, A.M., Gillis, N.: Minimum-volume rank-deficient nonnegative matrix factorizations. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3402–3406. IEEE (2019)
- [24] Leplat, V., Gillis, N., Ang, A.M.: Blind audio source separation with minimum-volume beta-divergence nmf. *IEEE Transactions on Signal Processing* **68**, 3400–3410 (2020)
- [25] Leplat, V., Gillis, N., Idier, J.: Multiplicative updates for nmf with beta-divergences under disjoint equality constraints. *SIAM Journal on Matrix Analysis and Applications* **42**(2), 730–752 (2021). DOI 10.1137/20M1377278
- [26] Li, X., Zhu, Z., Li, Q., Liu, K.: A provable splitting approach for symmetric non-negative matrix factorization. *IEEE Trans. Knowl. Data Eng.* **35**(3), 2206–2219 (2023). 10.1109/TKDE.2021.3125947

- [27] Li, Y., Sima, D.M., Van Cauter, S., Himmelreich, U., Pi, Y., Van Huffel, S.: Simulation study of tissue type differentiation using non-negative matrix factorization. In: BIOSIGNALS, pp. 212–217 (2012)
- [28] Lin, C.H., Ma, W.K., Li, W.C., Chi, C.Y., Ambikapathi, A.: Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case. *IEEE Transactions on Geoscience and Remote Sensing* **53**(10), 5530–5546 (2015)
- [29] Ma, W.K., Bioucas-Dias, J.M., Chan, T.H., Gillis, N., Gader, P., Plaza, A.J., Ambikapathi, A., Chi, C.Y.: A signal processing perspective on hyperspectral unmixing: Insights from remote sensing. *IEEE Signal Processing Magazine* **31**(1), 67–81 (2013)
- [30] Mairal, J.: Optimization with first-order surrogate functions. In: Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, ICML’13, pp. 783–791. JMLR.org (2013)
- [31] Mongia, A., Jhamb, N., Chouzenoux, E., Majumdar, A.: Deep latent factor model for collaborative filtering. *Signal Processing* **169**, 107,366 (2020)
- [32] Nesterov, Y.: Lectures on Convex Optimization. Springer Cham (2018)
- [33] Razaviyayn, M., Hong, M., Luo, Z.Q.: A unified convergence analysis of block successive minimization methods for nonsmooth optimization. *SIAM Journal on Optimization* **23**(2), 1126–1153 (2013)
- [34] Sun, Y., Babu, P., Palomar, D.: Majorization-minimization algorithms in signal processing, communications, and machine learning. *IEEE Transactions on Signal Processing* **65**, 794–816 (2017). DOI 10.1109/TSP.2016.2601299
- [35] Thanh, O.V., Ang, A., Gillis, N., Hien, L.T.K.: Inertial majorization-minimization algorithm for minimum-volume nmf. In: 2021 29th European Signal Processing Conference (EUSIPCO), pp. 1065–1069 (2021). DOI 10.23919/EUSIPCO54536.2021.9616152
- [36] Trigeorgis, G., Bousmalis, K., Zafeiriou, S., Schuller, B.W.: A deep semi-NMF model for learning hidden representations. In: International Conference on Machine Learning, pp. 1692–1700. PMLR (2014)
- [37] Trigeorgis, G., Bousmalis, K., Zafeiriou, S., Schuller, B.W.: A deep matrix factorization method for learning attribute representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **39**(3), 417–429 (2016)
- [38] Vavasis, S.A.: On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization* **20**(3), 1364–1377 (2010)
- [39] Will, T., Zhang, R., Sadovnik, E., Gao, M., Vendrow, J., Haddock, J., Molitor, D., Needell, D.: Neural nonnegative matrix factorization for hierarchical multilayer topic modeling. arXiv preprint arXiv:2303.00058 (2023)
- [40] Ye, F., Chen, C., Zheng, Z.: Deep autoencoder-like nonnegative matrix factorization for community detection. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1393–1402 (2018)