Supporting information:

Unsupervised Multi-Clustering and Decision-Making Strategies for 4D-STEM Orientation Mapping

Junhao Cao[1,2], Nicolas Folastre[1,2], Gozde Oney[3], Edgar Rauch[4], Stavros Nicolopoulos[5], Partha Pratim Das[5], Arnaud Demortière[1,2,6*]

[1]Laboratoire de Réactivité et Chimie des Solides (LRCS), CNRS UMR 7314, Université de Picardie Jules Verne, Hub de l'Energie, Rue Baudelocque, 80039 Amiens Cedex, France.

[2]Réseau sur le Stockage Electrochimique de l'Energie (RS2E), CNRS FR 3459, Hub de l'Energie, Rue Baudelocque, 80039 Amiens Cedex, France.

[3]Institut de Chimie de la Matière Condensée de Bordeaux (ICMCB), Bordeaux, France.

[4]Université Grenoble Alpes, CNRS, Grenoble INP, SIMAP, 38000 Grenoble, France

[5]NanoMegas company, Belgium

[6]ALISTORE-European Research Institute, CNRS FR 3104, Hub de l'Energie, Rue Baudelocque, 80039 Amiens Cedex, France.

Corresponding Author: *arnaud.demortiere@cnrs.fr

Non-negative matrix factorization algorithm

Lee and Seung provided the alternating optimized method to Equation (**S**2) in their seminal paper [29].

$$\min ||V - WH||_F, \text{ subject to } W \geq 0, H \geq 0 \qquad (2)$$

$$W^{n+1}{}_{ik} \leftarrow W_{ik} \sum_j \frac{V_{ij}}{(WH)_{ij}} H_{kj}$$

$$H^{n+1}{}_{kj} \leftarrow H_{kj} \sum_j \frac{V_{ij}}{(WH)_{ij}} W_{ik}$$

Schematizes the procedure of most NMF factorization [31].

```
Input: Non-negative matrix V (M * N) and component k
Output: (W, H) ≥ 0: V ≈ W * H
        The initialization of W and H,
    subject (W, H) ≥ 0
        For i = 1, 2, 3 …. max_iter do

        END for
```

Dataset processing from 4D to 2D

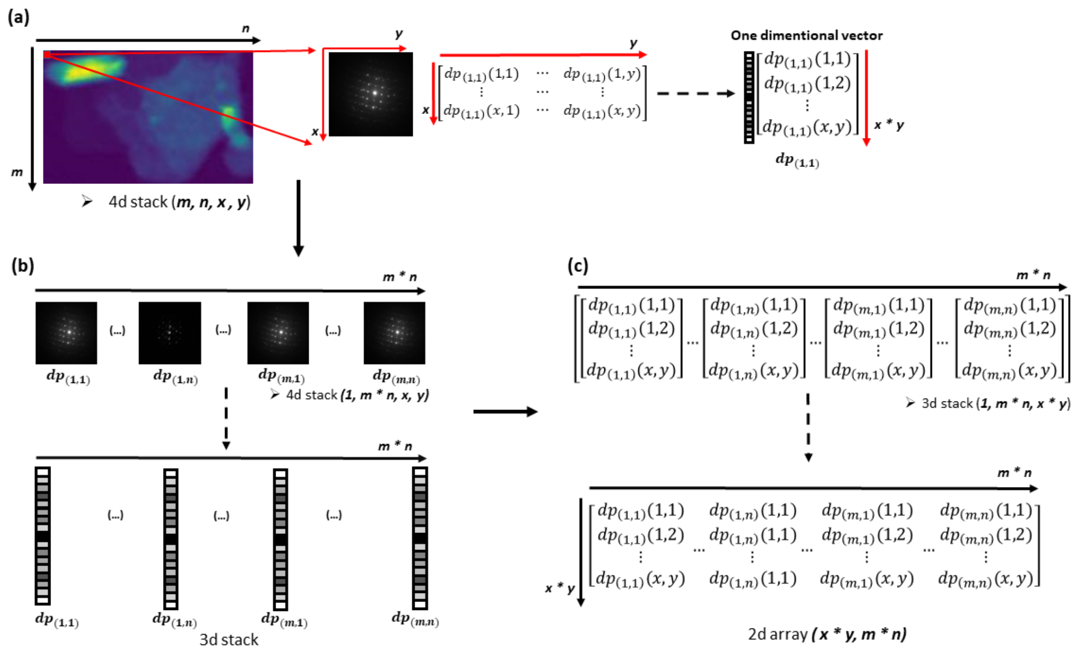

Figure SI_1 depicts the process of transforming a 4D dataset into a 2D array, which refers to the idea from the paper 'Non-negative matrix factorization for mining big data obtained using four-dimensional scanning transmission electron microscopy' [1].

(a) The initial dataset is a 4D stack denoted by dimensions ***(m, n, x, y)***. A 4D stack is shown on the left side with indices ***m*** and ***n*** representing the first two dimensions. Each point of ***(m, n)*** corresponds to a 2D diffraction pattern with dimensions ***(x, y)***, as depicted in the magnified section. For example, the red point represents one pixel (one diffraction pattern) in the original dataset. $dp_{(1,1)}$ represents is the first diffraction pattern in the original dataset, moreover, $dp_{(1,1)}(1,1)$ is first pixel in the first diffraction pattern. Each 2D diffraction pattern is flattened into a 1D vector. This vector has a length of ***x * y***, where each element is sequentially arranged into a single column vector, namely, from ***(x, y)*** to ***(x * y, 1)***. **(b)** The ***(m, n)*** grid of 2D diffraction patterns is flattened into a single dimension, where the first dimension is 1, the second dimension is ***m * n***, and the third and fourth dimensions remain as (x, y). Among them, each 2D diffraction pattern, now a 1D vector, is arranged sequentially along the new second dimension of the 3D stack. This essentially lines up the ***m * n*** 1D vectors side-by-side. **(c)** The 3D stack is further flattened into a 2D array. The first dimension of the 2D array corresponds to the length of the flattened 1D vectors, ***x* * *y***. The second dimension of the 2D array corresponds to the number of such vectors, ***m* * *n***. Finally, the 2D array is represented with dimensions (***x* * *y, m* * *n***). Specifically, each column in the 2D array represents a flattened 1D vector of a single diffraction pattern from the original 4D stack. Simultaneously, each row corresponds to a particular pixel value from the flattened diffraction patterns, the result of which is for suiting the input of NMF [1], [2].

Calculation of the value of Noise Standard Deviation (NSD)

$$\text{NSD} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(I_i - \mu)^2}$$

Algorithm SI_1 shows the calculating of the value of **Noise Standard Deviation (NSD)**

[3] (https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5946565)

NSD is the Noise Standard Deviation. ***N*** is the total number of pixels in the image. I_i is the pixel value at position i in the image. ***μ*** is the average pixel value of the image. Typically, the process involves the following steps:

1. Compute the average pixel value **(*μ*)** of the image.
2. Calculate the squared difference between each pixel value and the average pixel value.
3. Take the average of these squared differences.
4. Finally, take the square root of this average to obtain the ***NSD***.

Calculation of the value of Peak Signal-to-Noise Ratio (PSNR)

$$\text{PSNR} = 10\ log_{10}(\frac{MAX^2}{MSE})$$

$$\text{MSE} = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - R(i,\ j)]^2$$

Algorithm SI_2 shows the calculating of the value of **Peak Signal-to-Noise Ratio (PSNR)** between the original image and the reference image [4]

MAX is the maximum possible pixel value of the image. For an 8-bit image, this is typically 255. **MSE** is the **Mean Squared Error** between the original and reference images. For the calculation of MSE, normally, $I(i,j)$ is the pixel value at position (i,j) in the original image, likewise, $R(i, j)$ is the pixel value at corresponding position (i, j) in the reference image. Finally, ***m*** and ***n*** are the dimensions of the images [4] [5].

Calculation of the value of the Mean Deviation Similarity Index (MDSI)

The process of calculation concerns converting the images to luminance and chromaticity channels, computing gradient and chromaticity similarities, combining these similarities, and applying deviation pooling to get the final index [6].

$$L = 0.2989R + 0.5870G + 0.1140B \quad (1)$$

$$[H\ M\ ] = (0.30\ 0.04\ 0.34\ \ -0.60\ \ -0.35\ 0.17\ )[R\ G\ B\ ] \quad (2)$$

Algorithm SI_3_1 Convert the Images to Luminance and Chromaticity Channels

Algorithm SI_3_1 (1) converts the reference (R) and distorted (D) images to luminance (L) and chromaticity channels using the specified formula; Algorithm SI_3_1 (2) use the Gaussian color model to obtain the chromaticity channels H and M [6].

$$G_x(x) = h_x * f(x) \text{ and } G_y(x) = h_y * f(x) \quad (1)$$

$$G(x) = \sqrt{{G_x}^2(x) + {G_y}^2(x)} \quad (2)$$

$$GS(x) = \frac{2G_R(x)G_D(x) + C_1}{{G_R}^2(x) + {G_D}^2(x) + C_1} \quad (3)$$

$$F = 0.5 \times (R + D) \quad (4)$$

$$GS_{RF}(x) = \frac{2G_R(x)G_F(x) + C_2}{{G_R}^2(x) + {G_F}^2(x) + C_2} \quad (5)$$

$$GS_{DF}(x) = \frac{2G_D(x)G_F(x) + C_2}{{G_D}^2(x) + {G_F}^2(x) + C_2} \quad (6)$$

$$\widehat{GS}(x) = GS(x) + [GS_{DF}(x) - GS_{RF}(x)] \quad (7)$$

Algorithm SI_3_2 Compute Gradient Similarity (GS) and Proposed Gradient Similarity ($\hat{GS}$)

Algorithm SI_3_2 (1), (2), at first, calculate the gradient magnitudes of the luminance channels of the reference and distorted images using the Sobel operator [7]; then Algorithm SI_3_2 (3) compute the gradient similarity (GS) [6]. Algorithm SI_3_2 (4), (5), (6) will compute the fused luminance channel F and extra GS maps $GS_{RF}(x)$ *and* $GS_{DF}(x)$; finally, Algorithm SI_3_2 (7) calculate the proposed gradient similarity ($\hat{GS}(x)$), C_1, C_2 are constants used to control numerical stability [6].

$$\hat{CS}(x) = \frac{2H_R(x)H_D(x) + 2M_R(x)M_D(x) + C_3}{{H_R}^2(x) + {H_D}^2(x) + {M_R}^2(x) + {M_D}^2(x) + C_3}$$

Algorithm SI_3_3 Compute Chromaticity Similarity ($\hat{CS}$)

Algorithm SI_3_3 shows that calculate the chromaticity similarity using both chromaticity channels, C_3 has the same objective with C_1, C_2 [6].

$$G\hat{C}S(x) = \alpha\hat{GS}(x) + (1 - \alpha)\hat{CS}(x)$$

Algorithm SI_3_4 Combine Gradient and Chromaticity Similarity Maps

Algorithm SI_3_4 is to combine the gradient and chromaticity similarity maps using a weighted average, α is a parameter to adjust the relative importance of gradient and chromaticity similarity maps. [6].

$$MDSI = \sqrt{\frac{1}{N}\sum_{x=1}^{N}(G\hat{C}S(x) - \mu_{G\hat{C}S})^2}$$

Algorithm SI_3_5 Apply Deviation Pooling Strategy

Algorithm SI_3_5 is to use the deviation pooling to compute the final MDSI score. Deviation pooling is based on a general formulation. Here, $\mu_{G\hat{C}S}$ is the mean of the combined similarity map values, and N is the number of pixels [6].

Algorithm SI_4: Calculation of the value of Gradient Magnitude Similarity Deviation (GMSD)

Gradient Magnitude Similarity Deviation (GMSD) is a perceptual image quality assessment (IQA) metric designed to measure the quality of an image by comparing its gradient magnitudes to those of a reference image [8].

$$GMS\ (i,j) = \frac{2G_I(i,\ j)G_{I'}(i,j) + C}{G_I(i,\ j)^2 + G_{I'}(i,j)^2 + C}$$

Algorithm SI_4_1 Gradient Magnitude Similarity (GMS)

Algorithm SI_4_1 Compute the gradient magnitude for both the reference image and the distorted image using a gradient operator like the Sobel filter [7]. The gradient magnitude at a pixel is calculated as the square root of the sum of the squared gradients in the horizontal and vertical directions [7]. For each pixel, calculate the Gradient Magnitude Similarity (GMS) index between the reference image I and the distorted image *I'*. $G_I(i,\ j)$ and $G_{I'}(i,j)$ are the gradient magnitudes of the reference and distorted images at pixel *(i,j)*, and C is a small constant to avoid division by zero [8] *.*

$$GMSD = \sqrt{\frac{1}{N}\sum_{i\ =\ 1}^{N}(GMS(i)\ -\ MGMS(i))^2}$$

Algorithm SI_4_2 Gradient Magnitude Similarity Deviation (GMSD)

Algorithm SI_4_2 Mean Gradient Magnitude Similarity (MGMS) is the value of the mean of the GMS values over all pixels in the image [8]. Compute the standard deviation of the GMS values across all pixels [8]. The final GMSD value is given by the standard deviation of these GMS values [8]. N is the total number of pixels, and MGMS is the mean GMS value.

Algorithm SI_5: Calculation of the value of structural similarity index measure (SSIM)

The Structural Similarity Index Measure (SSIM) is a perceptual metric introduced by Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli in their 2004 paper titled "Image Quality Assessment: From Error Visibility to Structural Similarity"

[9]. SSIM is used for measuring the similarity between two images, aiming to quantify the visual impact of changes in image structure [9], [10].

$$\text{SSIM}(x, y) = [l(x,y)]^{\alpha} \times [c(x,y)]^{\beta} \times [s(x,y)]^{\gamma}$$

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{{\mu_x}^2 + {\mu_y}^2 + C_1}$$

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{{\sigma_x}^2 + {\sigma_y}^2 + C_2}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

Algorithm SI_5 The formula for SSIM between images x and y

Algorithm SI_5 SSIM combines three components: luminance (l), contrast (c), and structure (s):

- **Luminance**: Measures the similarity in brightness between images.
- **Contrast**: Evaluates the similarity in contrast.
- **Structure**: Assesses the similarity in structures of the images, such as edges and textures.

α, β, γ are parameters that define the relative importance of each component [9]. μ_x *and* μ_y are the means of image x and image y, σ_x and σ_y are the standard deviations, and $\sigma_{xy}$is the covariance of x and y. C_1, C_2, and C_3 are constants to avoid instability when the value of denominators is very small [9].

Global plan of NMF-Clustering Analysis

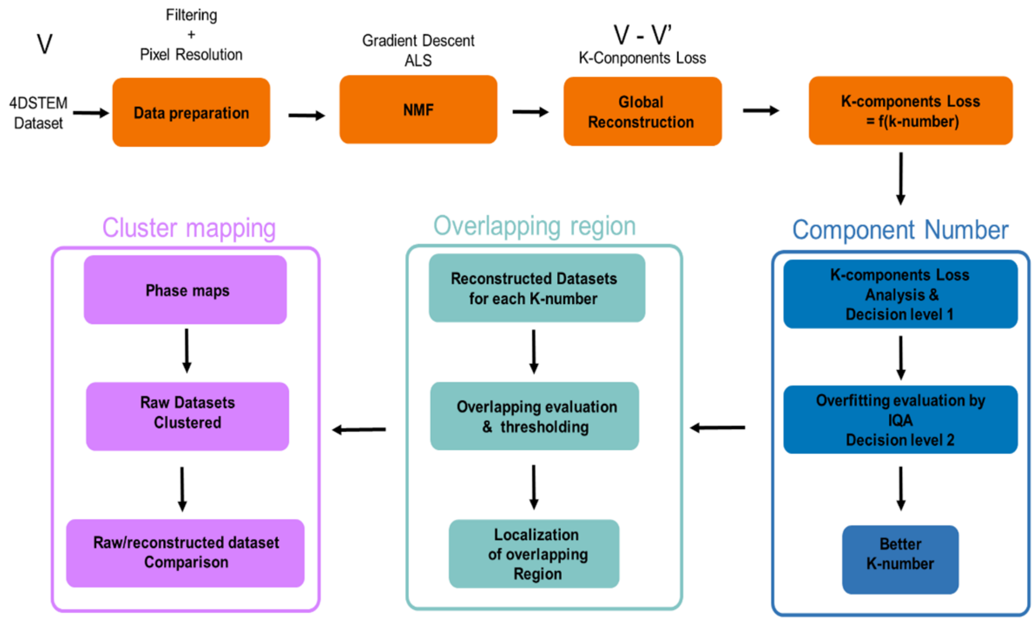
V
Filtering
+
Pixel Resolution
Gradient Descent
ALS
V - V'
K-Conponents Loss
4DSTEM Dataset
Data preparation
NMF
Global Reconstruction
K-components Loss = f(k-number)
Cluster mapping
Phase maps
Raw Datasets Clustered
Raw/reconstructed dataset Comparison
Overlapping region
Reconstructed Datasets for each K-number
Overlapping evaluation & thresholding
Localization of overlapping Region
Component Number
K-components Loss Analysis & Decision level 1
Overfitting evaluation by IQA Decision level 2
Better K-number

Figure SI_2 represents a workflow for analyzing a 4DSTEM dataset using non-negative matrix factorization (NMF). The 4DSTEM dataset V through the filtering and pixel resolution adjustment to prepare the data for further analysis. The prepared data is subjected to NMF using alternating least squares (ALS) methods to decompose the dataset into two components to facilitate the later analysis. The difference between the original dataset V and the reconstructed dataset V′ is calculated through K-components loss. The K-components loss is analyzed to determine the initial decision level (Decision level 1). An overfitting evaluation using an IQA (Image Quality Assessment) method is conducted to make a refined decision on the optimal number of components (Decision level 2). This process results in the determination of a better K-number for the dataset. Once the optimal (best/ optimal) cluster number is fixed, the reconstructed datasets for different K-numbers are evaluated to identify overlapping regions. The overlapping evaluation is dependent on thresholding performed to localize the overlapping regions within the dataset. Finally, the phase maps are created from the localized overlapping regions. Simultaneously, the raw datasets are clustered based on the phase maps. A comparison is made between the raw and reconstructed datasets to ensure the accuracy and relevance of the clustering.

Algorithm SI_6: ePattern Algorithm

The ePattern algorithm [11] is a sub-pixel adaptive image processing method developed to enhance diffraction pattern quality and reliability for pattern matching in 4D-STEM. It is based on a modular pipeline comprising preprocessing, registration, and reconstruction. The goal is to isolate and quantify the crystalline diffraction signal while minimizing noise, amorphous contributions, and background effects.

Preprocessing includes:

- Histogram optimization and alignment of raw diffraction patterns (DPs).

- Background subtraction using a rolling-ball filter.
- Application of a Gaussian blur to improve peak detection.

Registration detects and localizes diffraction peaks using:

- Adaptive prominence thresholds based on noise levels.
- Sub-pixel position refinement via center-of-mass calculations.
- Radius estimation through local standard deviation analysis.
- Intensity computation from average gray values within each spot.

This information is stored in a latent table containing spot coordinates, radius, and intensity per DP. Reconstruction generates a clean image by plotting only the registered Bragg peaks. This yields high signal-to-noise reconstructed DPs and enables strong compression (×100–1000) of 4D datasets.The method preserves relative peak intensities while removing background and diffuse scattering. ePattern does not rely on iterative training or convolutional layers, but adopts an encoder–latent–decoder logic. This approach improves ACOM pattern matching accuracy and reduces overfitting, especially in low-SNR or beam-sensitive samples.

References:

[1] F. Uesugi, S. Koshiya, J. Kikkawa, T. Nagai, K. Mitsuishi, and K. Kimoto, "Non-negative matrix factorization for mining big data obtained using four-dimensional scanning transmission electron microscopy," *Ultramicroscopy*, vol. 221, p. 113168, 2021, doi: https://doi.org/10.1016/j.ultramic.2020.113168.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999, doi: 10.1038/44565.

[3] D. K. Lee, J. In, and S. Lee, "Standard deviation and standard error of the mean," *Korean J Anesthesiol*, vol. 68, no. 3, p. 220, 2015.

[4] D. R. Bull and F. Zhang, "Chapter 4 - Digital picture formats and representations," in *Intelligent Image and Video Compression (Second Edition)*, D. R. Bull and F. Zhang, Eds., Oxford: Academic Press, 2021, pp. 107–142. doi: https://doi.org/10.1016/B978-0-12-820353-8.00013-X.

[5] N. Burningham, Z. Pizlo, and J. P. Allebach, "Image quality metrics," *Encyclopedia of imaging science and technology*, vol. 1, pp. 598–616, 2002.

[6] H. Z. Nafchi, A. Shahkolaei, R. Hedjam, and M. Cheriet, "Mean deviation similarity index: Efficient and reliable full-reference image quality evaluator," *Ieee Access*, vol. 4, pp. 5579–5590, 2016.

[7] I. Sobel, “An Isotropic 3x3 Image Gradient Operator,” *Presentation at Stanford A.I. Project 1968*, Feb. 2014.

[8] W. Xue, L. Zhang, X. Mou, and A. C. Bovik, “Gradient magnitude similarity deviation: A highly efficient perceptual image quality index,” *IEEE transactions on image processing*, vol. 23, no. 2, pp. 684–695, 2013.

[9] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004, doi: 10.1109/TIP.2003.819861.

[10] Z. Wang, A. C. Bovik, and H. R. Sheikh, “Structural similarity based image quality assessment,” in *Digital Video image quality and perceptual coding*, CRC Press, 2017, pp. 225–242.

[11] Folastre, N. *et al.* Improved ACOM pattern matching in 4D-STEM through adaptive sub-pixel peak detection and image reconstruction. *Scientific Reports* **14**, 12385 (2024).