UNROLLING REGULARIZED NON-NEGATIVE MATRIX FACTORIZATION FOR SPATIAL TRANSCRIPTOMICS DECONVOLUTION

Grant Konkel[⋆] *Jiasen Zhang*[⋆] *Weihong Guo*[⋆] *Liangliang Zhang*[†]

[⋆] Case Western Reserve University, Mathematics, Applied Mathematics and Statistics.
[†] Case Western Reserve University, Department of Population and Quantitative Health Sciences

ABSTRACT

Being able to locate cell types within the body is a task that can be critical for disease treatment and prevention and for making more general discoveries. Ståhl et al.'s discovery of using "Spatial Transcriptomics" data to aid us in this task proved to be a particularly useful tool [1]. By deconvolving Spatial Transcriptomics data, we can estimate cell type proportion versus locational data and identify regions with larger proportions of any certain cell type. Within this paper, we propose a method of deconvolving this spatial transcriptomics data using an unrolled non-negative matrix factorization neural network called Spatial Transcriptomics Learned Non-negative Matrix Factorization (STL NMF). STL NMF combines a standard location-based graph Laplacian regularity with region membership attained from histology information. This new regularity is then factored into its NMF multiplicative update rule. This rule is derived from an objective function whose parameters are learned through algorithm unrolling. STL NMF gives promising results and beats, or is comparable to, all methods we compared with.

***Index Terms*—** Spatial Transcriptomics, Deconvolution, Non-negative Matrix Factorization, Unrolled Algorithm, Graph Laplacian, Group Sparsity

1. INTRODUCTION

Biologically, we know that genes and cells are inherently linked through position. Using "Spatial Transcriptomics" (ST) data (first proposed by Ståhl et. al in 2016 [1]), we can transform a spatial mapping of genes into a spatial mapping of cell types. Though the applications of this are countless, this is particularly meaningful in biological studies such as cancer and tumor research [2][3].

Going from the ST data to the cell-type data, however, is a non-trivial task. One strategy in doing this is to locate cell-type abundances through matrix deconvolution. Though there are other deep learning alternatives, there are mathematical trade-offs to using these methods. Matrix deconvolution models provide mathematical interpretability that many deep learning models sacrifice for performance gains.

1.1. Deconvolution

In our setup of this strategy, our ST data X (gene count versus location matrix) is modeled as the product of our gene expression reference data B and our cell type proportion matrix V, giving us $X = BV + \varepsilon$. Our goal is to start with X and B and to "deconvolve" the data to solve for V. There are a number of strategies used for deconvolution and can generally be categorized as NMF-based, probabalistic, deep learning or optimal transport. Each method of deconvolution can also be placed into one of two overarching categories: reference-free or reference-based.

1.2. Literature Review

Reference-free methods are methods of deconvolution that attempt to solve for V with only X given (i.e. B and V are both unknowns). Reference-free methods can thus be used on more datasets, and can attempt to solve a higher percentage of problems. The leading method is FAST [4] which relies on NMF.

Reference-based models, on the other hand, are strategies of reproducing V assuming that the matrix B is available. Reference-based models can achieve higher accuracy than reference-free methods, but can only solve problems where we have access to B. This matrix, however, is available and can be found by biologists with domain knowledge, making B a reasonable requirement. Some examples of reference-based models include SPOTlight [5], CARD [6] and Stereoscope [7]. SPOTlight and CARD are NMF-based, and Stereoscope is probabalistic [8].

Within both the reference-free and reference-based categories, there exist deep learning methods that solve for V by relying on machine learning techiques. These methods can achieve higher accuracy, however, they do so at the sacrifice of mathematical interpretability.

1.3. Algorithm Unrolling

A sacrifice of interpretability is not the only weakness of deep learning models. In order to achieve accurate results, a model

must be carefully constructed and must train on a sufficient amount of high-quality, realistic data.

Algorithm unrolling (often called unfolding), attempts to soften the weaknesses of these deep learning methods (while preserving their strengths) by rooting them deeper into mathematical models [9]. First proposed in 2010 by Gregor and LeCun [10], algorithm unrolling consists of transforming an iterative algorithm into a neural network with each layer, an instance of the iterative update. This network can then train on the mathematical model's parameters, and by using standard back propagation, these parameters can quickly converge to the optimal values. These new models are easily interpretable and have weaker requirements of high-quality data.

1.4. Related Works

One existing method that is similar to ours is FAST [4]. FAST is also a regularized NMF-based model that operates under similar constraints to STL NMF. Largely, STL NMF can be seen as an unrolled version of FAST, with differences in the objective function and in application. FAST is a reference free method, meaning it attempts to solve for V *without* having the matrix B. FAST uses the following objective function:

$$\min_{V^T \geq 0, B \geq 0} \|X - BV\|_F^2 + \lambda_1 Tr\left(VLV^T\right) + \lambda_2 \|V^T J - J_m\|_F^2$$

Also similar is Deep NMF [11], which demonstrates the act of unrolling the NMF algorithm, though not of applying it to Spatial Transcriptomics data. DeepNMF uses the following objective function:

$$\hat{B}, \hat{V} = \arg \min_{B \geq 0, V^T \geq 0} D_b\left(X | BV\right) + \mu \|V\|_1$$

Other methods of ST deconvolution include SPOTlight [5], CARD [6] and Stereoscope [7] which served as our comparisons in testing STL NMF. SPOTlight uses biological marker genes and does deconvolution through NMF and NNLS. Stereoscope assumes a negative binomial distribution and then finds the cell type that was most likely responsible for generating the gene expression observed. CARD uses a conditional auto-regressor and builds upon a Non-negative Matrix Factorization model. We compared STL NMF against both CARD and Stereoscope.

1.5. STL NMF

In this paper, we propose Spatial Transcriptomics Learned Non-negative Matrix Factorization. Our method unrolls a regularized version of the NMF algorithm, with update rules derived from an objective function that minimizes the difference between X and BV (as in the classical NMF algorithm) alongside a graph laplacian regularization term which ensures "closeness" between locations, a group sparsity term [12] and a term encouraging columns to sum to one. This model is unrolled into an iterative algorithm and is trained to find optimal parameters.

2. PROBLEM FORMULATION AND METHOD

We revisit our problem and formulate it as follows. We start with the Spatial Transcriptomics data $X \in \mathbb{R}^{n \times m}$. This data represents the count of n genes in m locations. Because we are dealing in the reference-based setting, we are also given the matrix $B \in \mathbb{R}^{n \times k}$. This matrix B shows the abundance of n genes within k cell-types. We aim to find $V \in \mathbb{R}^{k \times m}$ which shows the proportion of m locations that is made up of k cell-types.

Given X and B, solving $X = BV + \varepsilon$ for V is still an ill-posed problem. In particular, the solution is non-unique. Fortunately, we have information on each of our matrices as well as additional histology information, allowing us to add additional constraints to find an accurate solution.

Biologically, locations that are close together and are within the same biological region should have genetic makeups much more similar than spots far away. This means that we can introduce an adjacency term within our objective function that limits the freedom of NMF and ensures similarities across locations. We define a square adjacency matrix $A \in \mathbb{R}^{m \times m}$ with:

$$a_{i,j} = \exp\left[-\frac{(x_i - x_j)^2 + (y_i - y_j)^2}{\sigma_1^2} - \frac{\|z_i - z_j\|_1^2}{\sigma_2^2}\right] \quad (1)$$

Note: $0 \leq a_{i,j} \leq 1$ for all $a_{i,j} \in A$. Within this equation, x_i, y_i, z_i represent the x-location, y-location and region membership for spot i. Region membership comes from annotating the histology information given and can be set to constant when histology is not available. σ_1 and σ_2 are parameters for weighting the importance of location versus region and are trained during the training of our neural network. We also define matrices L and D according to:

$$D_{jj} = \Sigma A_{jl} \quad \text{and} \quad L = D - A \quad (2)$$

To turn these matrices into an actual quantifiable term, we can use $\mathrm{Tr}(VLV^T)$, which can also be viewed as $\sum_{i,j} a_{i,j} \|V_i - V_j\|_2^2$. By minimizing this operation, we ensure that similar locations ($a_{i,j} \approx 1$) will have similar cell type proportions (V_j, V_i). The same such restrictions do not apply for dissimilar locations. This constraint can be rewritten as $\mathrm{Tr}(VDV^T) - \mathrm{Tr}(VAV^T)$.

We also know that because each column of V represents a proportion, each column should sum to one. To encourage our deconvolution to do this, we implement the same term as FAST. The term is as follows:

$$\|V^T J - J_m\|_F^2 \quad (3)$$

Where $J \in \mathbf{1}^{k \times m}$ and $J_m \in \mathbf{1}^{m \times m}$. $V^T J$ creates a matrix sized $m \times m$ with each index of each row set to the sum of that row of V^T.

Finally, we wanted to further weight the region membership through a group sparsity term that groups locations by region. This works by ensuring that locations in the same region have similar cell-type proportions by comparing the simulated cell type proportions with the estimated proportions (learned) [13]. Mathematically, this can be shown as follows:

$$\|WV\|_F^2 \tag{4}$$

Where W is our m-tall group sparsity matrix created by stacking region cell type proportion estimates, with each row corresponding to the region of each location. Each region membership vector is defined as $W_{i \in \{1,\ldots,\text{count}(regions)\}} = [\omega_1, ..., \omega_k]_i$ Where $\omega_1, \omega_2, ..., \omega_k$ are learned and are different for each region.

Adding these three terms to our objective function, we get:

$$\begin{aligned} \min_{V^T \geq 0} \|X - BV\|_F^2 + \lambda_1 \operatorname{Tr}(VLV^T) \\ + \lambda_2 \|V^T J - J_m\|_F^2 + \lambda_3 \|WV\|_F^2 \end{aligned} \tag{5}$$

By taking the derivative w.r.t. V^T, we get our multiplicative update rule:

$$V^T = \frac{V_{jk}^T (X^T B + \lambda_1 A V^T + \lambda_2 J_m J_m^T)_{jk}}{(V^T B^T B + \lambda_1 D V^T + \lambda_2 V^T J J^T + \lambda_3 V^T W^T W)_{jk}} \tag{6}$$

After each update, we divide each column of V by its 1-norm.

With this formulation, we have the backbone of our model. X and B are given (along with location $\rightarrow$ region membership classification, coming from our histology data). A is calculated according to parameters σ_1 and σ_2. Group sparsity is determined by our learned $W_1, W_2, ..., W_{nreg}$. The multiplicative update rule is determined from λ_1, λ_2 and λ_3.

The results of the model depend on σ_1 and σ_2 which balance location versus membership, $W_1, ..., W_{nreg}$ which are our estimates for cell type proportions, and $\lambda_1, \lambda_2, \lambda_3$ which balance our data fidelity and regularity terms. We set up an unrolled deep neural network to learn these parameters. This allows us to optimize our model while preserving the interpretability of our otherwise purely mathematical model. To do this, we unroll our iterative algorithm such that each layer of our network is an instance of our multiplicative update rule. The setup of the neural network can be visualized by Figure 1. An important step in training our model is generating synthetic data when ample data isn't available. Because

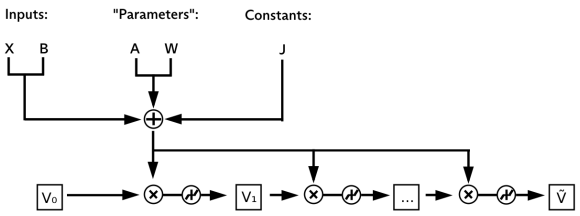

Fig. 1. Block Diagram

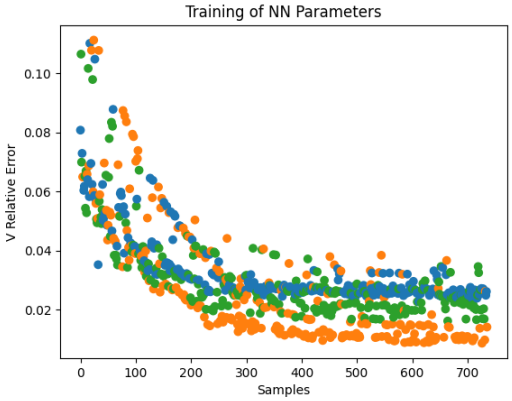

Fig. 2. Training Results

our model is supervised, this data also requires a ground truth V, which is often unavailable. To obtain enough X, B pairs with ground-truth V, we assumed constant B, generated many Vs and multiplied BV to get X. X and B are inputted into our network and the output $\tilde{V}$ is compared with V using the frobenius norm of their difference to calculate the loss function.

From there, we can train our method on the model parameters $\lambda_1, \lambda_2, \lambda_3, \sigma_1, \sigma_2, W_1, W_2, W_3$ (though the number of W_i is dependent on the number of regions in the dataset). The results of the training visualized through relative error of V can be seen in Figure 2.

3. DATA

To compare with other methods (SPOTlight, CARD and Stereoscope), we tested STL NMF on piecewise constant synthetic data in a 2 region $\times$ 2 region grid, piecewise synthetic data with 3 regions that mimic the PDAC-A dataset and on the real PDAC-A dataset. Information about the datasets and current results are shown below.

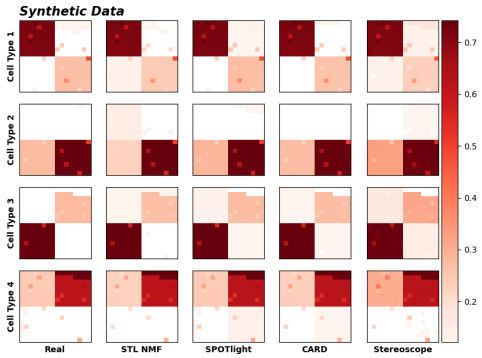

Fig. 3. V Visualization, 2x2 Grid

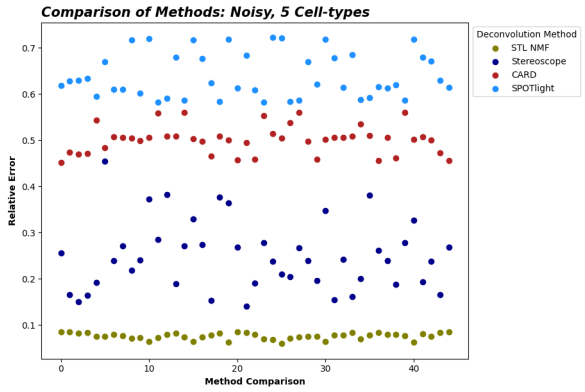

Fig. 4. Compare Error, 5 Cell Types, Noisy

3.1. Piecewise Synthetic Grid

The piecewise synthetic grid was generated according to 4 even cells and attempted to follow some realistic biological noise. After generating B and V, we calculated X, giving us a ground truth triplet.

3.2. PDAC Synthetic

To generate the PDAC synthetic data, we use the gene-expression matrix B generated from CARD, extract the columns we are interested in (corresponding to cell types), generate a piecewise synthetic V and finally calculate X from BV. We generated some X with no added noise, and some with added noise to test both the theoretical cap as well as more realistic robustness. For 3 cell type data, the columns of B we kept corresponded to Acinar cells, Cancer clone A and Cancer clone B. For 5 cell types, we kept Acinar Cells, Cancer Clone A, Endocrine cells, Fibroblasts and Ductal - terminal ductal like cell-types.

3.3. Real Data

Unlike the synthetic data described above, the real PDAC A dataset has no ground truth V, meaning its much harder to quantify the results. However, many cell-types have marker genes which are present in our spatial transcriptomics data. Suppose gene a is found primarily in cancer clone A and location i has an abundance of gene a, transitively there is a good chance that cancer clone A is dominant in location i. By plotting the normalized count of these marker genes, we can approximate a ground truth V to compare our own method with. Also we can compare our method with others to see if our results are close to their approximations.

4. RESULTS

Comparing STL NMF with other methods produces strong results. STL NMF gives comparable results to Stereoscope and CARD on clean (unrealistic) data, and significantly outperforms all methods when any noise is present. STL NMF beats SPOTlight across the board.

Starting with Figure 3., we can see the deconvolution results given 2×2 synthetic data which represents a simple problem. Though no method is perfect, all methods give comparable results and do not give conclusive findings for one method or another.

Though not shown, the results of the 3 cell type PDAC synthetic dataset tells much of the same story. SPOTlight could not be compared (it consistently ran into issues on NNLS), but STL NMF, CARD and Stereoscope all showed comparably strong results and could not point to one method being superior or inferior. Moving to the five cluster PDAC synthetic dataset, we begin to see the separation between STL NMF and the other methods. For five clusters with clean X files, STL NMF is better than Stereoscope, but both methods are competitive. CARD is not too far behind and SPOTlight is not able to deconvolve without error.

For five clusters and noisy X files, we see the largest definitive indication that STL NMF can outperform the other methods. STL NMF is better than stereoscope and far better than CARD or SPOTlight. This can be viewed in Figures 4 and 5.

When testing real data, we have no ground truth and so evaluating performance is difficult. One way to estimate the true V is to use the gene counts for certain marker genes which are especially prevalent in the cell-types we care about. For this we used the marker genes presented in CARD which are derived from domain knowledge. By calculating the absolute and relative errors, STL NMF was closer than the other methods, though qualitative analysis may be more appropriate. Looking at Figure 6., SPOTlight and Stereoscope both have intensities and sparsities that closer match the marker

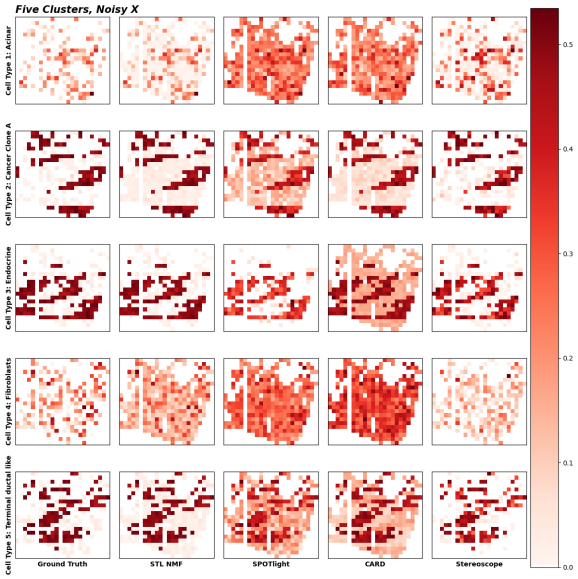

Fig. 5. V Visualization, 5 Cell Types, Noisy

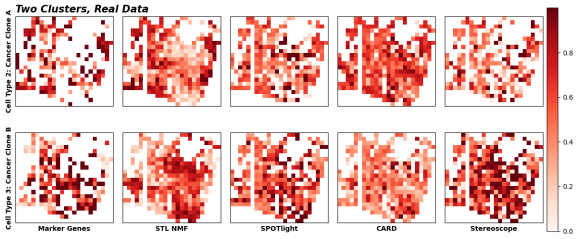

Fig. 6. V Visualization, 2 Cell Types, Real

genes. However, marker genes are often sparser than the cell types they represent and the general spatial structure that they reveal can be a more useful comparison. STL NMF shows the clearest spatial structure of the methods, which also looks to match the marker genes.

5. CONCLUSION

Though the results of the real data are not definitive, STL NMF seems to perform better than the other models in terms of spatial structure and outperforms when computing "loss" (though this "loss" term may not be fully accurate). For synthetic data, however, STL NMF has conclusively better results. This is especially true for the more-realistic, noisy synthetic data, which best shows off its strengths and promises. For perfect data, CARD and Stereoscope are comparable or better, but STL NMF remains competitive and the addition of any noise suggests that STL NMF could be the superior method. SPOTlight can be reasonably competitive for noisy data, but stuggles to run without error for the other datasets.

Outside of accuracy, it is also very important to note runtimes of each method. The training of STL NMF is pretty quick (usually $\approx$ 15 minutes with adequate parameter initializations) and testing of STL NMF takes about 10 seconds per X. SPOTlight is slightly slower, with initial training always required and ranging from 15-40 minutes depending on the size of the inputs. SPOTlight, CARD and Stereoscope all start with the single-cell matrix S (such that $SCN = B$, where S is gene count by single-cells, C is the clustering of single-cells representing cell-type membership and N is a normalization matrix that corresponds with C and is the number of cells in a cluster over the number of unique cells in a cluster). By starting with S, they calculate B and then deconvolve to find V which takes much more time and takes the same amount of time per run (little efficiency boost by training for CARD and Stereoscope, SPOTlight speeds up considerably). CARD takes about 25 minutes to test the 45 X matrices, SPOTlight takes about 20 minutes, Stereoscope takes over 2 hours.

Finally, another important factor when comparing the methods is that STL NMF needs to train on data before it can be tested. This, obviously, means that we must create synthetic data that behaves similarly to our test data (since in practice there are no ground truth V matrices for real data). As we do not always know how the input data will behave, this is not always possible. This requirement is one of the main downsides of a neural network based approach. However, because our model is unrolled from a mathematical model, we require much less training than other deep-learning methods and our requirements for realistic data are much weaker. In addition, even without proper training, we can behave as well as other mathematical models such as FAST (as FAST is similar to our method, but without optimized parameters).

6. FUTURE CONSIDERATIONS

Future directions for this project include taking a closer look at the group sparsity term. This term is helpful with training data that will closely mimic the test data, but may provide little to the model when similarity between the two is lacking. Another future direction is finding another regularizer that better takes advantage of algorithm unrolling. As the majority of the model parameters are scalars, running grid or random search on

7. ACKNOWLEDGEMENTS

This work was originally supported by a summer research scholarship provided by CWRU Undergraduate Research Office and the Case Alumni Association.

8. REFERENCES

[1] Patrik L Ståhl, Fredrik Salmén, Sanja Vickovic, Anna Lundmark, José F Navarro, Jens Magnusson, Stefania Giacomello, Michael Asp, Jakub O Westholm, Mikael Huss, Annelie Mollbrink, Sten Linnarsson, Simone Codeluppi, Åke Borg, Fredrik Pontén, Paul Igor Costea, Pelin Sahlén, Jan Mulder, Olaf Bergmann, Joakim Lundeberg, and Jonas Frisén, "Visualization and analysis of gene expression in tissue sections by spatial transcriptomics," *Science*, vol. 353, no. 6294, pp. 78–82, 2016.

[2] Q. Yu, M. Jiang, and L. Wu, "Spatial transcriptomics technology in cancer research," *Frontiers in Oncology*, vol. 12, pp. 1019111, Oct. 2022.

[3] R. Arora, C. Cao, M. Kumar, et al., "Spatial transcriptomics reveals distinct and conserved tumor core and edge architectures that predict survival and targeted therapy response," *Nature Communications*, vol. 14, pp. 5029, 2023.

[4] Meng Zhang, Yiwen Liu, Joel Parker, Lingling An, and Xiaoxiao Sun, "Flexible analysis of spatial transcriptomics data (fast): A deconvolution approach," *bioRxiv*, 2023.

[5] Marc Elosua-Bayes, Pablo Nieto, Elisabetta Mereu, Ivo Gut, and Holger Heyn, "Spotlight: seeded nmf regression to deconvolute spatial transcriptomics spots with single-cell transcriptomes," *Nucleic Acids Res*, vol. 49, no. 9, pp. e50, 2021.

[6] Y. Ma and X. Zhou, "Spatially informed cell-type deconvolution for spatial transcriptomics," *Nature Biotechnology*, vol. 40, no. 9, pp. 1349–1359, Sep. 2022.

[7] Anna Andersson, Joseph Bergenstråhle, Michaela Asp, and et al., "Single-cell and spatial transcriptomics enables probabilistic inference of cell type topography," *Commun Biol*, vol. 3, pp. 565, 2020.

[8] H. Li, J. Zhou, Z. Li, et al., "A comprehensive benchmarking with practical guidelines for cellular deconvolution of spatial transcriptomics," *Nature Communications*, vol. 14, pp. 1548, 2023.

[9] Vishal Monga, Yuelong Li, and Yonina C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," 2020.

[10] Karol Gregor and Yann Lecun, "Learning fast approximations of sparse coding," 08 2010.

[11] Rami Nasser, Yonina C. Eldar, and Roded Sharan, "Deep unfolding for non-negative matrix factorization with application to mutational signature analysis," *Journal of Computational Biology*, vol. 29, no. 1, pp. 45–55, 2022, PMID: 34986029.

[12] Chunyang Cui, Xinyu Wang, Shaoyu Wang, Liangpei Zhang, and Yanfei Zhong, "Unrolling nonnegative matrix factorization with group sparsity for blind hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 61, pp. 1–12, 2023.

[13] Chunyang Cui, Xinyu Wang, Shaoyu Wang, Liangpei Zhang, and Yanfei Zhong, "Unrolling nonnegative matrix factorization with group sparsity for blind hyperspectral unmixing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–1, 01 2023.