Journal of Machine Learning Research 25 (2024) 1-39 Submitted 5/23; Revised 10/24; Published 11/24

# Distributed Kernel-Driven Data Clustering

**Ioannis Schizas** IOANNIS.D.SCHIZAS.CIV@ARMY.MIL
*US Army Combat Capabilities Development Command*
*Army Research Lab*
*Aberdeen Proving Ground, MD 21005, USA*

**Editor:** Dan Alistarh

## Abstract

A novel fully distributed joint kernel learning and clustering framework is derived which is capable of determining clustering configurations in an unsupervised manner. Semidefinite programming is utilized to quantify closeness of candidate kernel similarity matrices to a block diagonal structure of certain rank. Utilizing difference of convex functions and block coordinate descent a recursive algorithm is derived that determines jointly a proper kernel similarity matrix and clustering factors. Reformulating the involved semidefinite programs in a separable way we build on the alternating direction method of multipliers, to construct a fully distributed scheme that enables joint kernel learning and clustering in ad hoc networks via collaborating neighboring agents. Convergence claims establish that the proposed algorithmic framework returns bounded similarity kernel updates promoting a block diagonal structure. Detailed numerical examples utilizing both synthetic and real data demonstrate that the distributed novel approach can achieve clustering performance that gets close or even exceeds the one achieved by existing centralized alternatives.

**Keywords:** Distributed learning, kernels, clustering, unsupervised learning, optimization

## 1. Introduction

Clustering data vectors into different groups sharing similar properties has been extensively studied as an unsupervised learning technique when data labels are not available. An essential aspect in data clustering is picking a proper similarity metric. For instance in K-Means (Li and Guo, 2018; Lloyd, 1982; Oliva et al., 2013), a workhorse in data clustering, each different cluster is represented by a centroid point and the data are assigned to the cluster whose centroid is closest with respect to a pre-selected distance metric. Different variants of matrix factorization have also been used to explore the clustering problem given a known data similarity matrix (Cai et al., 2010; Huang et al., 2013; Trigeorgis et al., 2016; Wang and Zhang, 2012).

To deal with non-linear settings kernel-based methods have been proposed that are capable of unveiling data vector correlations in higher-dimensional spaces. Kernel target alignment (Cortes et al., 2012; Cristianini et al., 2001; Müller et al., 2018) are such popular supervised approaches wherein the suitable kernel matrices are identified by finding the alignment or the normalized inner product between the kernel correlation matrices and the correlation of the class labels. Other kernel learning techniques for classification and regression problems rely on convex optimization (Ghari and Shen, 2020; Hoi et al., 2013; Jin et al., 2010; Motai, 2014), though they are still supervised or semi-supervised in nature.

Unsupervised approaches have been recently proposed to jointly construct a proper kernel similarity matrix while performing data clustering. The unsupervised approach in Ren and Sun (2020)

relies on building proper graph similarity matrices to model correlations among the data vectors via an affine weight strategy while preserving the data structure via proper constraints. Further, relying on tensor factorization the method in (Ren et al., 2020) offers a more computationally expensive option while improving clustering in a variety of different datasets. Another unsupervised joint kernel learning and clustering approach in (Malhotra and Schizas, 2020) utilizes a sparsity regularized non-negative matrix factorization along with eigenvalue maximization to find proper kernel covariance matrices that facilitate data clustering.

All aforementioned approaches are centralized in the sense of requiring a central processor to operate or multiple processing units in a tree formation (master-slave). Therefore they are not operational in ad hoc architectures involving multiple sensing/processing agents/units. For instance, distributed sensing units could correspond to accelerometers (e.g., smartphones) mounted on humans that engage in different activities, e.g., running, walking, jumping (Micucci et al., 2017) and so on based on the situation they are facing. Clustering the sensing units based on the activity they are monitoring is essential in facilitating situational awareness in large gatherings or tactical applications assessing how different groups of people behave. Distributed clustering in such situations is desirable to enable scaling as the number of agents increases.

Distributed data clustering techniques relying on splitting centralized K-means in localized processing tasks have been proposed (Chen et al., 2016; Oliva et al., 2013; Qin et al., 2016; Tsapanos et al., 2015) with some of them requiring a fusion center (master processing unit) (Chen et al., 2016; Li and Guo, 2018; Tsapanos et al., 2015). These approaches, as the centralized K-means approach rely on a preset data similarity metric which can considerably limit the clustering performance. On the other end, existing distributed kernel learning techniques rely on distributed optimization techniques to derive localized learning tasks and perform supervised learning tasks, i.e., regression and classification, (Bouboulis et al., 2017; Hong and Chae, 2021; Shin et al., 2018).

The main goal of this work is to derive a fully distributed algorithmic framework for *joint* kernel learning and clustering in an unsupervised manner. Existing approaches in clustering and kernel learning as mentioned earlier are centralized or rely on master-slave multiprocessing architectures (Malhotra and Schizas, 2020; Ren and Sun, 2020; Ren et al., 2020). Such approaches are not applicable in ad hoc multi-agent architectures where there is no central processing unit. Building on semidefinite programming (SDP) (Boyd and Vandenberghe, 2004), difference of convex functions (Tao and An, 1997) and block coordinate descent strategies (Tseng, 2001) we first obtain a centralized joint kernel learning and clustering formulation. Different from existing approaches, the novel minimization formulation is amenable to a separable reformulation which will enable the derivation of local learning tasks across the sensing units. Despite the fact that a reformulation is needed to obtain a separable non-equivalent formulation from the centralized one, due to single-hop connectivity in the network of sensing units, it is established that the centralized and distributed optimal solutions share block diagonal structure and equal rank which is crucial for the clustering task. The proposed distributed and centralized formulations are not equivalent from an optimization point of view, but their optimal solutions share a block diagonal structure.

Specifically, the alternating direction method of multipliers (ADMM) is employed to split the involved SDP programs in a set of local SDP tasks that can be tackled using local information at every sensing agent. A novel combination of ADMM, SDP programming and block coordinate descent is employed to devise a fully distributed algorithm that facilitates proper kernel learning and effective unsupervised clustering. Our work contributions are summarized as follows:

1. Derivation of a SDP-based framework quantifying closeness of candidate kernel mappings to a block diagonal structure that facilitates joint kernel learning and factorization-based clustering.

2. A novel minimization formulation that combines block coordinate descent and semidefinite programming that results a novel centralized joint kernel learning and clustering approach.

3. Derivation of pertinent localized SDP problems, where the alternating direction method of multipliers is being employed in the separable kernel learning and clustering formulation that leads to a fully distributed kernel learning and clustering approach that can be utilized in ad hoc multi-agent networks. It turns out that the communication costs among neighboring agents are linear in terms of the number of clusters, the size of the neighborhood and the kernel dictionary.

4. Convergence of the novel framework is established showing that the kernel iterates move towards a block diagonal structure facilitating data clustering.

5. Extensive numerical tests demonstrate the clustering accuracy advantage of the novel framework over existing centralized alternatives.

The problem setting and preliminary concepts are provided in Sec. 2, while explaining the idea of data clustering via a proper block diagonal kernel covariance matrix construction using an ad hoc multi-agent network. In Sec. 3 the novel SDP-based kernel learning and clustering formulation is derived, while a centralized algorithm is provided using block coordinate descent and difference of convex functions, with convergence to a finite bound being established. Sec. 4 builds a separable SDP-based formulation which is further tackled relying on ADMM that results localized kernel learning and clustering tasks that can be addressed locally at each agent which via collaboration with neighboring agents can tackle the global minimization formulation. Convergence analysis advocates that the novel algorithm learns a kernel data similarity matrix approaching a block diagonal structure that has the potential to reach the clustering performance of its centralized counterpart. Extensive numerical tests are performed in Sec. 5 for the novel algorithm along with state-of-the-art clustering methods using synthetic and real datasets and evaluating clustering accuracy, the normalized mutual information (NMI) and purity.

## 2. Problem Setting and Preliminaries

Consider $P$ agents spatially scattered across a field, with the $i$th unit acquiring signal measurements $x_i(t)$ across a time horizon of duration $t \in [0, T-1]$, i.e.,

$$x_i(t) = f_{q(i)}(s_{q(i)}(t)) + w_i(t), \ , i = 1, \ldots, P \tag{1}$$

where $f_{q(i)}(\cdot)$ represents a mapping between an underlying source $s_{q(i)}(t)$ and the measurement $x_i(t)$, while $w_i(t)$ denotes spatially uncorrelated sensing noise.

**Sensing assumption:** It is assumed that among the $Q$ underlying sources, each unit observes one of them, i.e., unit $i$ observes source $q(i)$ where $q(i) \in \{1, \ldots, Q\}$ is an underlying unknown mapping matching sources with observations. Each of the vectors $\mathbf{x}_i$ contains information about a

specific source namely $q(i) \in \{1, \ldots, Q\}$ (thus belongs to a specific cluster). Further, the mapping function $f_q(\cdot)$ is not available and nonlinear. Each agent groups its measurements in vector $\mathbf{x}_i := [x_i(0), \ldots, x_i(T-1)]^T$ with $T$ denoting the number of samples.

For the example given in the Introduction where different sensing units monitor the physical activity a person is engaging to (running, walking and so on), it is reasonable to assume that there is one dominant activity that will be measured from the accelerometers. Similarly, in hyperspectral imaging data (Sal, 2021) (tested in Numerical Simulations) the majority of pixels acquire information about multiple materials but only one has a dominant effect in the pixel value. Thus, the assumption that each sensing unit detects one source can hold approximately in several scenaria.

The sensing noise corresponds to thermal noise that is caused locally at each sensing unit by the sensing electronics equipment and it is not correlated with the source signals. Sensing units are different and placed in different spatial locations. Therefore, the thermal sensing noise caused by the individual sensing electronics at each unit can safely be assumed to be uncorrelated across different units see e.g., (Peng et al., 2020).

For simplicity in exposition, we consider temporarily an affine mapping $f_q(s) = \alpha_q \cdot s + \beta_q$ and uncorrelated sources $\{s_q(t)\}_{q=1}^{Q}$, then the data covariance matrix $\mathbf{\Sigma}_x := \mathbb{E}[(\mathbf{X}_t - \bar{\mathbf{X}})(\mathbf{X}_t - \bar{\mathbf{X}})^T]$, with $\mathbf{X}_t := [x_1(t), \ldots, x_P(t)]^T$ and $\bar{\mathbf{X}} := \mathbb{E}[\mathbf{X}_t]$ has entries ($\mathbb{E}$ denotes the expectation operator)

$$[\mathbf{\Sigma}_x]_{i,i'} = \mathbb{E}[(x_i(t) - \bar{x}_i)(x_{i'}(t) - \bar{x}_{i'})] = \alpha_q \alpha_{q'} \delta(q - q') \sigma_{s_q}^2 + \sigma_w^2 \delta(i - i'), \tag{2}$$

where $s_q(t)$ and $s_{q'}(t)$ denote the source signals sensed in measurements $x_i(t)$ and $x_{i'}(t)$ on sensing units $i$ and $i'$, $\sigma_w^2$ and $\sigma_{s_q}^2$ refer to the variances of $w_i(t)$ and $s_q(t)$ respectively, while $\delta(q-q')$ equals 1 if $q = q'$ and zero otherwise. Notice that if agents $i$ and $i'$ sense correlated source signals then the corresponding covariance entry in (2) will be nonzero, otherwise it will be equal to zero. It can be concluded that sensing units observing the same source signal $s_q(t)$ are correlated, hence by applying proper row and column exchanges to the covariance matrix $\mathbf{\Sigma}_x$ we can transform it into a block diagonal matrix having $Q$ diagonal blocks, with each block clustering together the sensing units observing the same source signal. It should be emphasized that affine mappings are not required by the novel framework; they just serve as an illustrative example to show the importance of a block diagonal structure and how diagonal blocks can be mapped to different groups of measurements with each group sensing the same source.

Clearly, the block diagonal structure of the data covariance matrix $\mathbf{\Sigma}_x$ can be used to cluster the sensed data according to their source content by using the indices of the diagonal blocks. Fig. 1 (left, lower center) indicate how measurements observing the same source, translate to a diagonal block in a properly defined similarity matrix (in the linear case $\mathbf{K}_x = \mathbf{\Sigma}_x$). Each of the covariance diagonal blocks, can be considered to be of rank one since sensed data from the same source-group (diagonal block) depend on the same source signal, while assuming that sensing noise has sufficiently low variance.

**Noise assumption:** Essentially the noise variance $\sigma_w^2$ should be much smaller than the variances $\sigma_{s_q}^2$ associated with source $s_q(t)$ for $q = 1, \ldots, Q$. This way each of the diagonal blocks is mainly affected by the source signal, while the noise contribution is negligible, resulting diagonal blocks of rank approximately equal to 1. There are several noise reduction techniques in signal processing, including adaptive filtering and Bayesian inference, that can effectively suppress noise variance and enhance the signal components of the sense data, see e.g., Vaseghi (2008). Thus, in the presence of $Q$ sources and an affine model in (1), the covariance matrix will contain $Q$, rank-1 diagonal blocks. A block diagonal similarity matrix will further facilitate data clustering by pertinent sparse

factorization of $\mathbf{K}_x$, where the nonzero entries of the recovered sparse factors $\mathbf{M}, \mathbf{N}$ [colored entries in Fig. 1 (right)] will reveal the location of the diagonal blocks and therefore the indices of units sensing the same source.

When the unknown mapping functions $f_q(\cdot)$ are non-linear in (1) and the source signals are not uncorrelated, then the resulting covariance matrix in (2) may not necessarily give rise to a block diagonal structure. To bypass these challenges, we will build on kernel-based data transformations to construct a pertinent kernel covariance matrix $\mathbf{K}_x$ which is as close as possible to i) having a block diagonal structure; and ii) having rank $Q$. Note that the covariance matrix in $\mathbf{\Sigma}_x$ in (2) is a special case of a kernel covariance matrix when using linear kernels, i.e., $\mathbf{K}_x = \mathbf{\Sigma}_x$. Depending on the sensed data, different kernel functions should be used to give rise to a block diagonal structure of rank $Q$ that will facilitate data clustering. We will construct an optimization framework that determines a proper kernel similarity matrix, via an optimal convex combination of kernels from a kernel dictionary of available mappings $\mathcal{D} := \{\mathbf{A}_x^1, \ldots, \mathbf{A}_x^B\}$ that lead to a desirable block diagonal kernel covariance structure [see Fig. 1 (upper center)]. The dictionary kernel matrices $\mathbf{A}_x^b$, are assumed to be pre-specified, and they can be evaluated from the available data using e.g., Gaussian, polynomial and other well-established kernel mappings (Ren and Sun, 2020; Ren et al., 2020).

### 2.1 Spatially distributed sensing agents

The sensing agents are spatially scattered in the observed field. A communication graph $\mathcal{G}$ is utilized to model the spatial configuration and connectivity of the units. Each agent corresponds to a graph node in set $\mathcal{V} := \{1, \ldots, P\}$, while the graph edges in set $\mathcal{E}$ correspond to active communication links. Connectivity in the graph is summarized by the adjacency matrix $\mathbf{A}$ whose $(i, j)$th entry is equal to 1 if units $i$ and $j$ communicate, otherwise is zero. The adjacency matrix is symmetric, while $\mathcal{N}_j$ is the single-hop neighborhood of unit $j$, i.e., the set of units which communicate directly (single-hop) with unit $j$. An example with $Q = 3$ and $P = 10$ is provided in Fig. 1. The communication

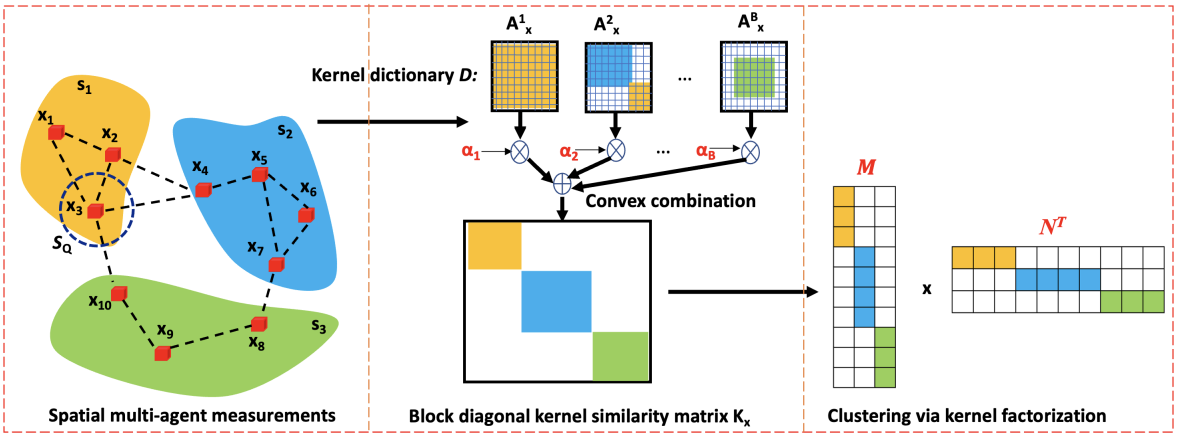


Figure 1: Joint kernel learning and clustering framework.

graph is assumed to be connected, thus there is a communication path (possibly having multiple edges) from one node to any other node.

**Connectivity assumption:** Further, it is assumed that there is at least one sensing unit $j$ whose single-hop neighborhood $\mathcal{N}_j$ consists of units with each of these units observing one source, but

when considering the set of these neighborhood observations $\{x_{j'}\}_{j' \in \mathcal{N}_j}$ they cover all the $Q$ sources present in the field., i.e., the measurements $\{x_i(t)\}_{i \in \mathcal{N}_j}$ in neighborhood $\mathcal{N}_j$ cover all source signals $\{s_q(t)\}_{q=1}^{Q}$; we denote this set of units as $\mathcal{S}_Q$. An example is provided in Fig. 1 (left) where $\mathcal{S}_Q$ contains unit 3 sensing $s_1$, though note that unit 3 has single-hop neighboring units 4 and 10 sensing sources $s_2$ and $s_3$, as well as units 1 and 2 sensing source $s_1$. Thus, the set of observations $x_3, x_1, x_2, x_{10}$ and $x_4$ cover all $Q = 3$ sources present. The latter assumption can be achieved using certain units that have transceivers with longer communication range to ensure these units can receive information for a number of neighboring units whose measurements contain information about all $Q$ sources. The objective is to allow each unit cluster their measurements according to their unknown source content by exchanging information only with their single-hop neighboring units in set $\mathcal{N}_j$. As we will explain in detail later (Sec. 4.3), satisfying this assumption does not require knowing the data clustering configuration.

## 3. SDP Kernel Learning and Clustering

The idea is to construct, using the available data, a kernel similarity matrix $\mathbf{K}_x$ that is block diagonal with a rank equal to $Q$, the number of underlying sources of interest. The diagonal blocks once identified will facilitate identifying the data clusters via proper sparse matrix factorization. A block diagonal matrix $\mathbf{K}_x$, with $Q$ diagonal blocks of rank 1 can be factorized as $\mathbf{K}_x = \mathbf{M}\mathbf{N}^T$, where the $P \times Q$ matrix factors $\mathbf{M}, \mathbf{N}$ have at most one nonzero entry across each of their rows [see Fig. 1 (right)]. A pertinent distance measure quantifying how far $\mathbf{K}_x$ is from a block diagonal structure is the following (Malhotra and Schizas, 2020)

$$\begin{aligned} F(\mathbf{K}_x) = \|\mathbf{K}_x - \mathbf{M}\mathbf{N}^T\|_F^2 + \nu \textstyle\sum_{i=1}^{P} [\|\mathbf{M}_{i,:}\|_1 - \|\mathbf{M}_{i,:}\|_2] \\ + \nu \textstyle\sum_{i=1}^{P} [\|\mathbf{N}_{i,:}\|_1 - \|\mathbf{N}_{i,:}\|_2] + \xi \|\mathbf{M} - \mathbf{N}\|_F^2, \end{aligned} \tag{3}$$

where $||\cdot||_1$ and $||\cdot||_2$ are the $\ell_1$ and $\ell_2$ norms of a vector, while $\nu, \xi$ are nonnegative regularization parameters controlling the sparsity of the rows of $\mathbf{M}, \mathbf{N}$, and similarity of factors $\mathbf{M}$ and $\mathbf{N}$, respectively. $\mathbf{M}_{i,:}, \mathbf{N}_{i,:}$ refer to the $i$th row of factors $\mathbf{M}$ and $\mathbf{N}$, respectively. Penalty term $||\mathbf{M}_{i,:}||_1 - ||\mathbf{M}_{i,:}||_2 \geq 0$ will be zero only when $\mathbf{M}_{i,:}$ has one nonzero entry or is equal to an all-zeroes vector. The $\ell_1 - \ell_2$ term has proven advantages over many of the other sparsity metrics; (see e.g., Yin et al., 2015). Note that function $F(\mathbf{K}_x) \geq 0$, attains its lowest value $F(\mathbf{K}_x) = 0$ if $\mathbf{K}_x$ has a block diagonal structure with each diagonal block having rank one, or $\mathbf{K}_x$ has rank $Q$. In these two aforementioned cases $||\mathbf{M}_{i,:}||_1 = ||\mathbf{M}_{i,:}||_2$ (and $||\mathbf{N}_{i,:}||_1 = ||\mathbf{N}_{i,:}||_2$), i.e., the ith row $\mathbf{M}_{i,:}$ will have at most one nonzero entry.

A good similarity matrix $\mathbf{K}_x$ is to have rank equal to $Q > 1$, where $Q$ corresponds to the number of diagonal blocks that need to be present in $\mathbf{K}_x$ equal to the $Q$ different groups of units sensing a different source. Our idea is to induce rank $Q$ by imposing constraints on the magnitude of the $Q$ largest eigenvalues of $\mathbf{K}_x$. Thus, it is important to ensure that the matrix has $Q$ strong eigenvalues. To this end, we propose maximizing the $Q$th largest eigenvalue, while introducing a mechanism that penalizes matrices of rank exceeding $Q$.

We resort to semidefinite programming (SDP) and linear matrix inequalities (LMIs) (Boyd and Vandenberghe, 2004) to enforce the aforementioned requirements. Further, the formulation in (3) is not amenable to distributed implementations. SDP will provide an effective formulation that can be further separated in local tasks across the sensing units. The following minimization formulation is

employed

$$
\begin{aligned}
&\arg\min -\mu \cdot w + v \cdot \textstyle\sum_{\ell=1}^{P} \left[\|\mathbf{M}_{\ell,:}\|_1 - \|\mathbf{M}_{\ell,:}\|_2\right] \\
&\qquad\qquad + v \cdot \textstyle\sum_{\ell=1}^{P} \left[\|\mathbf{N}_{\ell,:}\|_1 - \|\mathbf{N}_{\ell,:}\|_2\right] + \omega \cdot \psi + \xi \cdot \theta \\
&\text{s. to} \begin{bmatrix} \mathbf{I}_Q & \mathbf{M}^T \\ \mathbf{M} & \sum_{b=1}^{M} \alpha_b \mathbf{A}_x^b \end{bmatrix} \succeq 0, \begin{bmatrix} \frac{\theta}{\sqrt{Q}} \mathbf{I}_P & \mathbf{M} - \mathbf{N} \\ (\mathbf{M} - \mathbf{N})^T & \frac{1}{\sqrt{Q}} \mathbf{I}_Q \end{bmatrix} \succeq 0 \\
&\begin{bmatrix} \frac{\psi}{\sqrt{P}} \mathbf{I}_P & \sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b - \mathbf{M} \cdot \mathbf{N}^T \\ \left(\sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b - \mathbf{M} \cdot \mathbf{N}^T\right)^T & \frac{1}{\sqrt{P}} \mathbf{I}_P \end{bmatrix} \succeq 0 \\
&0.5 \cdot (\mathbf{N}^T \cdot \mathbf{M} + \mathbf{M}^T \cdot \mathbf{N}) \succeq w \cdot \mathbf{I}_Q, \quad w \geq 0, \\
&\{\alpha_b \geq 0\}_{b=1}^{B} \text{ and } \textstyle\sum_b \alpha_b = 1,
\end{aligned} \tag{4}
$$

where $\succeq$ corresponds to a matrix inequality, whereas $\geq$ indicates entry-wise inequality. The associated optimization variables are the $P \times Q$ matrix factors $\mathbf{M}, \mathbf{N}$, and the scalar variables $\boldsymbol{\alpha} := \{\alpha_b\}_{b=1}^{B}$, $w$, $\psi$ and $\theta$; $\mathbf{I}_Q$ denotes an identity matrix of size $Q \times Q$. The objective of the minimization formulation in (4) is to determine a proper convex combination $\sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b$ [see Fig. 1 (center)] of kernel covariances matrices available in a dictionary $\mathcal{D} := \{\mathbf{A}_x^b\}_{b=1}^{B}$ to minimize the block diagonal metric in (3). The penalty terms $\mu, v, \omega, \xi > 0$ are user-set.

Note that the kernel similarity matrix $\mathbf{K}_x$ is set as the convex combination, $\mathbf{K}_x := \sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b$, of the kernel matrices available in the dictionary $\mathcal{D}$. To satisfy the convexity of the $\mathbf{K}_x := \sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b$, the last two constraints in (4), forming a simplex, are employed. The first three matrix inequalities in (4) are equivalent (using the Schur complement; details are provided in Apdx. A) to

$$
\sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b \succeq \mathbf{M} \cdot \mathbf{M}^T, \ \|\mathbf{M} - \mathbf{N}\|_F^2 \leq \theta, \ \left\| \sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b - \mathbf{M} \cdot \mathbf{N}^T \right\|_F^2 \leq \psi,
$$

respectively. Thus, the second and third inequalitiers in (4) combined with the second and third terms in the cost in (4) ensure the minimization of the distance metric in (3).

Further, the fourth LMI in (4) combined with the first term in the cost of (4) ensure that both $\mathbf{M}$ and $\mathbf{N}$ are full-column rank equal to $Q$. Lastly, the first LMI in (4) is equivalent to $\sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b \succeq \mathbf{M}\mathbf{M}^T$ which combined with the fourth inequality in (4) ensure that $\sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b$ will also be of rank equal to $Q$ (details in Apdx. A). Thus, it is established (Apdx. A) that the formulation in (4) promotes the selection of dictionary kernel covariances that give rise to a block diagonal kernel $\sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b$ of rank $Q$, as long as the dictionary has kernel members supporting that.

**Proposition 1** *The formulation in* (4) *has an optimal solution involving coefficients* $\{\alpha_b^*\}$ *that result a kernel covariance matrix* $\sum_{b=1}^{B} \alpha_b^* \mathbf{A}_x^b$ *which is block diagonal and has rank* $Q$*, as long as the kernel dictionary* $\mathcal{D} := \{\mathbf{A}_x^b\}_{b=1}^{B}$ *contains a convex linear combination of its kernel elements which is block diagonal and has rank* $Q$*.*

**Remark:** It should be emphasized that the dictionary elements $\mathbf{A}_x^b$ do not have to be block diagonal and the size of the blocks does not have to be known. Only the number of sources $Q$ is assumed to be known in the present work. The proposed formulation in (4) (and related algorithms later on) aim to construct a convex combination $\mathbf{K}_x := \sum_{b=1}^{B} \alpha_b \mathbf{A}_x^b$ using the elements of dictionary $\mathcal{D}$ to construct a $\mathbf{K}_x$ as close to a block diagonal as possible.

### 3.1 Block Coordinate Descent and Difference of Convex Form

The formulation in (4) is nonconvex hindering the utilization of efficient optimization techniques. The main reasons are: i) the cost function contains concave terms, namely $-\|\mathbf{M}_{\ell,:}\|_2$ and $-\|\mathbf{N}_{\ell,:}\|_2$ which give rise to a difference of convex functions cost; and ii) several of the matrix inequality constraints contain the nonlinear terms $\mathbf{M}\cdot\mathbf{N}^T$ or $\mathbf{N}^T\cdot\mathbf{M}$ resulting nonconvex matrix inequality constraints.

To work around these challenges we resort i) to a difference of convex formulation; and ii) a block coordinate descent framework where we optimize with respect to $\mathbf{M}, \boldsymbol{\alpha}, w, \psi, \theta$ while keeping fixed the factor $\mathbf{N}$ to its more recent update during iteration $\tau$, namely $\mathbf{N}_\tau$. In detail, when updating $\mathbf{M}, \boldsymbol{\alpha}$ the nonlinear terms $\mathbf{N}^T\cdot\mathbf{M}$ and $\mathbf{M}\cdot\mathbf{N}^T$ will be replaced with $\mathbf{N}_\tau\cdot\mathbf{M}^T$ and $\mathbf{M}\cdot\mathbf{N}_\tau^T$ using the most recent update $\mathbf{N}_\tau$; while when updating $\mathbf{N}$ the nonlinear terms $\mathbf{M}\cdot\mathbf{N}^T$ and $\mathbf{N}^T\cdot\mathbf{M}$ will be replaced with $\mathbf{M}_{\tau+1}\cdot\mathbf{N}^T$ and $\mathbf{N}^T\cdot\mathbf{M}_{\tau+1}$ using update $\mathbf{M}_{\tau+1}$.

Notice that the cost in (4) contains the terms $\sum_{\ell=1}^P[\|\mathbf{M}_{\ell,:}\|_1-\|\mathbf{M}_{\ell,:}\|_2]$ and $\sum_{\ell=1}^P[\|\mathbf{N}_{\ell,:}\|_1 -\|\mathbf{N}_{\ell,:}\|_2]$. If we set $H(\mathbf{M})=\sum_{\ell=1}^P\|\mathbf{M}_{\ell,:}\|_1$ and $G(\mathbf{M})=\sum_{\ell=1}^P\|\mathbf{M}_{\ell,:}\|_2$ then the second and third terms in (4) is essentially the difference of two convex functions, i.e, $H(\mathbf{M})-G(\mathbf{M})$. To this end, we will resort to the difference of convex functions approach (Tao and An, 1997). The algorithm iteratively computes an affine majorization of the function $-\mathbf{G}(\mathbf{M})$, where the majorization during the $\tau$th iterate is given as, trace $\left(\mathbf{M}^T, \left\{\frac{\partial-\mathbf{G}(\mathbf{M})}{\partial\mathbf{M}}|_{\mathbf{M}=\mathbf{M}_\tau}\right\}\right)$.

The cost in (4) can be numerically minimized by utilizing block coordinate descent (Tseng, 2001) along with the difference of convex functions recursive approach. During iteration $\tau+1$ the proposed algorithm involves the following steps

**Step 1a:** Fix factor $\mathbf{N}$ to most recent update $\mathbf{N}_\tau$ and minimize (4) with respect to (wrt) the rest of the variables. The gradient of $\|\mathbf{M}_{\ell,:}\|_2$ is evaluated at $\mathbf{M}_{\tau,\kappa}$ which is obtained during difference of convex algorithm (DCA) iteration $\kappa$

$$\mathbf{m}_\ell^{\tau,\kappa}\in\partial\|\mathbf{M}_{\ell,:}\|_2\,\big|_{\mathbf{M}=\mathbf{M}_{\tau,\kappa}}=\|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2{}^{-1}\cdot\mathbf{M}_{\tau,\kappa,\ell,:}^T, \tag{5}$$

where $\mathbf{M}_{\tau,\kappa,\ell,:}^T$ is the $\ell$th row of update $\mathbf{M}_{\tau,\kappa}$.

**Step 1b:** Solve the majorized version of (4) after replacing $\mathbf{N}=\mathbf{N}_\tau$, and using the gradient in (5) to linearize the concave terms $-\|\mathbf{M}_{\ell,:}\|_2$, i.e. we obtain the SDP,

$$\begin{aligned}
&\{\mathbf{M}_{\tau,\kappa+1},\boldsymbol{\alpha}_{\tau,\kappa+1},w_{\tau,\kappa+1},\psi_{\tau,\kappa+1},\theta_{\tau,\kappa+1}\}\in\arg\min -\mu\cdot w\\
&\qquad\qquad+\omega\cdot\psi+\xi\cdot\theta+v\cdot\textstyle\sum_{\ell=1}^P\left[\|\mathbf{M}_{\ell,:}\|_1-\frac{\mathbf{M}_{\tau,\kappa,\ell,:}}{\|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2}\mathbf{M}_{\ell,:}^T\right]\\
&\textbf{s. to}\begin{bmatrix}\mathbf{I}_Q & \mathbf{M}^T\\ \mathbf{M} & \sum_{b=1}^B\alpha_b\mathbf{A}_x^b\end{bmatrix}\succeq 0,\ \begin{bmatrix}\frac{\theta}{\sqrt{Q}}\mathbf{I}_P & \mathbf{M}-\mathbf{N}_\tau\\ (\mathbf{M}-\mathbf{N}_\tau)^T & \frac{1}{\sqrt{Q}}\mathbf{I}_Q\end{bmatrix}\succeq 0\\
&\begin{bmatrix}\frac{\psi}{\sqrt{P}}\mathbf{I}_P & \sum_{b=1}^B\alpha_b\mathbf{A}_x^b-\mathbf{M}\cdot\mathbf{N}_\tau^T\\ \left(\sum_{b=1}^B\alpha_b\mathbf{A}_x^b-\mathbf{M}\cdot\mathbf{N}_\tau^T\right)^T & \frac{1}{\sqrt{P}}\mathbf{I}_P\end{bmatrix}\succeq 0\\
&0.5\cdot(\mathbf{N}_\tau^T\cdot\mathbf{M}+\mathbf{M}^T\cdot\mathbf{N}_\tau)\succeq w\cdot\mathbf{I}_Q,\ \ w\geq 0,\\
&\{\alpha_b\geq 0\}_{b=1}^B\text{ and }\textstyle\sum_b\alpha_b=1.
\end{aligned}\tag{6}$$

Steps 1a and 1b are repeated until $\mathbf{M}_{\tau,\kappa+1},\boldsymbol{\alpha}_{\tau,\kappa+1}$ converge (as will be established in Prop. 3), which algorithmically will be verified by checking when the update amounts $\|\mathbf{M}_{\tau,\kappa+1}-\mathbf{M}_{\tau,\kappa}\|_F$

and $\|\boldsymbol{\alpha}_{\tau,\kappa+1} - \boldsymbol{\alpha}_{\tau,\kappa}\|_F$ drop below a user-set threshold $\epsilon_1$. We denote the converging entities as $\mathbf{M}_{\tau+1}$ and $\boldsymbol{\alpha}_{\tau+1}$ which will be fixed next in (4) to update the factor $\mathbf{N}$ using the following two steps.

**Step 2a:** Fix factor $\mathbf{M}$ and $\boldsymbol{\alpha}$ to the most recent update $\mathbf{M}_{\tau+1}$ and $\boldsymbol{\alpha}_{\tau+1}$ respectively, and minimize (4) with respect to the rest of the variables. To this end, we calculate the gradient of $\|\mathbf{N}_{\ell,:}\|_2$ evaluated at $\mathbf{N}_{\tau,\kappa}$ that is obtained during DCA iteration $\kappa = 0, 1, 2, \ldots,$

$$\mathbf{n}_{\ell}^{\tau,\kappa} \in \partial\|\mathbf{N}_{\ell,:}\|_2 \,\big|_{\mathbf{N}=\mathbf{N}_{\tau,\kappa}} = \|\mathbf{N}_{\tau,\kappa,\ell,:}\|_2{}^{-1} \cdot \mathbf{N}_{\tau,\kappa,\ell,:}^T. \tag{7}$$

**Step 2b:** Solve the majorized version of (4) after replacing $\mathbf{M} = \mathbf{M}_{\tau+1}$ and $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{\tau+1}$, and using the gradient in (7) to linearize the concave terms $-\|\mathbf{N}_{\ell,:}\|_2$ for $\ell = 1, \ldots, P$, i.e.,

$$\begin{aligned}
&\{\mathbf{N}_{\tau,\kappa+1}, w_{\tau,\kappa+1}, \psi_{\tau,\kappa+1}, \theta_{\tau,\kappa+1}\} \in \arg\min -\mu \cdot w + \omega \cdot \psi \\
&\qquad + \xi \cdot \theta + v \cdot \textstyle\sum_{\ell=1}^{P} \left[\|\mathbf{N}_{\ell,:}\|_1 - \frac{\mathbf{N}_{\tau,\kappa,\ell,:}}{\|\mathbf{N}_{\tau,\kappa,\ell,:}\|_2}\mathbf{N}_{\ell,:}^T\right] \\
&\textbf{s. to} \begin{bmatrix} \frac{\theta}{\sqrt{Q}}\mathbf{I}_P & \mathbf{M}_{\tau+1} - \mathbf{N} \\ (\mathbf{M}_{\tau+1} - \mathbf{N})^T & \frac{1}{\sqrt{Q}}\mathbf{I}_Q \end{bmatrix} \succeq 0 \\
&\begin{bmatrix} \frac{\psi}{\sqrt{P}}\mathbf{I}_P & \sum_{b=1}^{B} \alpha_{\tau+1,b}\mathbf{A}_x^b - \mathbf{M}_{\tau+1} \cdot \mathbf{N}^T \\ \left(\sum_{b=1}^{B} \alpha_{\tau+1,b}\mathbf{A}_x^b - \mathbf{M}_{\tau+1} \cdot \mathbf{N}^T\right)^T & \frac{1}{\sqrt{P}}\mathbf{I}_P \end{bmatrix} \succeq 0 \\
&0.5 \cdot (\mathbf{N}^T \cdot \mathbf{M}_{\tau+1} + \mathbf{M}_{\tau+1}^T \cdot \mathbf{N}) \succeq w \cdot \mathbf{I}_Q, \;\; w \geq 0.
\end{aligned} \tag{8}$$

Steps 2a and 2b are repeated until $\mathbf{N}_{\tau,\kappa+1}$ converges (as will be established in Prop. 3), which algorithmically will be verified by checking when the update amount $\|\mathbf{N}_{\tau,\kappa+1} - \mathbf{N}_{\tau,\kappa}\|_F$ drops below a user-set threshold $\epsilon_1$. We denote the converging factor as $\mathbf{N}_{\tau+1}$. The recursive method for numerically solving (4) is tabulated in detail as Alg. 1. Notice that a similar breaking condition is also utilized to terminate the outer block coordinate descent loop (with iteration index $\tau$) using the user-defined threshold $\epsilon_2$. $K_{max}$ denotes the user-defined maximum number of DCA iterations employed during coordinate descent iteration $\tau$, in case it takes too long for the breaking conditions in lines 7 or 15 of Alg. 1 to be satisfied. We set this value equal to $K_{max} = 20$ for the numerical tests conducted in the paper ensuring convergence.

### 3.2 Convergence

Next we demonstrate that Alg. 1 returns updates $\{\mathbf{M}_\tau, \boldsymbol{\alpha}_\tau, w_\tau, \psi_\tau, \theta_\tau\}$ that result a non-increasing sequence of cost values in (4). First, we establish in Apdx. B that

**Lemma 2** *The cost functions in* (4)*,* (6) *and* (8) *are bounded below by a finite negative number. A negative value of the cost in* (4)*,* (6) *and* (8) *implies that factor* $\mathbf{M}_\tau$ *has full rank* $Q$*.*

Using Lemma 2 it is further established in Apdx. C that

**Proposition 3** *The iterates* $\{\mathbf{M}_\tau, \mathbf{N}_\tau, \boldsymbol{\alpha}_\tau, w_\tau, \psi_\tau, \theta_\tau\}$ *produced by Alg. 1 result a non-increasing cost function value sequence* $J_\tau(\cdot)$ *in* (4) *which converges to a finite value as* $\tau \to \infty$*, i.e.,*

$$\lim_{\tau\to\infty} J_\tau(\mathbf{M}_\tau, \boldsymbol{\alpha}_\tau, w_\tau, \psi_\tau) = J < \infty. \tag{9}$$

*Therefore using continuity of the cost in* (4) *the iterates* $\{\mathbf{M}_\tau, \mathbf{N}_\tau, \boldsymbol{\alpha}_\tau, w_\tau, \psi_\tau, \theta_\tau\}$ *converge too.*

**Algorithm 1** Centralized Joint Kernel Selection and Clustering (CKC)

1: Initialize $\mathbf{N}_{0,0}$ randomly, and kernel weights $\boldsymbol{\alpha}_{0,0,j} = \frac{1}{B} \forall j \in 1, ..., B$. Set penalty coefficients $\omega, \xi, \mu, v$.
2: The dictionary kernel matrices $\{\mathbf{A}_x^b\}_{b=1}^B$ are normalized to have unit trace.
3: **for** $\tau = 0, 1, 2...$ **do**
4:  **for** $\kappa = 0, 1, 2...K_{max} - 1$ **do**
5:  Form the gradients $\{\mathbf{m}_\ell^{\tau,\kappa}\}_{\ell=1}^P$ using (5).
6:  Solve SDP formulation in (6) to obtain updates $\{\mathbf{M}_{\tau,\kappa+1}, \boldsymbol{\alpha}_{\tau,\kappa+1}, w_{\tau,\kappa+1}, \psi_{\tau,\kappa+1}, \theta_{\tau,\kappa+1}\}$. This can be done using interior point methods, (e.g., Boyd and Vandenberghe, 2004; Grant and Boyd, 2014).
7:  **if** $(\|\mathbf{M}_{\tau,\kappa+1} - \mathbf{M}_{\tau,\kappa}\|_F + \|\boldsymbol{\alpha}_{\tau,\kappa+1} - \boldsymbol{\alpha}_{\tau,\kappa}\|_F < \epsilon_1)$ **then**
8:  $\mathbf{M}_{\tau+1} = \mathbf{M}_{\tau,\kappa+1}$ and $\boldsymbol{\alpha}_\tau = \boldsymbol{\alpha}_{\tau,\kappa+1}$.
9:  Break.
10:  **end if**
11:  **end for**
12:  **for** $\kappa = 0, 1, 2...K_{max} - 1$ **do**
13:  Form the gradients $\{\mathbf{n}_\ell^{\tau,\kappa}\}_{\ell=1}^P$ using (7).
14:  Solve SDP formulation in (8) to obtain updates $\{\mathbf{N}_{\tau,\kappa+1}, w_{\tau,\kappa+1}, \psi_{\tau,\kappa+1}, \theta_{\tau,\kappa+1}\}$. This can be done using interior point methods.
15:  **if** $(\|\mathbf{N}_{\tau,\kappa+1} - \mathbf{N}_{\tau,\kappa}\|_F < \epsilon_1)$ **then**
16:  $\mathbf{N}_{\tau+1} = \mathbf{N}_{\tau,\kappa+1}$.
17:  Break.
18:  **end if**
19:  **end for**
20:  **if** $\|\mathbf{M}_{\tau+1} - \mathbf{M}_\tau\|_F + \|\mathbf{N}_{\tau+1} - \mathbf{N}_\tau\|_F + \|\boldsymbol{\alpha}_{\tau+1} - \boldsymbol{\alpha}_\tau\|_F < \epsilon_2$ **then**
21:  Break.
22:  **end if**
23: **end for**

**Remarks:** Lemma 2 and Prop. 3 demonstrate that if the limit value $J$, to which iterates $J_\tau(\cdot)$ converge to, is negative then the iterates $\mathbf{M}_\tau$ will be full rank $Q$. For sufficiently large $\mu$ the updates $\omega_{\tau,\kappa}$ for sufficiently large $\tau$ should be strictly positive which combined with the LMI $0.5 \cdot (\mathbf{M}^T\mathbf{N} + \mathbf{N}^T\mathbf{M}) \succeq w \cdot \mathbf{I}_Q$ will return factors $\mathbf{M}_\tau, \mathbf{N}_\tau$ that have rank equal to $Q$ (see also proof in Apdx. B). Further, the iterates $\psi_{\tau,\kappa}$ are pushed towards zero as much as possible, and since $\|\sum_b \alpha_{\tau,\kappa+1,b}\mathbf{A}_x^b - \mathbf{M}_{\tau,\kappa+1}\mathbf{N}_\tau^T\|_F^2 \leq \psi_{\tau,\kappa+1}$ that further implies that the iterates $\alpha_{\tau,\kappa+1,b}$ are selected such that $\sum_b \alpha_{\tau,\kappa+1,b}\mathbf{A}_x^b$ get as close as possible to $\mathbf{M}_{\tau,\kappa+1}\mathbf{N}_\tau^T$ that has rank $Q$, while from the first LMI in (4) $\sum_b \alpha_{\tau,\kappa+1,b}\mathbf{A}_x^b$ has rank at least equal to $Q$. Finally, the terms $\|\mathbf{M}_{\tau,\kappa+1,\ell,:}\|_1 - \frac{\mathbf{M}_{\tau,\kappa,\ell,:}}{\|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2}(\mathbf{M}_{\tau,\kappa+1,\ell,:})^T$ in (6) can only attain the lowest value of zero only if the $\ell$th row $\mathbf{M}_{\tau,\kappa+1,\ell,:}$ contains at most one nonzero entry. Similar arguments can be applied for row updates $\mathbf{N}_{\tau,\kappa+1,\ell,:}$. Thus, $\mathbf{M}_{\tau+1}, \mathbf{N}_{\tau+1}$ factor iterates are pushed towards a block diagonal structure. These convergence claims hold as long as $K_{max}$ is sufficiently large to ensure that the breaking conditions on lines 7 and/or 15 in Alg.1 are met. Although, an analytical lower bound could not be obtained for $K_{max}$, a value of 20 meets the requirements for the numerical tests considered in this work.

## 4. Distributed Kernel-Based Clustering

The SDP formulations in (6) and (8) require storage of the dictionary kernels $\mathbf{A}_x^b$ and optimization variables in one central location. Numerically solving (6) or (8) via interior point methods, (e.g., Boyd and Vandenberghe, 2004; Grant and Boyd, 2014), has a complexity of $\mathcal{O}(P^4)$ where $P$ is the number of agents.

In the network setting considered in Sec. 2 each agent $j$ acquires a measurement $x_j(t)$ and can directly exchange information with its single-hop neighbors. The objective is to derive from (6) and (8) a separable minimization formulations that can be solved in a distributed fashion by solving small scale local SDP problems across the sensing agents and exchanging information with single-hop neighbors. Such a separable formulation will enable the implementation of joint kernel learning and clustering in ad hoc distributed settings.

To this end, we introduce local optimization variables that will help reformulate (6) and (8) in a separable fashion. At sensing unit $j$, let $\boldsymbol{\alpha}^j := \{\alpha_b^j\}_{b=1}^B$ denote the local version of the kernel coefficient variables $\{\alpha_b\}_{b=1}^B$. Similarly $\psi_j$, $w_j$ and $\theta_j$ are local replicas of the variables $\psi$, $w$ and $\theta$, respectively. Further, we introduce the local $|\mathcal{N}_j| \times Q$ factor matrix

$$\mathbf{M}_{\mathcal{N}_j} := \left[(\mathbf{M}_{j,:}^j)^T, (\mathbf{M}_{j',:}^j)^T, \ldots, (\mathbf{M}_{j'',:}^j)^T\right]^T, \tag{10}$$

where the indices $j, j', j''$ belong to $\mathcal{N}_j$. Thus, $\mathbf{M}_{\mathcal{N}_j}$ contains a local (at unit $j$) version $\mathbf{M}_{j',:}^j$ of the row $\mathbf{M}_{j',:}$ in central $P \times Q$ factor $\mathbf{M}$ in (6) for $j' \in \mathcal{N}_j$. Similarly, we can define a local version $\mathbf{N}_{\mathcal{N}_j}$ for factor $\mathbf{N}$ stored at agent $j$. Unit $j$ can exchange information directly with other units $j' \in \mathcal{N}_j$ in its single-hop neighborhood, therefore it can directly calculate the entries of dictionary kernels $\mathbf{A}_x^b(j', j'')$ for $j', j'' \in j \cup \mathcal{N}_j$ which form a $(|\mathcal{N}_j| + 1) \times (|\mathcal{N}_j| + 1)$ submatrix of $\mathbf{A}_x^b$ denoted $[\mathbf{A}_x^b]_{\mathcal{N}_j}$ [see Fig. 2]. All units in the distributed setting utilize the same pre-determined dictionary $\mathcal{D} = \{\mathbf{A}_x^b\}_{b=1}^B$. This can be hardwired in the local sensing units. The only task that local units need to perform with regard to the kernel elements in $\mathcal{D}$ is to extract the local submatrices $[\mathbf{A}_x^b]_{\mathcal{N}_j}$, which can be formed directly after keeping the rows and columns of $\mathbf{A}_x^b$ with indices in neighborhood $\mathcal{N}_j$.

Similarly, at unit $j$ only the entries with row and column indices from $\{j\} \cup \mathcal{N}_j$ can be evaluated directly via single-hop communications in $\mathbf{H}_\tau^1 := \mathbf{M} \cdot \mathbf{N}_\tau^T$ and $\mathbf{H}_\tau^2 := 0.5 \cdot (\mathbf{N}_\tau^T \cdot \mathbf{M} + \mathbf{M}^T \cdot \mathbf{N}_\tau)$ on the left hand side (lhs) of the third and fourth LMIs in (6), respectively. We denote this $(|\mathcal{N}_j| + 1) \times (|\mathcal{N}_j| + 1)$ local submatrices of $\mathbf{H}_\tau^1$ and $\mathbf{H}_\tau^2$, which are calculated locally at unit $j$ as

$$\begin{aligned} [\mathbf{H}_\tau^1(\mathbf{N}_\tau)]_{\mathcal{N}_j} &= [\mathbf{M} \cdot (\mathbf{N}_\tau)^T]_{\mathcal{N}_j} \\ [\mathbf{H}_\tau^2(\mathbf{N}_\tau)]_{\mathcal{N}_j} &= 0.5 \cdot \textstyle\sum_{j' \in \mathcal{N}_j \cup \{j\}} [(\mathbf{N}_{\tau,j',:}^j)^T \mathbf{M}_{j',:}^j + (\mathbf{M}_{j',:}^j)^T \mathbf{N}_{\tau,j',:}^j], \end{aligned} \tag{11}$$

where $\mathbf{N}_{\tau,j',:}^j$ a local version of row $j'$ of factor update $\mathbf{N}_\tau$ contained in $[\mathbf{N}_\tau]_{\mathcal{N}_j}$ at unit $j$.

### 4.1 Separable SDP Formulation

The first two LMI constraints in (6) involve the central factor variables $\mathbf{M}, \mathbf{N}_\tau$, as well as all the entries of dictionary kernels $\{\mathbf{A}_x^b\}$ which are not available in a single location in the distributed setting considered here. We substitute these two LMI constraints with the following set of local

LMIs

$$\mathbf{G}_{1,j} := \begin{bmatrix} \mathbf{I}_Q & (\mathbf{M}_{\mathcal{N}_j})^T \\ \mathbf{M}_{\mathcal{N}_j} & \sum_{b=1}^{B} \alpha_b^j [\mathbf{A}_x^b]_{\mathcal{N}_j} \end{bmatrix} \succeq 0, \tag{12}$$

$$\mathbf{G}_{\tau,2,j}(\mathbf{N}_\tau) := \begin{bmatrix} \frac{\theta_j}{\sqrt{Q}} \mathbf{I}_{|\mathcal{N}_j|+1} & [\mathbf{M} - \mathbf{N}_\tau]_{\mathcal{N}_j} \\ ([\mathbf{M} - \mathbf{N}_\tau]_{\mathcal{N}_j})^T & \frac{1}{\sqrt{Q}} \mathbf{I}_Q \end{bmatrix} \succeq 0 \tag{13}$$

for $j = 1, \ldots, P$ which utilizes locally available entities $\mathbf{M}_{\mathcal{N}_j}$, $[\mathbf{N}_\tau]_{\mathcal{N}_j}$ and $[\mathbf{A}_x^b]_{\mathcal{N}_j}$ and using the Schur complement ensures $\sum_{b=1}^{B} \alpha_b^j [\mathbf{A}_x^b]_{\mathcal{N}_j} \succeq \mathbf{M}_{\mathcal{N}_j} \mathbf{M}_{\mathcal{N}_j}^T$ and $\|[\mathbf{M} - \mathbf{N}_\tau]_{\mathcal{N}_j}\|_F^2 \leq \theta_j$. Using similar reasoning the third LMI in (6) is replaced with the following set of local LMI constraints for $j = 1, \ldots, P$

$$\mathbf{G}_{\tau,3,j}(\mathbf{N}_\tau) := \begin{bmatrix} \frac{\psi_j}{\sqrt{|\mathcal{N}_j|+1}} \mathbf{I}_{|\mathcal{N}_j|+1}, & \sum_{b=1}^{B} \alpha_b^j [\mathbf{A}_x^b]_{\mathcal{N}_j} - [\mathbf{H}_\tau^1(\mathbf{N}_\tau)]_{\mathcal{N}_j} \\ \left( \sum_{b=1}^{B} \alpha_b^j [\mathbf{A}_x^b]_{\mathcal{N}_j} - [\mathbf{H}_\tau^1(\mathbf{N}_\tau)]_{\mathcal{N}_j} \right)^T, & \frac{1}{\sqrt{|\mathcal{N}_j|+1}} \mathbf{I}_{|\mathcal{N}_j|+1} \end{bmatrix} \succeq 0, \tag{14}$$

which via the Schur complement is equivalent to $\| \sum_{b=1}^{B} \alpha_b^j [\mathbf{A}_x^b]_{\mathcal{N}_j} - [\mathbf{H}_\tau^1(\mathbf{N}_\tau)]_{\mathcal{N}_j} \|_F^2 \leq \psi_j$. Similarly the fourth LMI is replaced with $[\mathbf{H}_\tau^2(\mathbf{N}_\tau)]_{\mathcal{N}_j} \succeq w_j$ for $j \in \mathcal{S}_Q$ with $\mathcal{S}_Q$ denoting the set of those units $j$ whose neighbors in $\mathcal{N}_j$ sense all $Q$ sources. The entry-wise constraint (in)-equalities are substituted with the local versions $w_j \geq 0$, $\{\alpha_b^j \geq 0\}$, $\sum_b \alpha_b^j = 1$ for $j = 1, \ldots, P$.

Starting from the cost function in (6) we replace the global variables $w, \psi$ and $\theta$ with the the average of their local versions, i.e., $|\mathcal{S}_Q|^{-1} \sum_{j \in \mathcal{S}_Q} w_j$, $P^{-1} \sum_{j=1}^{P} \psi_j$ and $P^{-1} \sum_{j=1}^{P} \theta_j$, respectively, with $|\mathcal{S}_Q|$ denoting the cardinality of set $\mathcal{S}_Q$. In the summation term of cost (6) factors $\mathbf{M}_{\ell,:}$ are replaced with their local representation $\mathbf{M}_{\ell,:}^\ell$, while $\mathbf{M}_{\tau,\kappa,\ell,:}$ are replaced with local update $\mathbf{M}_{\tau,\kappa,\ell,:}^\ell$.

We also introduce constraints to ensure equality among the local versions of the variables representing the coefficients $\{\alpha_b\}_{b=1}^B$, among the local versions of the factor rows $\{\mathbf{M}_{j,:}\}_{j=1}^P$ and $\{\mathbf{N}_{j,:}\}_{j=1}^P$, respectively. Auxiliary variables $\boldsymbol{\beta}_{j,j'} := \{\beta_b^{j,j'}\}_{b=1}^B$, and row vectors $\mathbf{Z}_{j,j'}$, $\boldsymbol{\Theta}_{j,j'}$ are introduced in the following local equality constraints:

$$\{\alpha_b^j = \beta_b^{j,j'}, \alpha_b^{j'} = \beta_b^{j,j'}\}_{b=1}^B, \text{ for } j = 1, \ldots, P, \ j' \in \mathcal{N}_j \tag{15}$$

$$\mathbf{M}_{j,:}^j = \mathbf{Z}_{j,j'}, \ \mathbf{M}_{j,:}^{j'} = \mathbf{Z}_{j,j'}, \text{ for } j' \in \mathcal{N}_j, \ j = 1, \ldots, P, \tag{16}$$

$$\mathbf{N}_{j,:}^j = \tilde{\mathbf{Z}}_{j,j'}, \ \mathbf{N}_{j,:}^{j'} = \tilde{\mathbf{Z}}_{j,j'}, \text{ for } j' \in \mathcal{N}_j, \ j = 1, \ldots, P. \tag{17}$$

The constraints in (15) result that $\alpha_b^j = \alpha_b^{j'}$ for $j' \in \mathcal{N}_j$ and $j = 1, \ldots, P$; since the communication graph of the sensing units is connected this further implies that $\alpha_b^1 = \alpha_b^2 = \ldots = \alpha_b^P$ for $b = 1, \ldots, P$. Thus, all the local coefficient variables will be identical since they all represent the centralized set of coefficients $\alpha_b$. Similarly, the equalities in (16) and (17) ensure that $\mathbf{M}_{j,:}^j = \mathbf{M}_{j,:}^{j'}$ and $\mathbf{N}_{j,:}^j = \mathbf{N}_{j,:}^{j'}$ for $j' \in \mathcal{N}_j$ and $j = 1, \ldots, P$ which combined with the communication graph connectivity will ensure that all local row vectors $\mathbf{M}_{j,:}^j$ will be identical for $j' \in \mathcal{N}_j$ since they represent the centralized row variable $\mathbf{M}_{j,:}$ for $j = 1, \ldots, P$ (similarly all local $\mathbf{N}_{j,:}^{j'}$ will be equal for $j' \in \mathcal{N}_j \cup \{j\}$ ). The variables $\boldsymbol{\beta}_b^{j,j'}$, $\mathbf{Z}_{j,j'}$ and $\tilde{\mathbf{Z}}_{j,j'}$ are utilized to facilitate distributed minimization

of (6) and (8) via the alternating direction method of multipliers and eventually there will be no need to update them separately in the recursive updates obtained, since they will be written as linear functions of other optimization variables been updated. The following separable formulation of (6) is obtained

$$\begin{aligned}
&\arg\min -\mu|\mathcal{S}_Q|^{-1}\cdot\textstyle\sum_{j\in\mathcal{S}_Q} w_j+\omega P^{-1}\sum_j \psi_j \qquad (18)\\
&\qquad\qquad +\xi P^{-1}\textstyle\sum_j \theta_j+v\cdot\sum_{\ell=1}^{P}\left[\|\mathbf{M}^{\ell}_{\ell,:}\|_1-\frac{\mathbf{M}^{\ell}_{\tau,\kappa,\ell,:}}{\|\mathbf{M}^{\ell}_{\tau,\kappa,\ell,:}\|_2}(\mathbf{M}^{\ell}_{\ell,:})^T\right],\\
&\text{s. to } \mathbf{G}_{1,j}\succeq 0, \mathbf{G}_{\tau,2,j}(\mathbf{N}_\tau)\succeq 0, \mathbf{G}_{\tau,3,j}(\mathbf{N}_\tau)\succeq 0,\\
&\{[\mathbf{H}^2_\tau(\mathbf{N}_\tau)]_{\mathcal{N}_j}\succeq w_j\cdot\mathbf{I}_Q\}_{j\in\mathcal{S}_Q},\ \ w_j\geq 0, \{\alpha^j_b\geq 0\}_{b=1}^B\\
&\textstyle\sum_b \alpha^j_b=1,\ \text{ for } j=1,\ldots,P,\\
&\{\alpha^j_b=\beta^{j,j'}_b\}_{b=1}^B, \{\alpha^{j'}_b=\beta^{j,j'}_b\}_{b=1}^B\\
&\mathbf{M}^j_{j,:}=\mathbf{Z}_{j,j'},\ \mathbf{M}^{j'}_{j,:}=\mathbf{Z}_{j,j'},\ \text{for } j'\in\mathcal{N}_j,\ j=1,\ldots,P.
\end{aligned}$$

Using similar steps, the SDP program in (8) is reformulated in the following separable formulation

$$\begin{aligned}
&\arg\min -\mu|\mathcal{S}_Q|^{-1}\textstyle\sum_{j\in\mathcal{S}_Q} w_j+\omega P^{-1}\cdot\sum_j \psi_j \qquad (19)\\
&\qquad\qquad +\xi P^{-1}\cdot\textstyle\sum_j \theta_j+v\cdot\sum_{\ell=1}^{P}\left[\|\mathbf{N}^{\ell}_{\ell,:}\|_1-\frac{\mathbf{N}^{\ell}_{\tau,\kappa,\ell,:}}{\|\mathbf{N}^{\ell}_{\tau,\kappa,\ell,:}\|_2}(\mathbf{N}^{\ell}_{\ell,:})^T\right],\\
&\text{s. to } \mathbf{G}_{\tau,2,j}(\mathbf{M}_{\tau+1})\succeq 0, \mathbf{G}_{\tau,3,j}(\mathbf{M}_{\tau+1})\succeq 0,\\
&\{[\mathbf{H}^2_\tau(\mathbf{M}_{\tau+1})]_{\mathcal{N}_j}\succeq w_j\cdot\mathbf{I}_Q\}_{j\in\mathcal{S}_Q},\ \ w_j\geq 0, \mathbf{N}^j_{j,:}=\mathbf{\Theta}_{j,j'},\ \mathbf{N}^{j'}_{j,:}=\mathbf{\Theta}_{j,j'},\ \text{for } j'\in\mathcal{N}_j,\ j=1,\ldots,P,
\end{aligned}$$

where $\mathbf{G}_{\tau,2,j}(\mathbf{M}_\tau), \mathbf{G}_{\tau,3,j}(\mathbf{M}_\tau), [\mathbf{H}^2_\tau(\mathbf{M}_\tau)]_{\mathcal{N}_j}$ have the same structure given in (11),(12) and (14) respectively, after replacing $\mathbf{M}$ with the most recent update $\mathbf{M}_{\tau+1}$ and $\mathbf{N}_\tau$ with $\mathbf{N}$ which is the main optimization variable in (8).

### 4.2 Separable vs. Centralized Formulation

The formulations in (18) and (19) are separable versions of (6) and (8) that will enable distributed minimization of the corresponding cost functions within a connected network of sensing units. These separable formulations will be an approximation of (6) and (8) due to the utilization of the local LMI constraints. From the equality constraints in the last two lines of (18) and (19) notice that $\mathbf{M}^{j'}_{j,:}=\mathbf{M}_{j,:}$, $\mathbf{N}^{j'}_{j,:}=\mathbf{N}_{j,:}$ and $\{\alpha^j_b=\alpha_b\}_{b=1}^B$ for all $j'\in\mathcal{N}_j\cup\{j\}$ and $j=1,\ldots,P$. The next result proved in Apdx. D delineates the relationship between the separable formulations in (18) and (19), and the centralized formulation in (6) and (8).

**Proposition 4** *The set of local LMIs* $[\mathbf{H}^2_\tau(\mathbf{N}_\tau)]_{\mathcal{N}_j}\succeq w_j\cdot\mathbf{I}_Q$ *and* $[\mathbf{H}^2_\tau(\mathbf{M}_{\tau+1})]_{\mathcal{N}_j}\succeq w_j\cdot\mathbf{I}_Q$ *for* $j\in\mathcal{S}_Q$ *in* (18) *and* (19) *respectively, guarantee that their minimizers* $\mathbf{M}^*, \mathbf{N}^*$ *have rank equal to* $Q$ *.*

*Further, the local LMI constraints* $\mathbf{G}_{1,j}\succeq 0$ *for* $j=1,\ldots,P$ *in* (18) *ensure that the feasible set of* (18) *contains a minimizing factor* $\mathbf{M}^*$ *and kernel coefficients* $\{\alpha^*_b\}_{b=1}^B$ *such that* $\sum_b \alpha^*_b\mathbf{A}^b_x$ *has rank at least* $Q$*.*

*If the kernel dictionary $\mathcal{D}$ contains a unique subset of kernels whose convex combination is block diagonal with $Q$ diagonal blocks of rank 1, then factors $\mathbf{M}^*, \mathbf{N}^*$ and coefficients $\{\alpha_b^*\}_{b=1}^B$ exist that are minimizers of both* (6),(8) *and* (18)-(19) *for sufficiently large $\omega$ and $\upsilon$ parameters.*

Prop. 4 implies that although the formulations in (6) [ or (8)] and (18) [or (19)] are not equivalent, the minimizing $\mathbf{M}^*, \mathbf{N}^*$ and $\{\alpha_b^*\}_{b=1}^B$ will be selected such that $\text{rank}(\mathbf{M}^*) = \text{rank}(\mathbf{N}^*) = Q$ and $\text{rank}(\sum_b \alpha_b^* \mathbf{A}_x^b) \geq Q$. These are the exact same rank requirement imposed by the first and fourth LMI constraints in the centralized formulation in (4).

### 4.3 Distributed Minimization via Alternating Direction Method of Multipliers (ADMM)

ADMM (Bertsekas and Tsitsiklis, 2015; Boyd et al., 2011) is utilized to solve the minimization problems in (18) and (19) in a distributed fashion that allows communication only between single-hop neighboring units. The augmented Lagrangian associated with (18) is

$$\mathcal{L}_{\tau,\kappa}(\{\mathbf{M}_{\mathcal{N}_j}, \boldsymbol{\alpha}^j, w_j, \psi_j\}_{j=1}^P, \boldsymbol{\zeta}, \bar{\boldsymbol{\zeta}}, \boldsymbol{\xi}, \bar{\boldsymbol{\xi}}) := -\mu \cdot |\mathcal{S}_Q|^{-1} \sum_{j \in \mathcal{S}_Q} w_j + P^{-1} \sum_j [\omega \psi_j + \xi \theta_j] \tag{20}$$

$$+ \upsilon \sum_{\ell=1}^P \left[ \|\mathbf{M}_{\ell,:}^\ell\|_1 - \frac{\mathbf{M}_{\tau,\kappa,\ell,:}^\ell}{\|\mathbf{M}_{\tau,\kappa,\ell,:}^\ell\|_2} (\mathbf{M}_{\ell,:}^\ell)^T \right] + \sum_{j=1}^P \sum_{j' \in \mathcal{N}_j} \boldsymbol{\zeta}_{j,j'}^T [\mathbf{M}_{j,:}^j - \mathbf{Z}_{j,j'}]^T + \sum_{j=1}^P \sum_{j' \in \mathcal{N}_j} \bar{\boldsymbol{\zeta}}_{j',j}^T [\mathbf{M}_{j,:}^{j'}$$

$$- \mathbf{Z}_{j,j'}]^T + \sum_{j=1}^P \sum_{j' \in \mathcal{N}_j} \boldsymbol{\xi}_{j,j'}^T [\boldsymbol{\alpha}^j - \boldsymbol{\beta}_{j,j'}] + \sum_{j=1}^P \sum_{j' \in \mathcal{N}_j} \bar{\boldsymbol{\xi}}_{j',j}^T [\boldsymbol{\alpha}^{j'} - \boldsymbol{\beta}_{j,j'}]$$

$$+ 0.5 \cdot c \cdot \sum_{j=1}^P \sum_{j' \in \mathcal{N}_j} \left[ \|\mathbf{M}_{j,:}^j - \mathbf{Z}_{j,j'}\|_2^2 + \|\mathbf{M}_{j,:}^{j'} - \mathbf{Z}_{j,j'}\|_2^2 \right],$$

$$+ 0.5 \cdot c \cdot \sum_{j=1}^P \sum_{j' \in \mathcal{N}_j} \left[ \|\boldsymbol{\alpha}^j - \boldsymbol{\beta}_{j,j'}\|_2^2 + \|\boldsymbol{\alpha}^{j'} - \boldsymbol{\beta}_{j,j'}\|_2^2 \right]$$

where $\boldsymbol{\alpha}^j = [\alpha_1^j, \ldots, \alpha_B^j]$, $\boldsymbol{\beta}_{j,j'} = [\beta_1^{j,j'}, \ldots, \beta_B^{j,j'}]$, while $\boldsymbol{\zeta} := \{\boldsymbol{\zeta}_{j,j'}\}, \bar{\boldsymbol{\zeta}} := \{\bar{\boldsymbol{\zeta}}_{j,j'}\}$, $\boldsymbol{\xi} := \{\boldsymbol{\xi}_{j,j'}\}, \bar{\boldsymbol{\xi}} := \{\bar{\boldsymbol{\xi}}_{j,j'}\}$ contain the $Q \times 1$ Lagrange multiplier vectors $\boldsymbol{\zeta}_{j,j'}$ and $\bar{\boldsymbol{\zeta}}_{j',j}$ associated with the constraints $\mathbf{M}_{j,:}^j = \mathbf{Z}_{j,j'}$ and $\mathbf{M}_{j,:}^{j'} = \mathbf{Z}_{j,j'}$, respectively, whereas the $B \times 1$ multiplier vectors $\boldsymbol{\xi}_{j,j'}$ and $\bar{\boldsymbol{\xi}}_{j',j}$ are associated with the equality constraints $\boldsymbol{\alpha}^j = \boldsymbol{\beta}_{j,j'}$ and $\boldsymbol{\alpha}^{j'} = \boldsymbol{\beta}_{j,j'}$, respectively. Constant $c$ is a nonnegative coefficient imposing strict convexity and it will be acting as a step-size for updating the multipliers.

ADMM updates the variables and multipliers in (20) during alternating iteration $\rho + 1$ through the following step

**Step D1a:** Unit $j$ obtains local iterates $\{\mathbf{M}_{\mathcal{N}_j}^{\tau,\kappa}(\rho+1), \boldsymbol{\alpha}_{\tau,\kappa}^j(\rho+1), w_j^{\tau,\kappa}(\rho+1), \psi_j^{\tau,\kappa}(\rho+1), \theta_j^{\tau,\kappa}(\rho+1)\}$ by solving the following local minimization problem that stems from (20) and the constraints

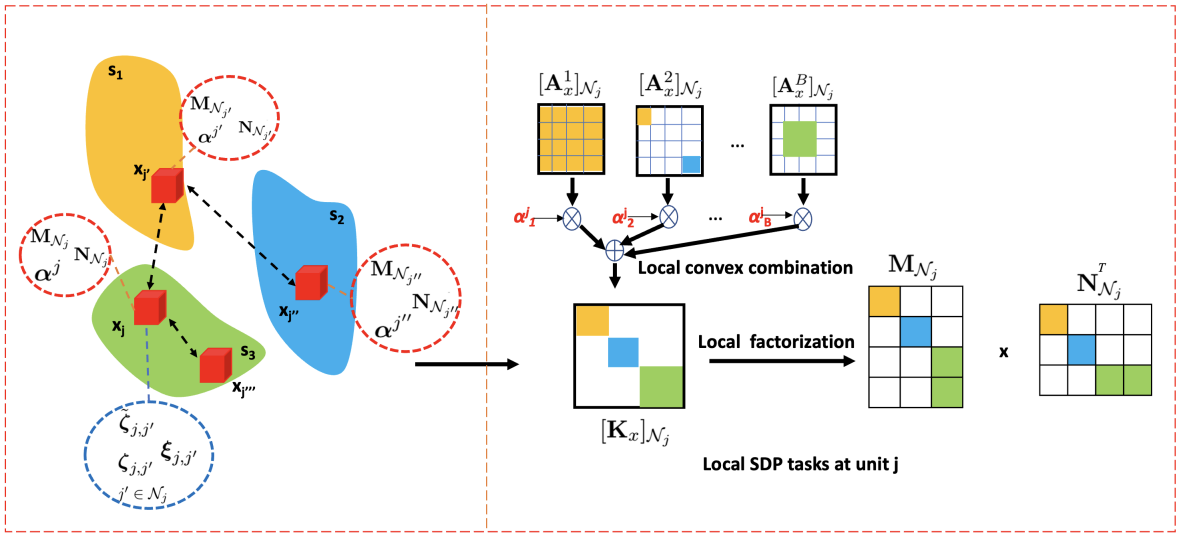


Figure 2: Localized kernel learning and clustering tasks.

in (18), after keeping the $j$th unit local terms/variables

$$
\begin{aligned}
&\arg \min_{\mathbf{M}_{\mathcal{N}_j}, \boldsymbol{\alpha}^j, w_j, \psi_j, \theta_j} -\mu \mathbb{1}_{j \in \mathcal{S}_Q} |\mathcal{S}_Q|^{-1} w_j + \omega P^{-1} \psi_j + \xi P^{-1} \theta_j + \left[ \|\mathbf{M}^j_{j,:}\|_1 - \frac{\mathbf{M}^j_{\tau,\kappa,j,:}}{\|\mathbf{M}^j_{\tau,\kappa,j,:}\|_2} (\mathbf{M}^j_{j,:})^T \right] \\
&+ \sum_{j' \in \mathcal{N}_j} (\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'})^T(\rho) [\mathbf{M}^j_{j,:} - \mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho)]^T + \sum_{j' \in \mathcal{N}_j} (\bar{\boldsymbol{\zeta}}^{\tau,\kappa}_{j,j'})^T(\rho) [\mathbf{M}^j_{j',:} - \mathbf{Z}^{\tau,\kappa}_{j',j}(\rho)]^T \\
&+ \sum_{j' \in \mathcal{N}_j} (\boldsymbol{\xi}^{\tau,\kappa}_{j,j'})^T(\rho) [\boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho)] + \sum_{j' \in \mathcal{N}_j} (\bar{\boldsymbol{\xi}}^{\tau,\kappa}_{j,j'})^T(\rho) [\boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j',j}(\rho)] \\
&+ 0.5 \cdot c \cdot \sum_{j' \in \mathcal{N}_j} \left[ \|\mathbf{M}^j_{j,:} - \mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho)\|_2^2 + \|\mathbf{M}^j_{j',:} - \mathbf{Z}^{\tau,\kappa}_{j',j}(\rho)\|_2^2 \right] \\
&+ 0.5 \cdot c \cdot \sum_{j' \in \mathcal{N}_j} \left[ \|\boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho)\|_2^2 + \|\boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j',j}(\rho)\|_2^2 \right], \\
&\text{s. to } \mathbf{G}_{1,j} \succeq 0, \mathbf{G}_{\tau,2,j}(\mathbf{N}_\tau) \succeq 0, \mathbf{G}_{\tau,3,j}(\mathbf{N}_\tau) \succeq 0, \\
&\mathbb{1}_{j \in \mathcal{S}_Q} \cdot \left( [\mathbf{H}^2_\tau(\mathbf{N}_\tau)]_{\mathcal{N}_j} - w_j \cdot \mathbf{I}_Q \right) \succeq \mathbf{0}, w_j \geq 0, \{\alpha^j_b \geq 0\}_{b=1}^B, \textstyle\sum_b \alpha^j_b = 1,
\end{aligned} \tag{21}
$$

where $\mathbb{1}_{j \in \mathcal{S}_Q}$ is an indicator function equal to 1 if $j \in \mathcal{S}_Q$ and zero otherwise.

**Step D1b:** Unit $j$ updates the auxiliary variables $\mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho+1), \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho+1)$ for $j' \in \mathcal{N}_j$ via the quadratic program

$$\begin{aligned}\{\mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho+1), \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho+1)\} = \arg \min_{\mathbf{Z}_{j,j'}, \boldsymbol{\beta}_{j,j'}} & -(\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'})^T(\rho)\cdot \mathbf{Z}_{j,j'} \\ & - (\bar{\boldsymbol{\zeta}}^{\tau,\kappa}_{j',j})^T(\rho)\cdot \mathbf{Z}_{j,j'} - (\boldsymbol{\xi}^{\tau,\kappa}_{j,j'})^T(\rho)\cdot \boldsymbol{\beta}_{j,j'} - \bar{\boldsymbol{\xi}}^T_{j',j}(\rho)\boldsymbol{\beta}_{j,j'} \\ & + 0.5\cdot c\left[\|\mathbf{M}^j_{\tau,\kappa,j,:}(\rho+1) - \mathbf{Z}_{j,j'}\|_2^2 + \|\mathbf{M}^{j'}_{\tau,\kappa,j,:}(\rho+1) - \mathbf{Z}_{j,j'}\|_2^2\right] \\ & + 0.5\cdot c\cdot\left[\|\boldsymbol{\alpha}^j_{\tau,\kappa}(\rho+1) - \boldsymbol{\beta}_{j,j'}\|_2^2 + \|\boldsymbol{\alpha}^{j'}_{\tau,\kappa}(\rho+1) - \boldsymbol{\beta}_{j,j'}\|_2^2\right],\end{aligned} \tag{22}$$

which after applying first-order optimality conditions results:

$$\mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho+1) = 0.5\cdot c^{-1}[\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'}(\rho) + \bar{\boldsymbol{\zeta}}^{\tau,\kappa}_{j',j}(\rho)] + 0.5\cdot[\mathbf{M}^j_{\tau,\kappa,j,:}(\rho+1) + \mathbf{M}^{j'}_{\tau,\kappa,j,:}(\rho+1)], \tag{23}$$

$$\boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho+1) = 0.5\cdot c^{-1}[\boldsymbol{\xi}^{\tau,\kappa}_{j,j'}(\rho) + \bar{\boldsymbol{\xi}}^{\tau,\kappa}_{j',j}(\rho)] + 0.5\cdot[\boldsymbol{\alpha}^j_{\tau,\kappa}(\rho+1) + \boldsymbol{\alpha}^{j'}_{\tau,\kappa}(\rho+1)]. \tag{24}$$

**Step D1c:** Sensing unit $j = 1,\ldots,P$ updates the Lagrange multipliers $\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'}(\rho+1)$, $\bar{\boldsymbol{\zeta}}^{\tau,\kappa}_{j,j'}(\rho+1)$, $\boldsymbol{\xi}^{\tau,\kappa}_{j,j'}(\rho+1)$, $\bar{\boldsymbol{\xi}}^{\tau,\kappa}_{j,j'}(\rho+1)$ for $j' \in \mathcal{N}_j$ using the gradient ascent iterations:

$$\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'}(\rho+1) = \boldsymbol{\zeta}^{\tau,\kappa}_{j,j'}(\rho) + c\cdot[\mathbf{M}^j_{\tau,\kappa,j,:}(\rho+1) - \mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho+1)], \tag{25}$$

$$\bar{\boldsymbol{\zeta}}^{\tau,\kappa}_{j,j'}(\rho+1) = \bar{\boldsymbol{\zeta}}^{\tau,\kappa}_{j,j'}(\rho) + c\cdot[\mathbf{M}^j_{\tau,\kappa,j',:}(\rho+1) - \mathbf{Z}^{\tau,\kappa}_{j',j}(\rho+1)], \tag{26}$$

$$\boldsymbol{\xi}^{\tau,\kappa}_{j,j'}(\rho+1) = \boldsymbol{\xi}^{\tau,\kappa}_{j,j'}(\rho) + c\cdot[\boldsymbol{\alpha}^j_{\tau,\kappa}(\rho+1) - \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho+1)], \tag{27}$$

$$\bar{\boldsymbol{\xi}}^{\tau,\kappa}_{j,j'}(\rho+1) = \bar{\boldsymbol{\xi}}^{\tau,\kappa}_{j,j'}(\rho) + c\cdot[\boldsymbol{\alpha}^j_{\tau,\kappa}(\rho+1) - \boldsymbol{\beta}^{\tau,\kappa}_{j',j}(\rho+1)]. \tag{28}$$

Substituting (23) into (25) and (26), and (24) into (27) and (28), it follows that if the Lagrange multipliers are initialized such that $\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'}(0) = -\bar{\boldsymbol{\zeta}}^{\tau,\kappa}_{j',j}(0)$ and $\boldsymbol{\xi}^{\tau,\kappa}_{j,j'}(0) = -\bar{\boldsymbol{\xi}}^{\tau,\kappa}_{j',j}(0)$ then $\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'}(\rho) = -\bar{\boldsymbol{\zeta}}^{\tau,\kappa}_{j',j}(\rho)$ and $\boldsymbol{\xi}^{\tau,\kappa}_{j,j'}(\rho) = -\bar{\boldsymbol{\xi}}^{\tau,\kappa}_{j',j}(\rho)$ for all $\tau,\kappa$ and $\rho$ indices. This implies that there is no need to keep track of the multipliers $\bar{\boldsymbol{\zeta}}_{j',j}$ and $\bar{\boldsymbol{\xi}}_{j',j}$ as long as the multipliers $\boldsymbol{\zeta}_{j,j'}$ and $\boldsymbol{\xi}_{j,j'}$ are updated using the recursions

$$\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'}(\rho+1) = \boldsymbol{\zeta}^{\tau,\kappa}_{j,j'}(\rho) + 0.5\cdot c\cdot[\mathbf{M}^j_{\tau,\kappa,j,:}(\rho+1) - \mathbf{M}^{j'}_{\tau,\kappa,j,:}(\rho+1)], \tag{29}$$

$$\boldsymbol{\xi}^{\tau,\kappa}_{j,j'}(\rho+1) = \boldsymbol{\xi}^{\tau,\kappa}_{j,j'}(\rho) + 0.5\cdot c\cdot[\boldsymbol{\alpha}^j_{\tau,\kappa}(\rho+1) - \boldsymbol{\alpha}^{j'}_{\tau,\kappa}(\rho+1)], \tag{30}$$

which can be obtained after employing (23) and (24) in (25) and (27), respectively. Further, eqs. (23) and (24) are simplified

$$\mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho+1) = 0.5\cdot[\mathbf{M}^j_{\tau,\kappa,j,:}(\rho+1) + \mathbf{M}^{j'}_{\tau,\kappa,j,:}(\rho+1)] \tag{31}$$

$$\boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho+1) = 0.5\cdot[\boldsymbol{\alpha}^j_{\tau,\kappa}(\rho+1) + \boldsymbol{\alpha}^{j'}_{\tau,\kappa}(\rho+1)], \tag{32}$$

and replace $\mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho+1)$ and $\boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho+1)$ in (21) with expressions in (31) and (32), respectively. Thus, $\mathbf{Z}^{\tau,\kappa}_{j,j'}, \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}$ do not have to be updated as separate variables.

The formulation in (21) can be transformed in a SDP formulation after introducing local variables $\delta^1_{j,j'}$, $\delta^2_{j,j'}$, and $\delta^3_{j,j'}$ for $j' \in \mathcal{N}_j$, that will replace the last four quadratic terms in the cost of (21) by introducing the inequality constraints

$$\begin{aligned}&\|\mathbf{M}^j_{j,:} - \mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho)\|_2^2 \leq \delta^1_{j,j'}, \|\mathbf{M}^j_{j',:} - \mathbf{Z}^{\tau,\kappa}_{j',j}(\rho)\|_2^2 \leq \delta^2_{j,j'} \\ &\|\boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho)\|_2^2 = \|\boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j',j}(\rho)\|_2^2 \leq \delta^3_{j,j'},\end{aligned} \tag{33}$$

which can be rewritten using the following LMIs

$$\boldsymbol{\Delta}^1_{j,j'} := \begin{bmatrix} \delta^1_{j,j'} \cdot \mathbf{I}_Q & (\mathbf{M}^j_{j,:} - \mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho))^T \\ \mathbf{M}^j_{j,:} - \mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho) & 1 \end{bmatrix} \succeq \mathbf{0},$$
$$\boldsymbol{\Delta}^2_{j,j'} := \begin{bmatrix} \delta^2_{j,j'} \cdot \mathbf{I}_Q & (\mathbf{M}^j_{j',:} - \mathbf{Z}^{\tau,\kappa}_{j',j}(\rho))^T \\ \mathbf{M}^j_{j',:} - \mathbf{Z}^{\tau,\kappa}_{j',j}(\rho) & 1 \end{bmatrix} \succeq \mathbf{0},$$
$$\boldsymbol{\Delta}^3_{j,j'} := \begin{bmatrix} \delta^3_{j,j'} \cdot \mathbf{I}_Q & (\boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho))^T \\ \boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho) & 1 \end{bmatrix} \succeq \mathbf{0}. \tag{34}$$

Thus, the minimization formulation (21) in Step D1 can be rewritten as an SDP in the following way:

$$\begin{aligned} &\arg \min_{\mathbf{M}_{\mathcal{N}_j}, \boldsymbol{\alpha}^j, w_j, \psi_j, \theta_j} -\mu \mathbb{1}_{j \in \mathcal{S}_Q} |\mathcal{S}_Q|^{-1} w_j + \omega P^{-1} \psi_j + \xi P^{-1} \theta_j + \left[ \|\mathbf{M}^j_{j,:}\|_1 - \frac{\mathbf{M}^j_{\tau,\kappa,j,:}}{\|\mathbf{M}^j_{\tau,\kappa,j,:}\|_2} (\mathbf{M}^j_{j,:})^T \right] \\ &+ \sum_{j' \in \mathcal{N}_j} (\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'})^T(\rho)[\mathbf{M}^j_{j,:} - \mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho)] - \sum_{j' \in \mathcal{N}_j} (\boldsymbol{\zeta}^{\tau,\kappa}_{j',j})^T(\rho)[\mathbf{M}^j_{j',:} - \mathbf{Z}^{\tau,\kappa}_{j',j}(\rho)] \\ &+ \sum_{j' \in \mathcal{N}_j} (\boldsymbol{\xi}^{\tau,\kappa}_{j,j'})^T(\rho)[\boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho)] - \sum_{j' \in \mathcal{N}_j} (\boldsymbol{\xi}^{\tau,\kappa}_{j',j})^T(\rho)[\boldsymbol{\alpha}^j - \boldsymbol{\beta}^{\tau,\kappa}_{j',j}(\rho)] \\ &+ 0.5 \cdot c \cdot \sum_{j' \in \mathcal{N}_j} \left[ \delta^1_{j,j'} + \delta^2_{j,j'} + 2 \cdot \delta^3_{j,j'} \right] \\ &\text{subject to } \mathbf{G}_{1,j} \succeq 0, \mathbf{G}_{\tau,2,j}(\mathbf{N}_\tau) \succeq 0, \mathbf{G}_{\tau,3,j}(\mathbf{N}_\tau) \succeq 0, \mathbb{1}_{j \in \mathcal{S}_Q} \cdot \left( [\mathbf{H}^2_\tau(\mathbf{N}_\tau)]_{\mathcal{N}_j} - w_j \cdot \mathbf{I}_Q \right) \succeq \mathbf{0}, \\ &\boldsymbol{\Delta}^1_{j,j'} \succeq 0,\ \boldsymbol{\Delta}^2_{j,j'} \succeq 0,\ \ \boldsymbol{\Delta}^3_{j,j'} \succeq 0,\ j' \in \mathcal{N}_j, w_j \geq 0, \{\alpha^j_b \geq 0\}_{b=1}^B, \textstyle\sum_b \alpha^j_b = 1, \end{aligned} \tag{35}$$

where the equation $\boldsymbol{\zeta}^{\tau,\kappa}_{j,j'}(\rho) = -\bar{\boldsymbol{\zeta}}^{\tau,\kappa}_{j',j}(\rho)$ and $\boldsymbol{\xi}^{\tau,\kappa}_{j,j'}(\rho) = -\bar{\boldsymbol{\xi}}^{\tau,\kappa}_{j',j}(\rho)$ for all $\tau, \kappa$ and $\rho$ has been employed, while (31) and (32) can be employed to replace $\mathbf{Z}^{\tau,\kappa}_{j,j'}(\rho)$ and $\boldsymbol{\beta}^{\tau,\kappa}_{j,j'}(\rho)$.

Essentially the distributed steps D1a-D1c facilitate solving the SDP cost (6) involved in Step 1b of the centralized algorithm in Sec. 3.1, i.e., provide a distributed implementation of line 6 in Alg. 1. Steps D1a-D1c essentially consist a third recursive layer nested within that $\tau$, and $\kappa$ iterations in Alg. 1. A similar process can be obtained when tackling (8) via ADMM that involves the following steps

**Step D2a:** Sensing unit $j = 1, \ldots, P$ obtains local iterates $\{\mathbf{N}^{\tau,\kappa}_{\mathcal{N}_j}(\rho+1), w^{\tau,\kappa}_j(\rho+1), \psi^{\tau,\kappa}_j(\rho+1), \theta^{\tau,\kappa}_j(\rho+1)\}$ by solving

$$\begin{aligned} &\arg \min_{\mathbf{N}_{\mathcal{N}_j}, w_j, \psi_j, \theta_j} -\mu \mathbb{1}_{j \in \mathcal{S}_Q} |\mathcal{S}_Q|^{-1} w_j + \omega P^{-1} \psi_j + \xi P^{-1} \theta_j + \left[ \|\mathbf{N}^j_{j,:}\|_1 - \frac{\mathbf{N}^j_{\tau,\kappa,j,:}}{\|\mathbf{N}^j_{\tau,\kappa,j,:}\|_2} (\mathbf{N}^j_{j,:})^T \right] \\ &+ \sum_{j' \in \mathcal{N}_j} (\tilde{\boldsymbol{\zeta}}^{\tau,\kappa}_{j,j'})^T(\rho)[\mathbf{N}^j_{j,:} - \tilde{\mathbf{Z}}^{\tau,\kappa}_{j,j'}(\rho)] - \sum_{j' \in \mathcal{N}_j} (\tilde{\boldsymbol{\zeta}}^{\tau,\kappa}_{j',j})^T(\rho)[\mathbf{N}^j_{j',:} - \tilde{\mathbf{Z}}^{\tau,\kappa}_{j',j}(\rho)] + 0.5 \cdot c \cdot \sum_{j' \in \mathcal{N}_j} \left[ \tilde{\delta}^1_{j,j'} + \tilde{\delta}^2_{j,j'} \right] \\ &\text{s. to } \mathbf{G}_{\tau,2,j}(\mathbf{M}_{\tau+1}) \succeq 0, \mathbf{G}_{\tau,3,j}(\mathbf{M}_{\tau+1}) \succeq 0, \\ &\mathbb{1}_{j \in \mathcal{S}_Q} \cdot \left( [\mathbf{H}^2_\tau(\mathbf{M}_{\tau+1})]_{\mathcal{N}_j} - w_j \cdot \mathbf{I}_Q \right) \succeq \mathbf{0}, \\ &\tilde{\boldsymbol{\Delta}}^1_{j,j'} \succeq 0,\ \tilde{\boldsymbol{\Delta}}^2_{j,j'} \succeq 0,\ j' \in \mathcal{N}_j,\ w_j \geq 0, \end{aligned} \tag{36}$$

**Algorithm 2** Distributed Kernel Selection and Clustering (DKC)

1: Each unit $j = 1, \ldots, P$ initializes the multipliers $\boldsymbol{\zeta}_{j,j'}^{\tau,\kappa}(0)$, and $\boldsymbol{\xi}_{j,j'}^{\tau,\kappa}(0)$, e.g., set them to all-zero vectors .
2: **for** $\rho = 0, 1, 2..., \rho_t$ **do**
3: Unit $j = 1, \ldots, P$ transmits multipliers $\boldsymbol{\zeta}_{j,j'}^{\tau,\kappa}(\rho), \boldsymbol{\xi}_{j,j'}^{\tau,\kappa}(\rho)$, row factors $\mathbf{M}_{\tau,\kappa,j',:}^{j}(\rho+1), \mathbf{M}_{\tau,\kappa,j,:}^{j}(\rho+1)$ and kernel coefficients $\boldsymbol{\alpha}_{\tau,\kappa}^{j}(\rho+1)$ to its neighbors $j' \in \mathcal{N}_j$.
4: Unit $j = 1, \ldots, P$ receives from neighbors $j' \in \mathcal{N}_j$ the factors $\mathbf{M}_{\tau,\kappa,j,:}^{j'}(\rho+1)$, $\mathbf{M}_{\tau,\kappa,j',:}^{j'}(\rho+1)$, the kernel coefficients $\boldsymbol{\alpha}_{\tau,\kappa}^{j'}(\rho+1)$ and the Lagrange multipliers $\boldsymbol{\zeta}_{j',j}^{\tau,\kappa}(\rho)$ and $\boldsymbol{\xi}_{j',j}^{\tau,\kappa}(\rho)$.
5: Update the auxiliary variables $\mathbf{Z}_{j,j'}^{\tau,\kappa}(\rho)$, $\mathbf{Z}_{j',j}^{\tau,\kappa}(\rho)$ and $\boldsymbol{\beta}_{j,j'}^{\tau,\kappa}(\rho)$ using (31) and (32).
6: Unit $j = 1, \ldots, P$ fsolves the local SDP program in (35) to obtain the updates $\mathbf{M}_{\mathcal{N}_j}^{\tau,\kappa}(\rho+1), \boldsymbol{\alpha}_{\tau,\kappa}^{j}(\rho+1)$. This can be done locally at unit $j$ using interior point methods.
7: Unit $j$ updates the local Lagrange multipliers $\boldsymbol{\zeta}_{j,j'}^{\tau,\kappa}(\rho+1)$ and $\boldsymbol{\beta}_{j,j'}^{\tau,\kappa}(\rho+1)$ for $j' \in \mathcal{N}_j$ using (29) and (30).
8: **end for**
9: Form $\mathbf{M}_{\tau,\kappa+1}$ using the locally factor rows, i.e., $\mathbf{M}_{\tau,\kappa+1} = \{\mathbf{M}_{\tau,\kappa,j,:}^{j}(\rho_t+1)\}_{j=1}^{P}$, and $\boldsymbol{\alpha}_{\tau,\kappa+1} = \boldsymbol{\alpha}_{\tau,\kappa}^{j}(\rho_t+1)$ for any $j$.

where local variables and multipliers $\tilde{\boldsymbol{\zeta}}_{j,j'}^{\tau,\kappa}(\rho), \tilde{\boldsymbol{\xi}}_{j,j'}^{\tau,\kappa}(\rho), \tilde{\mathbf{Z}}_{j,j'}^{\tau,\kappa}(\rho), \tilde{\boldsymbol{\beta}}_{j,j'}^{\tau,\kappa}(\rho), \tilde{\delta}_{j,j'}^{1}, \tilde{\delta}_{j,j'}^{2}, \tilde{\boldsymbol{\Delta}}_{j,j'}^{1}, \tilde{\boldsymbol{\Delta}}_{j,j'}^{2}$, are defined similarly to the corresponding variables without the˜notation in (35), and replacing $\mathbf{M}$ with $\mathbf{N}$, $\delta$ with $\tilde{\delta}$, $\boldsymbol{\Delta}$ with $\tilde{\boldsymbol{\Delta}}$ and $\mathbf{Z}$ with $\tilde{\mathbf{Z}}$ in (33)-(34).

**Step D2b:** Sensing unit $j = 1, \ldots, P$ updates the auxiliary variables, i.e., $\tilde{\mathbf{Z}}_{j,j'}^{\tau,\kappa}(\rho+1)$ for $j' \in \mathcal{N}_j$ via [similar process as in (23)]

$$\tilde{\mathbf{Z}}_{j,j'}^{\tau,\kappa}(\rho+1) = 0.5 \cdot [\mathbf{N}_{\tau,\kappa,j,:}^{j}(\rho+1) + \mathbf{N}_{\tau,\kappa,j,:}^{j'}(\rho+1)], \tag{37}$$

and can be used in place of $\tilde{\mathbf{Z}}_{j,j'}^{\tau,\kappa}(\rho+1)$ in (36). Thus, $\tilde{\mathbf{Z}}_{j,j'}^{\tau,\kappa}$ does not have to be updated as a separate variable.

**Step D2c:** Sensing unit $j = 1, \ldots, P$ updates the Lagrange multipliers $\tilde{\boldsymbol{\zeta}}_{j,j'}^{\tau,\kappa}(\rho+1)$ using the gradient ascent iterations:

$$\tilde{\boldsymbol{\zeta}}_{j,j'}^{\tau,\kappa}(\rho+1) = \tilde{\boldsymbol{\zeta}}_{j,j'}^{\tau,\kappa}(\rho) + 0.5 \cdot c \cdot [\mathbf{N}_{\tau,\kappa,j,:}^{j}(\rho+1) - \mathbf{N}_{\tau,\kappa,j,:}^{j'}(\rho+1)], \tag{38}$$

Alg. 2 tabulates in detail the steps involved in implementing D1a-D1c (similarly for D2a-D2c); see also Fig. 2. Alg. 2 can also be used to summarize steps D2a-D2c after replacing i) $\boldsymbol{\zeta}_{j,j'}^{\tau,\kappa}$ with $\tilde{\boldsymbol{\zeta}}_{j,j'}^{\tau,\kappa}$; ii) $\mathbf{M}_{\tau,\kappa,j,:}^{j'}(\rho+1)$ with $\mathbf{N}_{\tau,\kappa,j,:}^{j'}(\rho+1)$; iii) $\mathbf{Z}_{j',j}^{\tau,\kappa}(\rho)$ with $\tilde{\mathbf{Z}}_{j',j}^{\tau,\kappa}(\rho)$; iv) remove $\boldsymbol{\alpha}^{j,\tau,\kappa}$, $\boldsymbol{\xi}_{j',j}^{\tau,\kappa}$, and $\boldsymbol{\beta}_{j,j'}^{\tau,\kappa}$; and replace (35) with (36). Further, replace eqs. (31) and (32) with (37), and eqs. (29) and (30) with (38). Note that $\rho_t$ denotes the number of ADMM iterations applied for every $\tau$ and $\kappa$ iterations.

**Communication Costs:** During steps D1a-D1c unit $j$ has to receive i) $|\mathcal{N}_j|$ Lagrange multiplier vectors $\{\boldsymbol{\zeta}_{j',j}^{\tau,\kappa}(\rho)\}_{j'\in\mathcal{N}_j}$, $2 \cdot |\mathcal{N}_j|$ row factors $\{\mathbf{M}_{\tau,\kappa,j,:}^{j'}(\rho+1)\}_{j'\in\mathcal{N}_j}$, $\{\mathbf{M}_{j',:}^{j',\tau,\kappa}(\rho+1)\}_{j'\in\mathcal{N}_j}$ each having $Q$ scalar entries; and ii) $|\mathcal{N}_j|$ Lagrange multiplier vectors $\{\boldsymbol{\xi}_{j',j}^{\tau,\kappa}(\rho)\}_{j'\in\mathcal{N}_j}$, and $|\mathcal{N}_j|$ kernel coefficient vectors $\{\boldsymbol{\alpha}_{\tau,\kappa}^{j'}(\rho+1)\}_{j'\in\mathcal{N}_j}$ of length $B$. Thus, the number of scalars unit $j$ has to receive during ADMM iteration $\rho$ amounts to $3|\mathcal{N}_j| \cdot Q + 2|\mathcal{N}_j| \cdot B$.

Further, unit $j$ has to transmit i) $|\mathcal{N}_j|$ Lagrange multiplier vectors $\{\boldsymbol{\zeta}_{j,j'}^{\tau,\kappa}(\rho)\}_{j'\in\mathcal{N}_j}$, $|\mathcal{N}_j|+1$ row factors $\{\{\mathbf{M}_{\tau,\kappa,j',:}^{j}(\rho+1)\}_{j'\in\mathcal{N}_j}, \mathbf{M}_{\tau,\kappa,j,:}^{j}(\rho+1)\}$ each of length $Q$; and ii) $|\mathcal{N}_j|$ Lagrange multiplier vectors $\{\boldsymbol{\xi}_{j,j'}^{\tau,\kappa}(\rho)\}_{j'\in\mathcal{N}_j}$, and one kernel coefficient vector $\{\boldsymbol{\alpha}_{\tau,\kappa}^{j}(\rho+1)\}_{j'\in\mathcal{N}_j}$ each of

length $B$. Thus, the total number of scalars to be transmitted during ADMM iteration $\rho$ amounts to $(2 \cdot |\mathcal{N}_j| + 1)Q + (|\mathcal{N}_j| + 1)B$.

During steps D2a-D2c unit $j$ has to receive $|\mathcal{N}_j|$ Lagrange multiplier vectors $\{\tilde{\boldsymbol{\zeta}}^{\tau,\kappa}_{j',j}(\rho)\}_{j' \in \mathcal{N}_j}$, $2 \cdot |\mathcal{N}_j|$ row factors $\{\{\mathbf{N}^{j'}_{\tau,\kappa,j,:}(\rho+1)\}_{j' \in \mathcal{N}_j}, \{\mathbf{N}^{j'}_{\tau,\kappa,j',:}(\rho+1)\}\}_{j' \in \mathcal{N}_j}$ each having $Q$ scalar entries. Thus, the number of scalars unit $j$ has to receive during iteration $\rho$ amounts to $3|\mathcal{N}_j| \cdot Q$. Further, unit $j$ has to transmit $|\mathcal{N}_j|$ Lagrange multiplier vectors $\{\tilde{\boldsymbol{\zeta}}^{\tau,\kappa}_{j,j'}(\rho)\}_{j' \in \mathcal{N}_j}$, $|\mathcal{N}_j| + 1$ row factors $\{\mathbf{N}^{j}_{\tau,\kappa,j',:}(\rho+1)\}_{j' \in \mathcal{N}_j}, \mathbf{N}^{j}_{\tau,\kappa,j,:}(\rho+1)$ each of length $Q$. Thus, the total number of scalars to be transmitted during ADMM iteration $\rho$ amounts to $(2 \cdot |\mathcal{N}_j| + 1)Q$.

In summary, the communication cost is proportional to the neighborhood size $|\mathcal{N}_j|$, the number of different classes/sources $Q$ and the size of the kernel dictionary $B$ which in practice are much smaller compared to the number of sensing agents $P$.

**Remark on finding set $\mathcal{S}_Q$:** The clustering problem does not have to be solved to determine which such nodes belong in $\mathcal{S}_Q$ and have access to neighborhood measurements that cover all $Q$ sources present. Specifically, each unit $S_j$ can solve a local version of (4) [considering only its neighborhood observations and corresponding kernel similarity matrices]. This can be done once during an initialization phase. If the neighborhood $\mathcal{N}_j$ is large enough such that the associated measurements contain information about all $Q$ sources, then when solving a local version of (4) at the candidate unit $S_j$ the optimal variable $w$ in (4) will be strictly positive (details in Apdx. A), whereas if the neighborhood measurements were covering less than $Q$ sources then $w$ will be zero (rank-reduced $\mathbf{M}$). Thus, each unit $S_j$ can identify whether they belong to a set $\mathcal{S}_Q$ and adjust their corresponding constraints in steps D1a and D2a accordingly.

The same idea can be applied when setting up the network of sensing units during an initialization phase to ensure the presence of such a node, i.e., nonempty set $\mathcal{S}_Q$. During initialization, one such node $S_j$ with longer receiving range capabilities can start gradually increasing reception range, thus incrementally increasing its neighborhood radius, and therefore the set of neighborhood measurements $\mathbf{x}_{j'}$ for $j \in \mathcal{N}_j$ that it can access. Unit $S_j$ can then solve a local version of (4) with each current neighborhood size and if $w = 0$ then it keeps increasing the neighborhood size (or communication range) , until $w > 0$ in which case $S_j$ knows its neighborhood measurements cover all $Q$ sources present.

**Distributed Convergence:** Given that the separable formulations in (18) and (19) are convex, we prove in Apdx. E that the updates $\{\mathbf{M}^{\tau,\kappa}_{\mathcal{N}_j}(\rho+1), \boldsymbol{\alpha}^{j}_{\tau,\kappa}(\rho+1)\}$ and $\mathbf{N}^{\tau,\kappa}_{\mathcal{N}_j}(\rho+1)$ obtained via steps D1a-D1c and D2a-D2c to cluster the data and determine the kernel similarity matrix, converge, i.e.,

**Proposition 5** *Under the assumptions of Prop. 4 and for fixed iteration indices $\tau, \kappa$ the ADMM iterates from* (18) *and* (19) $\{\mathbf{M}^{\tau,\kappa}_{\mathcal{N}_j}(\rho+1), \mathbf{N}^{\tau,\kappa}_{\mathcal{N}_j}(\rho+1), \boldsymbol{\alpha}^{j}_{\tau,\kappa}(\rho+1)\}$ *converge as $\rho \to \infty$, i.e.,*

$$\lim_{\rho\to\infty} \mathbf{M}^{j'}_{\tau,\kappa,j,:}(\rho+1) = \tilde{\mathbf{M}}_{\tau,\kappa+1,j,:}, \tag{39}$$

$$\lim_{\rho\to\infty} \mathbf{N}^{j'}_{\tau,\kappa,j,:}(\rho+1) = \tilde{\mathbf{N}}_{\tau,\kappa+1,j,:}, \text{ for } j' \in \mathcal{N}_j \cup \{j\} \tag{40}$$

$$\lim_{\rho\to\infty} \boldsymbol{\alpha}^{j}_{\tau,\kappa}(\rho+1) = \tilde{\boldsymbol{\alpha}}_{\tau,\kappa+1},$$

*for $j = 1, \ldots, P$, implying that all local vectors $\{\mathbf{M}^{j'}_{\tau,\kappa,j,:}(\rho+1), \mathbf{N}^{j'}_{\tau,\kappa,j,:}(\rho+1)\}_{j' \in \mathcal{N}_j \cup \{j\}}$ converge to $\tilde{\mathbf{M}}_{\tau,\kappa+1,j,:}$ and $\tilde{\mathbf{N}}_{\tau,\kappa+1,j,:}$ respectively, while all local coefficient vectors $\{\boldsymbol{\alpha}^{j}_{\tau,\kappa}(\rho+1)\}_{j=1}^{P}$*

*converge to the same limit $\tilde{\boldsymbol{\alpha}}_{\tau,\kappa+1}$; thus satisfying the equality constraints in* (18) *and* (19)*. Further,*

$$\lim_{\rho\to\infty}\Bigg[-\frac{\mu}{|\mathcal{S}_Q|}\sum_{j\in\mathcal{S}_Q} w_j^{\tau,\kappa}(\rho)+\frac{\omega}{P}\sum_j[\psi_j^{\tau,\kappa}(\rho)+\xi\theta_j^{\tau,\kappa}(\rho)] \tag{41}$$
$$+\upsilon\sum_{\ell=1}^{P}\left[\|\mathbf{M}^{\ell}_{\tau,\kappa,\ell,:}(\rho)\|_1-\frac{\mathbf{M}^{\ell}_{\tau,\kappa,\ell,:}}{\|\mathbf{M}^{\ell}_{\tau,\kappa,\ell,:}\|_2}(\mathbf{M}^{\ell}_{\tau,\kappa,\ell,:}(\rho))^T\right]\Bigg]=f^*_{\tau,\kappa},$$

$$\lim_{\rho\to\infty}\Bigg[-\frac{\mu}{|\mathcal{S}_Q|}\sum_{j\in\mathcal{S}_Q} w_j^{\tau,\kappa}(\rho)+\frac{\omega}{P}\sum_j[\psi_j^{\tau,\kappa}(\rho)+\xi\theta_j^{\tau,\kappa}(\rho)] \tag{42}$$
$$+\upsilon\sum_{\ell=1}^{P}\left[\|\mathbf{N}^{\ell}_{\tau,\kappa,\ell,:}(\rho)\|_1-\frac{\mathbf{N}^{\ell}_{\tau,\kappa,\ell,:}}{\|\mathbf{N}^{\ell}_{\tau,\kappa,\ell,:}\|_2}(\mathbf{N}^{\ell}_{\tau,\kappa,\ell,:}(\rho))^T\right]\Bigg]=g^*_{\tau,\kappa},$$

*where $f^*_{\tau,\kappa}$ and $g^*_{\tau,\kappa}$ correspond to the minimum values of the costs in* (18) *and* (19) *respectively.*

The result of Prop. (5) combined with Prop. 4 implies that for $\rho\to\infty$ the factor and coefficient iterates $\{\mathbf{M}^j_{\tau,\kappa,j,:}(\rho+1),\mathbf{N}^j_{\tau,\kappa,j,:}(\rho+1),\boldsymbol{\alpha}^j_{\tau,\kappa}(\rho+1)\}_{j=1}^P$ will converge to limits $\tilde{\mathbf{M}}_{\tau,\kappa+1,j,:}$, $\tilde{\mathbf{N}}_{\tau,\kappa+1,j,:}$ and $\tilde{\boldsymbol{\alpha}}_{\tau,\kappa+1}$ such that $\sum_b\tilde{\alpha}_{\tau,\kappa+1,b}\mathbf{A}^b_x$ is block diagonal with rank $Q$ as $\tau,\kappa\to\infty$. Thus, despite the fact that the converging limits of the centralized and distributed algorithms may not be the same, they share the block diagonal property crucial for facilitating the clustering task. In practice a sufficiently large finite number of ADMM iterations $\rho_t<\infty$ suffices to approach convergence. This is demonstrated using the numerical examples following next.

**Computational Complexity:** The computational complexity of our novel framework is compared with the complexity of the tensor multiple kernel graph-based clustering (TMKGC) scheme in (Ren et al., 2020), the structure-preserving multiple kernel clustering approach (SPMKC) in (Ren and Sun, 2020) and the K-means algorithm.

The computational complexity per iteration in TMKGC (Ren et al., 2020) is $\mathcal{O}(BP^2\log_2(P)+B^2P^2)$ where $B$ is the number of elements in the kernel dictionary, $P$ the number of data vectors (or sensing units) and $Q$ the number of underlying sources. The computational complexity per iteration in SPMKC (Ren and Sun, 2020) is $\mathcal{O}(P^3)$. The computational complexity of K-means is $\mathcal{O}(P^2)$.

The computational complexity in the novel distributed kernel clustering (DKC) scheme per node per iteration is mainly accounting for steps D1a-D1c and D2a-D2c. These can be be divided into two tasks at unit $j$: i) Updating the $|\mathcal{N}_j|$ multipliers with a cumulative complexity of $\mathcal{O}(|\mathcal{N}_j|(B+Q))$; and ii) Solve the local SDP problems involved in steps D1c and D2c with a complexity of $\mathcal{O}((B+Q)^4)$ and $\mathcal{O}((Q)^4)$ respectively [see e.g., (Vandenberghe et al., 2005)].

It is practical to assume $B<<P$ and $Q<<P$, since the size of the kernel dictionary $B$ and the number of underlying sources $Q$ are smaller than the number of measurement vectors/ sensing units $P$.

Comparing the computational complexity of DKC and TMKGC, we study under what conditions the following inequality

$$|\mathcal{N}_j|(B+Q)+(B+Q)^4+Q^4<B^2P^2\Leftrightarrow\frac{|\mathcal{N}_j|}{B}(1+\frac{Q}{B})+\left(1+\frac{Q}{B}\right)^2(B+Q)^2+\frac{Q^4}{B^2}<P^2, \tag{43}$$

is satisfied. Taking into account that in practice $P>>B,Q$ and $Q<B$ since the number of elements in the dictionary can be chosen to be larger than the number of sources $Q$ then (43) can be

satisfied if

$$B+Q \leq \frac{\sqrt{P^2-3}}{2} \approx \frac{P}{2}, \tag{44}$$

which holds true for large $P$ and $B, Q << P$.

Comparing the computational complexity of DKC and SPMKC, we study under what conditions the following inequality

$$|\mathcal{N}_j|(B+Q)+(B+Q)^4+Q^4 < P^3 \Leftrightarrow |\mathcal{N}_j|\frac{B+Q}{P^3}+\left(\frac{B+Q}{P}\right)^3(B+Q)+\frac{Q^4}{P^3}<1, \tag{45}$$

is satisfied. The second inequality in (45) can be easily satisfied when $B, Q << P$. Thus, the computational complexity of our approach per unit per iteration is definitely lower when the amount of data vectors $P$ is considerably larger than the cardinality of the dictionary $B$ and the number of sources $Q$.

When studying under what conditions our approach has lower computational complexity per iteration per unit than K-means, i.e.,

$$|\mathcal{N}_j|(B+Q)+(B+Q)^4+Q^4 < P^2 \Leftrightarrow |\mathcal{N}_j|\frac{B+Q}{P^2}+\left(\frac{B+Q}{P}\right)^2(B+Q)^2+\frac{Q^4}{P^2}<1, \tag{46}$$

then the computational complexity of our approach will be lower than the one of K-means when the amount of data vectors $P$ is considerably larger than the cardinality of the dictionary $B$ and the number of sources $Q$.

## 5. Numerical Simulations

Numerical tests are utilized next to study the effectiveness of the proposed algorithmic framework on multiple datasets. We compare the performance of the novel approach with competing alternatives using three different datasets: (1) The Unimib dataset (Micucci et al., 2017) corresponding to a collection of smartphone-based human activity detection readings with $Q = 3$ different classes; (2) The Salinas dataset in (Sal, 2021) which consists of 3-dimensional hyperspectral images with $Q = 4$; and (3) a synthetic dataset with $Q = 4$.

The performance of the novel clustering and learning scheme proposed in this paper (abbreviated as CKC for the centralized version and DKC for the distributed version) is compared with: (1) the tensor multiple kernel graph-based clustering (TMKGC) scheme in (Ren et al., 2020); (2) the structure-preserving multiple kernel clustering approach (SPMKC) in (Ren and Sun, 2020); and (3) the K-means clustering algorithm using Euclidean distance. Note that the distributed K-means approaches in (Chen et al., 2016; Oliva et al., 2013; Qin et al., 2016; Tsapanos et al., 2015) are performing at best similar to the centralized K-means. This is the reason we are using the centralized K-means for comparisons reasons. Further, the only other distributed kernel-learning clustering approach available is the one in (Ren et al., 2020) for which we have performed extensive comparisons. It has to be emphasized though that this comparison is unfair for our approach, in the sense that TMKGC requires a central processing center acting as a fusion center, whereas our framework does not require a fusion center and can operate in ad hoc network architectures.

Three figures of merit are employed to compare the clustering performance of CKC/DKC with the aforementioned alternatives: accuracy, NMI, and purity. Accuracy quantifies the success per-

centage in clustering data in the correct groups

$$\text{Accuracy} = P^{-1} \cdot \sum_{j=1}^{P} \delta(\tilde{C}_j - \text{map}(\hat{C}_j)) \tag{47}$$

where $\hat{C}_j$ is the cluster label returned by the clustering algorithms considered, whereas $\tilde{C}_j$ is the ground truth label, $\delta(\cdot)$ denotes the Kronecker delta function and map$(\hat{C}_j)$ is the mapping of cluster label $\hat{C}_j$ to a class label using the Hungarian assignment algorithm (Kuhn, 1955). NMI quantifies the quality of clusters
NMI$(\Psi, \tilde{\Psi}) = \frac{I(\Psi;\tilde{\Psi})}{\sqrt{H(\Psi)\cdot H(\tilde{\Psi})}}$,
where $I$ and $H$ refer to the mutual information and entropy respectively evaluated as:

$$I(\Psi;\tilde{\Psi}) := \sum_{q=1}^{Q}\sum_{p=1}^{Q} \frac{|\Psi_q \cap \tilde{\Psi}_p|}{P} \log\left[\frac{P\cdot|\Psi_q \cap \tilde{\Psi}_p|}{|\Psi_q|\cdot|\tilde{\Psi}_p|}\right], \tag{48}$$

$$H(\Psi) := \sum_{q=1}^{Q} \frac{|\Psi_q|}{P}\log\left[\frac{|\Psi_q|}{P}\right], H(\tilde{\Psi}) := \sum_{q=1}^{Q} \frac{|\tilde{\Psi}_q|}{P}\log\left[\frac{|\tilde{\Psi}_q|}{P}\right]$$

where $\Psi = \{\Psi_1, \Psi_2, ..., \Psi_Q\}$ correspond to the $Q$ clusters found by the clustering scheme, while $\tilde{\Psi} = \{\tilde{\Psi}_1, \tilde{\Psi}_2, ..., \tilde{\Psi}_Q\}$ denotes the ground-truth cluster sets that contain correctly assigned data entries. It pertains more to a setting where the data labels are given (not the case in our setting) but it is provided for a complete performance analysis.

Purity quantifies the degree at which the found clusters contain elements from one class. Purity is calculated as

$$\text{Purity} = P^{-1} * \sum_{q=1}^{Q} \max_{j=1,...,Q} |\Psi_q \cap \tilde{\Psi}_j|.$$

Each cluster is matched with the class set that has the most overlap and the number of correctly assigned elements is used, i.e., $\max_{j=1,...,Q}|\Psi_q \cap \tilde{\Psi}_j|$.

The performance of DKC will be tested in an ad hoc connected network of sensing units comprising of $P = 60$ units placed randomly in the area $[0, 1] \times [0, 1]$. Two sensing units are connected if their Euclidean distance is less than $0.25$. For the synthetic data, and the Salinas dataset the set of sensors $\mathcal{S}_Q$ whose single-hop neighbors sense all $Q$ clusters has 7 units, whereas in the Unimib dataset it has $8$.

### 5.1 Synthetic Data

The synthetic dataset comprises of $Q = 4$ different classes modeled as a kernel dictionary with six $60 \times 60$ kernel elements $\{\mathbf{A}^1(\mathbf{x})\}_{b=1}^{6}$. Kernel matrix $\mathbf{A}_x^1$ is equal to a $60 \times 60$ random matrix. Kernel matrix $\mathbf{A}_x^2$ consists of two diagonal blocks occupying the first 15 rows and columns, and rows and columns with indices 29 to 46, respectively. Dictionary kernel $\mathbf{A}_x^3$ contains one diagonal block occupying rows with indices 16 to 28 and columns 16 to 28 while the remainder of the entries are set to zeros. Dictionary kernel $\mathbf{A}_x^4$ contains one diagonal block occupying rows with indices 47 to 60 and columns 47 to 60, while the remainder of the entries are set to zeros. Additionally, $\mathbf{A}_x^5$ is set to an all-ones matrix normalized to have trace one, while $\mathbf{A}_x^6$ is set equal to the identity matrix normalized to have trace one.

Fig. 3 depicts the trajectory of the six kernel coefficients $\{\alpha_b\}_{b=1}^6$ versus the DCA number of iterations. Each DCA iteration entails $\rho_t = 5$ ADMM iterations when utilizing Alg. 2 to tackle the tasks in line 6 and 14 of Alg. 1 (DKC). All kernel coefficients are initialized at the same value $\{\alpha_{0,0,b} = 1/6\}_{b=1}^6$. Throughout Sec. 5, the kernel factors $\mathbf{M}_{j,:}^j$ and $\mathbf{N}_{j,:}^j$ for $j \in \mathcal{S}_Q$ are initialized applying CKC locally at each sensing unit $j \in \mathcal{S}_Q$, while the rest of the sensing units $j \notin \mathcal{S}_Q$ set the entries of $\mathbf{M}_{j,:}^j$ and $\mathbf{N}_{j,:}^j$ to zero initially. As the DKC algorithm progresses through iterations it can be seen only the coefficients weighing the block diagonal kernels in the dictionary are converging to a nonzero value $(\alpha_2, \alpha_3, \alpha_4)$, while the rest $\alpha_1, \alpha_5, \alpha_6$ converge to zero since they correspond to kernel matrices that do not promote a block diagonal structure in the synthetic dictionary.

Figs. 4 and 5 depict the accuracy, purity and NMI for DKC vsersus DCA iteration index for various number of ADMM iterations $\rho_t$. The parameters for DKC where set as $v = 0.01$, $\mu = 10, \omega = 2, \xi = 15$ and $c = 0.01$. All methods in this synthetic example are able to reach accuracy, NMI and purity equal to one, though the novel DKC approach does not require the presence of a central processor. Note that as the number of ADMM iterations increases the rate of convergence goes up; especially when increasing from $\rho_t = 1$ to $\rho_t = 3$ beyond which the rate gains are negligible. Fig. 3 (Right) depicts the cumulative disagreement between the local kernel coefficients across all 60 units, i.e., $\sum_{j,j'} \delta_{j,j'}^3$ in (36). It can be seen that the disagreement converges to zero demonstrating that all units estimate the same kernel coefficients and therefore same kernel similarity matrix.

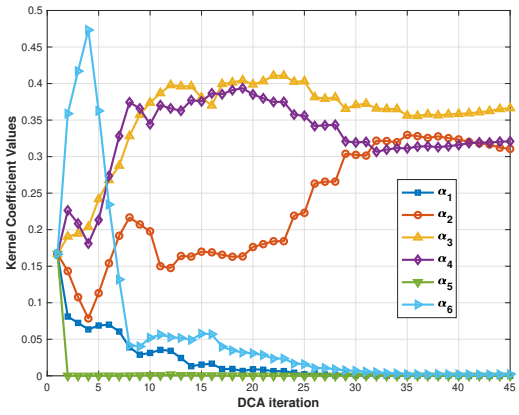


Figure 3: Trajectory of the kernel coefficients $\{\alpha_b\}_{b=1}^6$ vs. DCA iteration index.

### 5.2 Unimib Dataset

The second dataset corresponds to the University of Milano Bicocca Smartphone-based Human Activity Recognition (Unimib) which consists of accelerometer readings of users that participated in activities such as walking, running, and climbing stairs ($Q = 3$). The signals are pre-processed such that the data vectors are grouped into individual epochs with each of them conformed of 51 samples in length and centered around the peak of the epochs. Since the accelerometer readings are considered along all the 3-D axes, the concatenated signal is 153 samples long. In the distributed setting, unit $j$ acquires data entry $\mathbf{x}_t(j)$; 20 sensing units acquire data entries $\mathbf{x}_t(j)$ corresponding

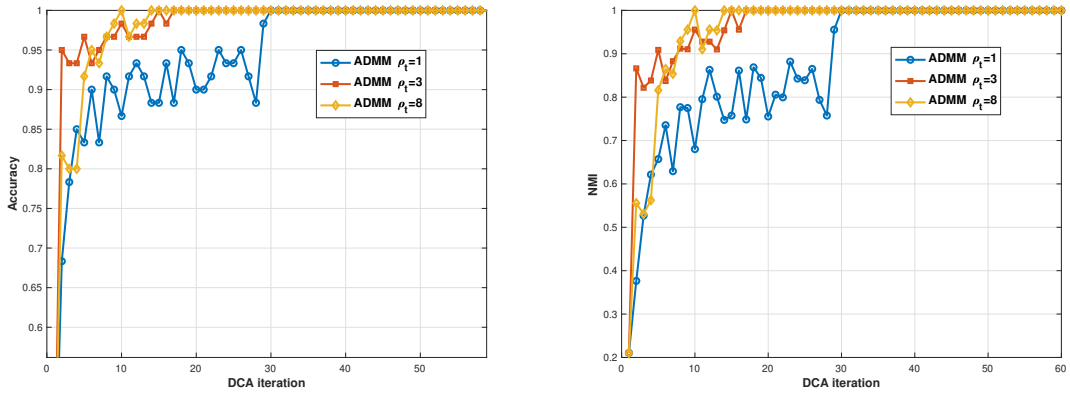


Figure 4: (Left) Accuracy vs. DCA iteration index for different number of ADMM iterations; (Right) NMI vs. DCA iteration index.

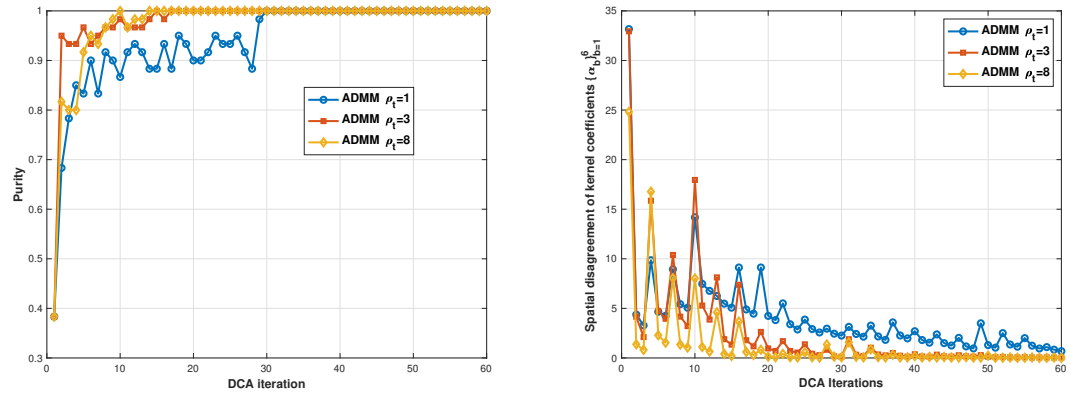


Figure 5: (Left) Purity vs. DCA iteration index for different number of ADMM iterations; (Right) Cumulative spatial disagreement between the local kernel coefficients across all units vs. DCA iteration index.

to walking, 20 units corresponding to running, and 20 units correspond to climbing stairs. The objective is to cluster the signals based on the activity class they belong to.

Figs. 6-8 depict a box plot of the accuracy, NMI and purity respectively of DKC, TMKGC, SPMKC and K-means averaged over 30 independent trials of the Unimib dataset. The parameters utilized for DKC are $v = 0.01$, $\mu = 10, \omega = 2, \xi = 15$ and $c = 0.01$, further the kernel dictionary $\mathcal{D}$ consists of 17 candidate kernel matrices among which 10 are Gaussian with variances in the interval $[10^{-4}, 10^{8}]$ and 7 are polynomial with degrees in $\{1, \ldots, 7\}$. For SPMKC parameter values that gave good performance were $\lambda_1 = 4, \lambda_2 = 1, \lambda_3 = 400$ and $\lambda_4 = 1$, while for TMKGC $\alpha = 0.1, \beta = 10^{-3}$. The red mark inside these box plots indicates the median accuracy for each method and the edges of the box mark the 25 and 75 percentiles of the accuracy across the number of trials utilized.

Fig. 6 and 8 show that both CKC and DKC outperform TMKGC, SPMKC as well as K-Means in terms of accuracy and purity across 30 independent trials of Unimib data. CKC is formulated to learn a block diagonal similarity structure of certain rank in contrast to TMKGC. Interestingly, DKC also outperforms TMKGC showing the effectiveness of the collaboration between single-hop neighboring units in carrying out the clustering task. For the NMI in Fig. 7 TMKGC got the largest value and our novel CKC/DKC framework has the second highest value, outperforming SPMKC and K-Means. One reason is the utilization of tensors in TMKGC that leads to higher-order graph learning, leading to better quality clusters, though at the cost of much higher computational complexity.

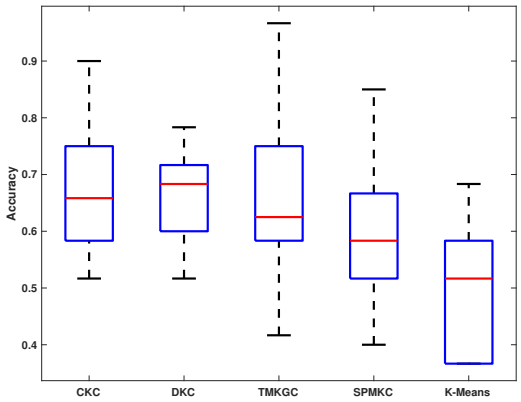


Figure 6: Accuracy box plot over 30 independent trials on the Unimib dataset.

Figs. 9-10 depict the accuracy, NMI, purity and cumulative disagreement of DKC, versus block coordinate descent iteration index $\tau$ for different number of DCA iterations $K$ and ADMM iterations $\rho_t$. For comparison the accuracy, NMI and purity of centralized TMKGC, SPMKC and K-means methods are also provided. The results are depicted for the parameter values selected earlier. One dataset among the 30 independent ones was utilized here to show the convergence properties of DKC. Fig. 9 shows that both the accuracy and NMI, as the number of block iterations $\tau$ increases, converge to the corresponding values achieved by CKC for sufficiently large number of ADMM iterations $\rho_\tau$ and DCA iterations $K$. CKC and TMKGC have close accuracies, though DKC running for limited $K$ and $\rho_t$ falls short of the centralized performance. NMI is still better for

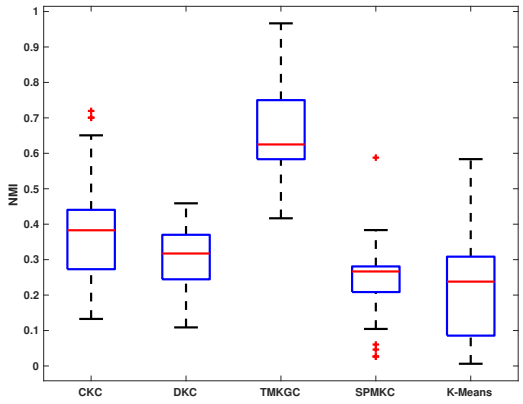


Figure 7: NMI box plot over 30 independent trials on the Unimib dataset.

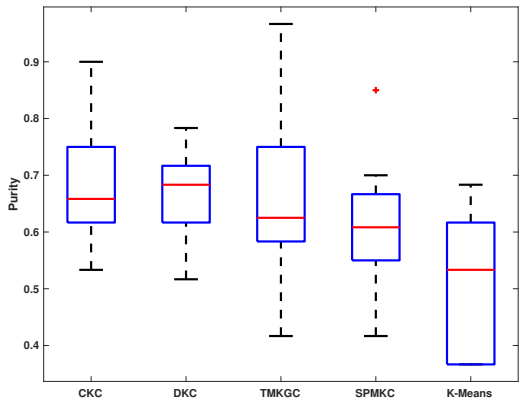


Figure 8: Purity box plot over 30 independent trials on the Unimib dataset.

TMKGC for this dataset. In fact DKC outperforms the clustering performance metrics achieved by the centralized approaches SPMKC and K-Means. This advocates the collaborative nature of our novel approach which allows information to diffuse more and more across the agents as the ADMM iterations increase.

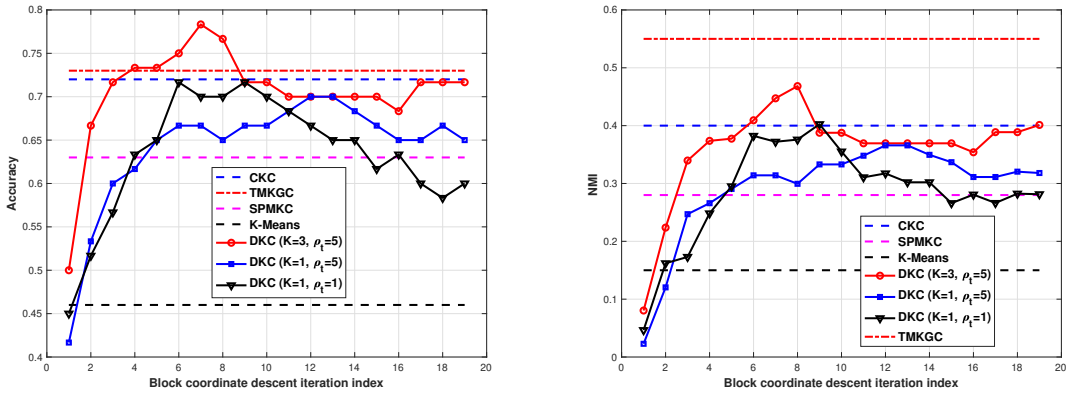


Figure 9: (Left) Accuracy vs. block coordinate iteration index for different number of ADMM and DCA iterations; (Right) NMI vs. iteration index for different number of ADMM and DCA iterations on the Unimib dataset.

Similar conclusions can be drawn for the purity metric in Fig. 10 (Left). Fig. 10 (Right) depicts the cumulative disagreement between the local kernel coefficients vs. block coordinate descent iteration index $\tau$ and similarly to the synthetic scenario as the ADMM iterations $\rho_\tau$ and DCA iterations $K$ increase it gets closer to zero.

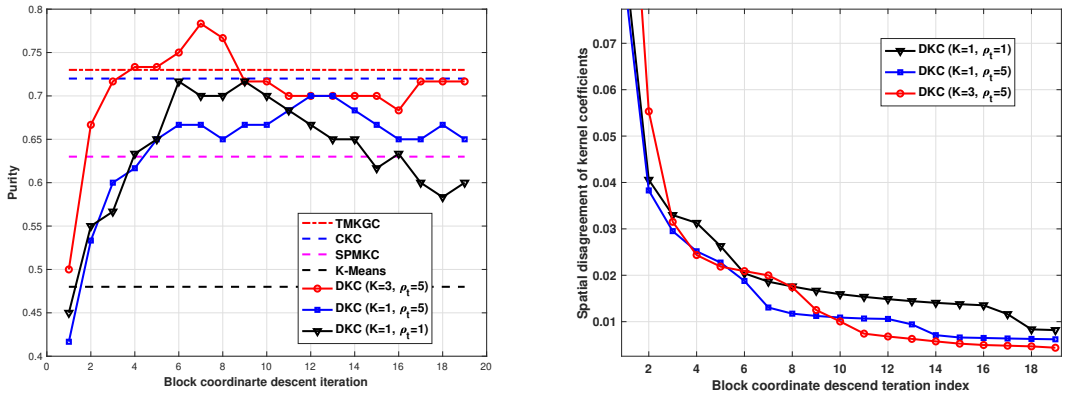


Figure 10: (Left) Purity vs. block coordinate iteration index for different number of ADMM and DCA iterations; (Right) Cumulative spatial disagreement between the local kernel coefficients across all units vs. block coordinate iteration index for different number of ADMM and DCA iterations on the Unimib dataset.

### 5.3 Salinas Dataset

This is a hyperspectral image dataset captured by an Aviris sensing system over the Salinas valley, California. These are primarily farmland images that indicate the presence of different crops/materials in different parts of the images. Each sensing unit measurement vector $\mathbf{x}_i$ has 224 entries; each of these vectors will be clustered into different groups based on the crop/material they belong to. A total of $Q = 4$ different randomly selected materials were considered and 25 random pixels were chosen.

In the distributed setting, unit $j$ acquires data entry $\mathbf{x}_t(j)$ for $j = 1, \ldots, 60$; 15 sensing units acquire data entries $\mathbf{x}_t(j)$ corresponding to the first material, 13 units corresponding to the second material, 18 units correspond to the third material and 14 units to the fourth material. The objective is to cluster the signals based on the crop/material class they belong to. The parameters utilized for DKC are $v = 0.01$, $\mu = 10, \omega = 2, \xi = 15$ and $c = 1$. Further, the kernel dictionary $\mathcal{D}$ consists of 17 candidate kernel matrices among which 10 are Gaussian with variances in the interval $[10^{-4}, 10^{8}]$ and 7 are polynomial with degrees in $\{1, \ldots, 7\}$. For SPMKC values that gave good performance were $\lambda_1 = 6, \lambda_2 = 1, \lambda_3 = 600$ and $\lambda_4 = 1$, while for TMKGC $\alpha = 0.1, \beta = 10^{-4}$.

Figs. 11-13 depict a box plot of the accuracy, purity and NMI respectively of DKC, TMKGC, SPMKC and K-means averaged over 50 independent trials of the Salinas dataset. The red mark inside these box plots indicates the median accuracy for each method and the edges of the box mark the 25 and 75 percentiles of the accuracy across the number of trials utilized.

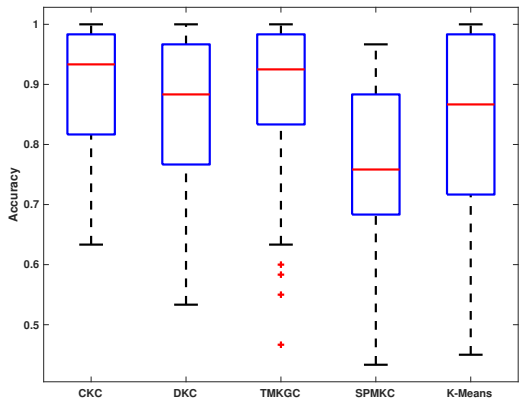


Figure 11: Accuracy box plot over 50 independent trials on the Salinas dataset.

Fig. 11 and 13 show that both CKC outperforms TMKGC, SPMKC as well as K-Means in terms of accuracy and purity. DKC accuracy gets really close to the centralized one achieved by CKC. For the NMI, which here is given for completeness since labels are not available, TMKGC got the largest value, though CKC and DKC get really close and outperform SPMKC and K-Means. NMI is better in TMKGC again due to the usage of a tensor based formulation extracting higher-order graph similarities.

Figs. 14-15 depict the accuracy, NMI, purity and cumulative disagreement of DKC, versus block coordinate descent iteration index $\tau$ for different number of DCA iterations $K$ and ADMM iterations $\rho_t$. For comparison the accuracy, NMI and purity of centralized TMKGC, SPMKC and K-

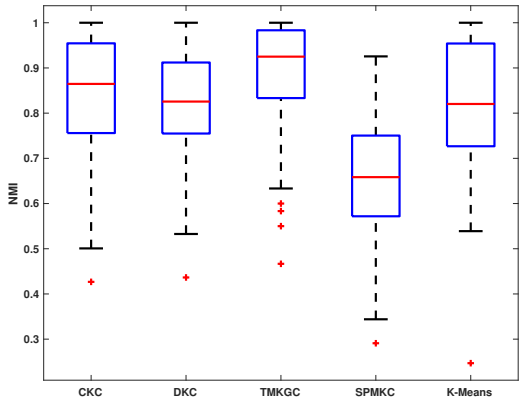


Figure 12: NMI box plot over 50 independent trials on the Salinas dataset.

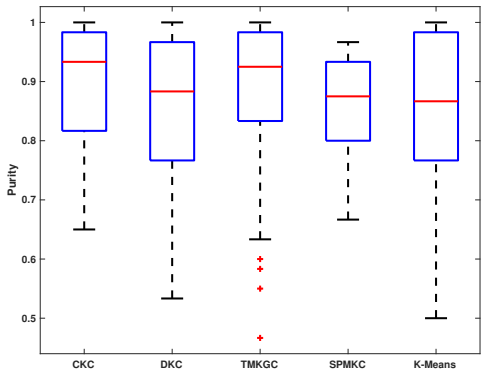


Figure 13: Purity box plot over 50 independent trials on the Salinas dataset.

means methods are also provided. The results are depicted for the parameter values selected earlier. One dataset among the 50 independent Salinas datasets was utilized here to show the convergence properties of DKC. Fig. 14 shows that both accuracy and NMI, as the number of block iterations $\tau$ increases, approach closely the corresponding values achieved by CKC as the number of ADMM iterations $\rho_\tau$ and DCA iterations $K$ increase. In fact DKC outperforms the clustering performance metrics achieved by the centralized approaches TMKGC, SPMKC and K-Means for sufficiently large number of ADMM and DCA iterations. The results here are depicted for one dataset among the 50 realizations for which case CKC outperforms TMKGC. Again the objective is to demonstrate the potential of DKC in getting close to the CKC performance and potentially outperforming alternative centralized approaches.

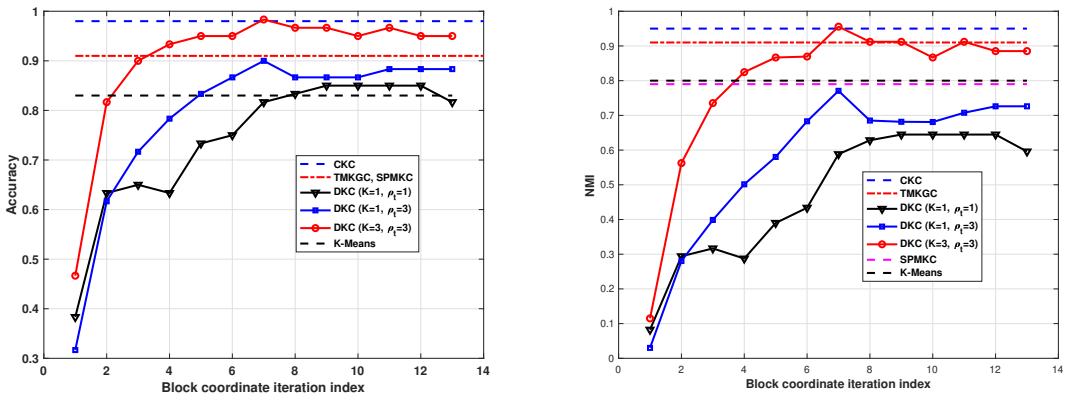


Figure 14: (Left) Accuracy vs. block coordinate iteration index for different number of ADMM and DCA iterations; (Right) NMI vs. iteration index for different number of ADMM and DCA iterations on the Salinas dataset.

Similar conclusions can be drawn for the purity metric in Fig. 15 (Left). Fig. 15 (Right) depicts the cumulative spatial disagreement between the local kernel coefficients vs. block coordinate descent iteration index $\tau$ and similarly to the synthetic scenario as the ADMM iterations $\rho_\tau$ and DCA iterations $K$ increase it gets closer to zero. Again Figs. 14-15 advocate the collaborative nature of our novel approach which allows information to diffuse more and more across the network as the ADMM iterations increase.

## 6. Conclusions

A novel distributed framework for joint kernel learning and clustering was derived capable of determining clustering configurations in an unsupervised manner. Utilizing principles from semidefinite programming, block coordinate descent and difference of convex formulations we arrive at a minimization formulation that facilitates the selection of proper block diagonal kernel similarity matrices that allow effective unsupervised data clustering. The SDP problems solved during a block coordinate cycle are further reformulated in a separable fashion allowing the application of ADMM which leads to a fully distributed joint kernel learning and clustering approach. Convergence guarantees demonstrate that the novel framework promotes the construction of block diagonal data similar-

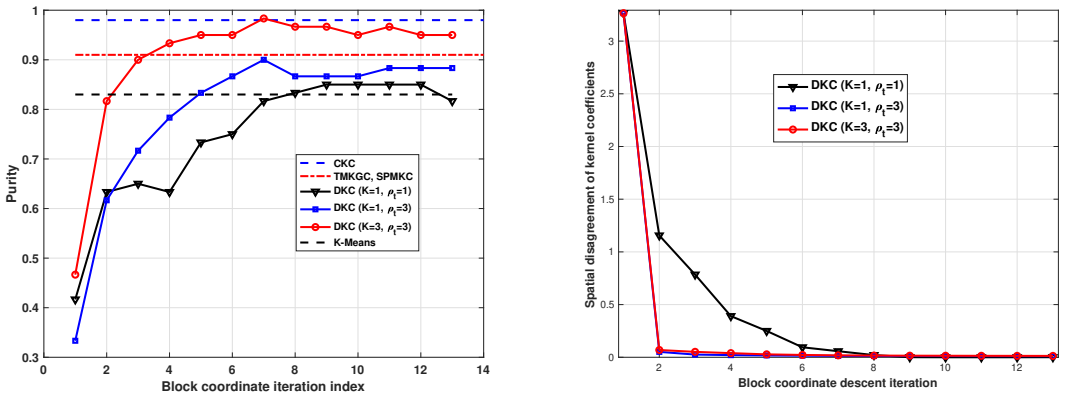


Figure 15: (Top) Purity vs. block coordinate iteration index for different number of ADMM and DCA iterations; (Bottom) Cumulative disagreement between the local kernel coefficients across all sensing units vs. block coordinate iteration index for different number of ADMM and DCA iterations on the Salinas dataset.

ity matrices. Detailed numerical examples show the superior clustering accuracy achieved by the distributed framework over existing alternatives. Last but not least, it should be emphasized that multi-hop communication typically comes with severe challenges on message reception, scheduling and routing to name a few. In this work we emphasized more on the learning aspects of the clustering problem. Moreover, kernel dictionary selection and unknown number of sources are also practical aspects that should be considered. Future work will focus on addressing the aforementioned difficulties to generalize the capabilities of our novel clustering framework.

## Acknowledgments

The work in this paper was supported by ARO grant W911NF-21-1-0231.

## Appendix A. Proof of Proposition 1

Assume that the dictionary of kernels $\mathcal{D} := \{\mathbf{A}_x^b\}_{b=1}^B$ contains elements for which there exist non-negative scalars $\beta_1, \ldots, \beta_B$ with $\sum_{b=1}^B \beta_b = 1$ such that $\sum_{b=1}^B \beta_b \mathbf{A}_x^b$ is block diagonal with $Q$ diagonal blocks and equal rank. We show that the formulation in (4) is able to find a set of optimal $\alpha_b^*$ coefficients such that $\sum_{b=1}^B \alpha_b^* \mathbf{A}_x^b$ is also block diagonal with $Q$ diagonal blocks each of rank one. Employing the Schur complement (Boyd and Vandenberghe, 2004) the first matrix inequality in (4) is rewritten

$$\left[ \begin{array}{cc} \mathbf{I}_Q & \mathbf{M}^T \\ \mathbf{M} & \sum_{b=1}^B \alpha_b \mathbf{A}_x^b \end{array} \right] \succeq \mathbf{0} \Leftrightarrow \sum_{b=1}^B \alpha_b \mathbf{A}_x^b \succeq \mathbf{M} \cdot \mathbf{M}^T. \tag{49}$$

The fourth matrix inequality $(\mathbf{N}^T\mathbf{M} + \mathbf{M}^T\mathbf{N}) \geq w \cdot \mathbf{I}_Q, \;\; w > 0$ in (4) implies that the rank of $\mathbf{M}$ and $\mathbf{N}$ should be equal to Q. To prove this consider the singular value decomposition of $\mathbf{M}^T\mathbf{N} = \mathbf{U}_{mn}\mathbf{\Sigma}_{mn}\mathbf{V}_{mn}^T$ where $\mathbf{U}_{mn}$ and $\mathbf{V}_{mn}$ are $Q \times Q$ orthonormal matrices, while $\mathbf{\Sigma}_{mn}$ is a $Q \times Q$ diagonal matrix. Let us assume that rank$(\mathbf{M}^T\mathbf{N}) = Q - Z < Q$, and let $\mathbf{V}_Z \in \mathbb{R}^{Q \times Z}$ correspond to these columns of $\mathbf{V}_{mn}$ that span the nullspace of $\mathbf{M}^T\mathbf{N}$; note that the dimensionality of nullpace$(\mathbf{M}^T\mathbf{N})$ is equal to $Z > 0$. Given that $\mathbf{V}_Z^T\mathbf{V}_Z = \mathbf{I}_Z$ we multiply the left and right parts of the matrix inequality in (4) from the left with $\mathbf{V}_Z^T$ and the right with $\mathbf{V}_Z$ to obtain

$$\begin{aligned} &\mathbf{V}_Z^T(\mathbf{U}_{mn}\mathbf{\Sigma}_{mn}\mathbf{V}_{mn}^T + \mathbf{V}_{mn}\mathbf{\Sigma}_{mn}\mathbf{U}_{mn}^T)\mathbf{V}_Z \succeq w \cdot \mathbf{I}_Z \\ &\Leftrightarrow \mathbf{V}_Z^T\mathbf{U}_{mn}\mathbf{\Sigma}_{mn}\mathbf{V}_{mn}^T\mathbf{V}_Z + \mathbf{V}_Z^T\mathbf{V}_{mn}\mathbf{\Sigma}_{mn}\mathbf{U}_{mn}^T\mathbf{V}_Z \succeq w \cdot \mathbf{I}_Z. \end{aligned} \tag{50}$$

Assuming that $\mathbf{M}^T\mathbf{N}$ is rank deficient, then the nullspace of $\mathbf{M}^T\mathbf{N}$ is nonempty with dimensionality $Z \geq 1$. Then, it follows that the lhs side of the second inequality in (50) will be zero, and $\mathbf{0}_{Z \times Z} \succeq w \cdot \mathbf{I}_Z$ which cannot be true since $w > 0$. Therefore matrix $\mathbf{M}^T\mathbf{N}$ has full rank equal to $Q$, therefore both factors $\mathbf{M}$ and $\mathbf{N}$ have full rank equal to $Q$ when $w > 0$ [for sufficiently large coefficient $\mu$ in (4)].

Thus, from (49) $\sum_{b=1}^B \alpha_b^* \mathbf{A}_x^b$ will have rank at least $Q$ for $w > 0$. The second matrix inequality in (4) gives

$$\left[ \begin{array}{cc} \frac{\theta}{\sqrt{Q}}\mathbf{I}_P & \mathbf{M} - \mathbf{N} \\ (\mathbf{M} - \mathbf{N})^T & \frac{1}{\sqrt{Q}}\mathbf{I}_Q \end{array} \right] \succeq 0 \Leftrightarrow \|\mathbf{M} - \mathbf{N}\|_F^2 \leq \theta, \tag{51}$$

which for sufficiently large $\xi$ ensures that $\mathbf{M}^*$ and $\mathbf{N}^*$ are approximately equal.

From the Schur complement the third matrix inequality in (4)

$$\begin{aligned} &\left[ \begin{array}{cc} \frac{\psi}{\sqrt{P}}\mathbf{I}_P & \sum_{b=1}^B \alpha_b \mathbf{A}_x^b - \mathbf{M} \cdot \mathbf{N}^T \\ (\sum_{b=1}^B \alpha_b \mathbf{A}_x^b - \mathbf{M} \cdot \mathbf{N}^T)^T & \frac{1}{\sqrt{P}}\mathbf{I}_P \end{array} \right] \succeq 0 \\ &\Leftrightarrow \left\| \sum_{b=1}^B \alpha_b \mathbf{A}_x^b - \mathbf{M} \cdot \mathbf{N}^T \right\|_F^2 \leq \psi. \end{aligned} \tag{52}$$

Optimal $\psi^* = 0$, since there exists a convex combination $\sum_{b=1}^B \beta_b \mathbf{A}_{x^b}$ which has rank $Q$. $\psi^* = 0$ can be achieved by choosing optimal $\alpha_b^*$ that result a rank $Q$ kernel $\sum_{b=1}^B \alpha_b^* \mathbf{A}_x^b$, while there exist optimal factors $\mathbf{M}^*, \mathbf{N}^*$ such that $\sum_{b=1}^B \alpha_b^* \mathbf{A}_x^b = \mathbf{M}^*(\mathbf{N}^*)^T$ which results $\psi^* = 0$.

Finally the term $v \cdot \sum_{\ell=1}^{P} [\|\mathbf{M}_{\ell,:}\|_1 - \|\mathbf{M}_{\ell,:}\|_2] \geq 0$ can get its lowest value of zero if factor $\mathbf{M}^*$ has at most one nonzero element on each row $\mathbf{M}^*_{\ell,:}$ for $\ell = 1, \ldots, Q$ (cf. Sec. 3). The same holds for term $v \cdot \sum_{\ell=1}^{P} [\|\mathbf{N}_{\ell,:}\|_1 - \|\mathbf{N}_{\ell,:}\|_2] \geq 0$.

Given the $P \times Q$ factors $\mathbf{M}^*, \mathbf{N}^*$ have rank $Q$, this implies the presence of $Q$ linearly independent columns in $\mathbf{M}^*$, $\mathbf{N}^*$. Thus, the nonzero elements in each column of $\mathbf{M}^*$ should be in nonoverlaping positions resulting a block diagonal matrix $\sum_{b=1}^{B} \alpha_b^* \mathbf{A}_x^b$ with $Q$ blocks with positions corresponding to the support of each of the $Q$ columns of $\mathbf{M}^*$ and $\mathbf{N}^*$. Every row $\mathbf{M}^*_{\ell,:}$ (and $\mathbf{N}^*_{\ell,:}$ ) should have exactly one nonzero entry, otherwise there is at least one diagonal block with rank greater than 1 resulting rank$(\sum_{b=1}^{B} \alpha_b^* \mathbf{A}_x^b) > Q$ which is not true. □

## Appendix B. Proof of Lemma 2

We establish that the cost function, namely $J_{M,\tau,\kappa}(\cdot)$, in (6) is bounded below by a finite negative number for any $\tau$ and $\kappa$ [similar arguments can be applied for (4)]. Note that variable $w_{\tau,\kappa+1} \geq 0$. For the variable $\psi$, after using Schur complement in the third LMI in (6), we obtain

$$\left\| \sum_{b=1}^{B} \alpha_{\tau,b} \mathbf{A}_x^b - \mathbf{M}_{\tau,\kappa+1} \mathbf{N}_\tau^T \right\|_F^2 \leq \psi_{\tau,\kappa+1}, \tag{53}$$

from which $\psi_{\tau,\kappa+1} \geq 0$. Similarly, $\theta_{\tau,\kappa+1} \geq 0$. From the constraints in (6)

$$\begin{aligned} w_{\tau,\kappa+1} &\leq Q^{-1} \text{trace}((\mathbf{M}_{\tau,\kappa+1})^T \mathbf{N}_\tau) \\ &\leq \textstyle\sum_{\ell=1}^{P} \|\mathbf{N}_{\tau,\ell,:}\|_2 \|\mathbf{M}_{\tau,\kappa+1,\ell,:}\|_2 \\ &\leq \textstyle\sum_{\ell=1}^{P} \max(\|\mathbf{N}_{\tau,\ell,:}\|_2^2, \|\mathbf{M}_{\tau,\kappa+1,\ell,:}\|_2^2) \\ &\leq \textstyle\sum_{\ell=1}^{P} [\|\mathbf{N}_{\tau,\ell,:}\|_2^2 + \|\mathbf{M}_{\tau,\kappa+1,\ell,:}\|_2^2]. \end{aligned} \tag{54}$$

The second inequality in (54) follows from the Cauchy-Schwarz inequality (Horn and Johnson, 2012). Now, from the first LMI in (6) it follows that trace$(\mathbf{M}_{\tau,\kappa+1}(\mathbf{M}_{\tau,\kappa+1})^T) \leq$ trace$(\sum_b \alpha_{\tau+1,b} \mathbf{A}_x^b)$ $= 1$, since trace$(\mathbf{A}_x^b) = 1$ and $\sum_b \alpha_b = 1$. The second LMI in (6) gives

$$\|\mathbf{M}_{\tau,\kappa+1} - \mathbf{N}_\tau\|_F^2 \leq \theta_{\tau,\kappa+1}, \tag{55}$$

from which it follows that $\mathbf{N}_\tau = \mathbf{M}_{\tau,\kappa+1} + \mathbf{E}_{\tau,\kappa+1}$ with $\|\mathbf{E}_{\tau,\kappa+1}\|_F^2 \leq \theta_{\tau,\kappa+1}$. Then, it follows that

$$\begin{aligned} \|\mathbf{N}_\tau\|_F^2 &\leq \|\mathbf{M}_{\tau,\kappa+1}\|_F^2 + \|\mathbf{E}_{\tau,\kappa+1}\|_F^2 + 2\text{trace}(\mathbf{M}_{\tau,\kappa+1}^T \mathbf{E}_{\tau,\kappa+1}) \\ &\leq 1 + \theta_{\tau,\kappa+1} + 2 \cdot \sum_{\ell=1}^{P} [\|\mathbf{M}_{\tau,\kappa+1,\ell,:}\|_2^2 + \|\mathbf{E}_{\tau,\kappa+1,\ell,:}\|_2^2] \\ &\leq 3 + 3\theta_{\tau,\kappa+1}, \end{aligned} \tag{56}$$

where the second and third inequalities in (56) resulted from (54), $\|\mathbf{E}_{\tau,\kappa+1}\|_F^2 \leq \theta_{\tau,\kappa+1}$ and $\|\mathbf{M}_{\tau,\kappa+1}\|_F^2 \leq 1$. Combining (54) and (56) it follows for any $\tau, \kappa$ that

$$w_{\tau,\kappa+1} \leq Q^{-1} \cdot [4 + 3\theta_{\tau,\kappa+1}]. \tag{57}$$

By increasing $\xi$, $\theta_{\tau,\kappa+1}$ can be made small and finite. Next we demonstrate that terms $\|\mathbf{M}_{\ell,:}\|_1 - \frac{\mathbf{M}_{\tau,\kappa,\ell,:}}{\|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2}\mathbf{M}_{\ell,:}^T$ are nonnegative. Specifically

$$\begin{aligned}&\mathbf{M}_{\tau,\kappa,\ell,:}\mathbf{M}_{\ell,:}^T \leq \|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2 \cdot \|\mathbf{M}_{\ell,:}\|_2 \leq \|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2 \cdot \|\mathbf{M}_{\ell,:}\|_1 \\ &\Rightarrow \|\mathbf{M}_{\ell,:}\|_1 - \frac{\mathbf{M}_{\tau,\kappa,\ell,:}}{\|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2}\mathbf{M}_{\ell,:}^T \geq 0. \end{aligned} \tag{58}$$

The first inequality in (58) is direct application of the Cauchy-Schwarz inequality (Horn and Johnson, 2012), whereas the second inequality is the result of the $\ell_1$ norm of a vector been larger than or equal to the $\ell_2$ norm. The second line in (58) follows after dividing both sides of the first line with $\|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2$. Note that $\|\mathbf{M}_{\ell,:}\|_1 - \frac{\mathbf{M}_{\tau,\kappa,\ell,:}}{\{\mathbf{M}_{\tau,\kappa,\ell,:}\|_2}\mathbf{M}_{\ell,:} = 0$ if and only if $\mathbf{M}_{\ell,:}$ has one nonzero element in which case $\|\mathbf{M}_{\ell,:}\|_2 = \|\mathbf{M}_{\ell,:}\|_1$. Thus, we conclude that the cost function $J_{M,\tau,\kappa}(\cdot) \geq -\mu \cdot Q^{-1} \cdot [4 + 3\theta_{\tau,\kappa}]$ and therefore is bounded below from a finite negative value. The same process is applied for the cost in (4) and (8).

If the cost function $J_{M,\tau,\kappa+1}(\cdot)$ reaches a negative value that implies that $w_{\tau,\kappa+1} > 0$ otherwise the cost would be positive. This further implies from (6) that

$$\mathbf{0}_{Q\times Q} \preceq w_{\tau,\kappa+1} \cdot \mathbf{I}_Q \preceq 0.5 \cdot (\mathbf{N}_\tau^T\mathbf{M}_{\tau,\kappa+1} + \mathbf{M}_{\tau,\kappa+1}\mathbf{N}_\tau^T). \tag{59}$$

Using the line of arguments between eqs. (49) and (51) it follows that (59) guarantees that $\mathbf{M}_{\tau,\kappa+1}$ has full rank $Q$. Similar arguments can be applied for the costs in (4) and (8). □

## Appendix C. Proof of Proposition 3

Let $J_{\tau,0}(\mathbf{M}, w, \psi, \theta, \boldsymbol{\alpha}, \mathbf{N}_\tau)$ denote the cost function in (4), when fixing factor $\mathbf{N}$ with $\mathbf{N}_\tau$. Utilizing Thm 3. in (Tao and An, 1997), it turns out that Steps 1a and 1b of Alg. 1 applied to minimize convex formulation (6) for fixed $\mathbf{N} = \mathbf{N}_\tau$, result a decreasing sequence of cost values, i.e., for $\kappa = 0, 1, \ldots$

$$\begin{aligned}&J_{\tau,\kappa+1}(\mathbf{M}_{\tau,\kappa+1}, \boldsymbol{\alpha}_{\tau,\kappa+1}, w_{\tau,\kappa+1}, \psi_{\tau,\kappa+1}, \theta_{\tau,\kappa+1}, \mathbf{N}_\tau) \\ &\leq J_{\tau,\kappa}(\mathbf{M}_{\tau,\kappa}, \boldsymbol{\alpha}_{\tau,\kappa}, w_{\tau,\kappa}, \psi_{\tau,\kappa}, \theta_{\tau,\kappa}, \mathbf{N}_\tau). \end{aligned} \tag{60}$$

Lemma 2 and (60) (monotone convergence theorem) imply

$$\lim_{\kappa\to\infty} J_{\tau,\kappa+1}(\cdot) = J_\tau, \tag{61}$$

where $J_\tau$ is finite. Thus, there exists a sufficiently large iteration index $K \geq 0$ such that $|J_{\tau,\kappa+1}(\cdot) - J_{\tau,\kappa}(\cdot)| \leq \epsilon, \forall\ k \geq K$. Since cost $J_{\tau,\kappa+1}(\cdot)$ is a continuous function there exists such integer $K$ such that $\|\mathbf{M}_{\tau,\kappa+1} - \mathbf{M}_{\tau,\kappa}\|_F \leq \epsilon_1$ and $\|\boldsymbol{\alpha}_{\tau,\kappa+1} - \boldsymbol{\alpha}_{\tau,\kappa}\|_F \leq \epsilon_1$ for all $\kappa \geq K$. This follows also from the convergence properties of DCA in (Tao and An, 1997).

At iteration $\kappa = K$ the stopping criterion will be satisfied at the end of Step 1b. At this point the iterates will be $\mathbf{M}_{\tau+1}, \boldsymbol{\alpha}_{\tau+1}, \mathbf{N}_\tau, w_{\tau,K}, \psi_{\tau,K}$ and $\theta_{\tau,K}$. The corresponding cost function value of (4) will be equal to $J_{\tau,K} = -\mu\cdot w_{\tau,K} + \omega\cdot\psi_{\tau,K} + \xi\cdot\theta_{\tau,K} + v\cdot\sum_{\ell=1}^P[\|\mathbf{M}_{\tau+1,\ell,:}\|_1 - \|\mathbf{M}_{\tau+1,\ell,:}\|_2] + v\cdot\sum_{\ell=1}^P[\|\mathbf{N}_{\tau,\ell,:}\|_1 - \|\mathbf{N}_{\tau,\ell,:}\|_2]$.

During iteration $\tau$ and before Steps 2a and 2b are recursively carried out, the initial cost value for (4), is equal to $J_{\tau,K}$ calculated for using the most recent updates $\mathbf{M}_{\tau+1}, \boldsymbol{\alpha}_{\tau+1}, \mathbf{N}_\tau, w_{\tau,K}, \psi_{\tau,K}$

and $\theta_{\tau,K}$. During Steps 2a and 2b factor $\mathbf{M} = \mathbf{M}_{\tau+1}$, kernel coefficients $\boldsymbol{\alpha} = \boldsymbol{\alpha}_{\tau+1}$ and the cost in (4) is minimized wrt $\mathbf{N}$, $w, \psi$ and $\theta$. Again using Thm. 3 in (Tao and An, 1997), it turns out that Steps 2a and 2b of Alg. 1 applied to minimize convex formulation (4) for fixed $\mathbf{M}$, and $\boldsymbol{\alpha}$ result a decreasing sequence of cost values, i.e., for $\kappa = 0, 1, \ldots$

$$\begin{aligned} &J_{\tau,\kappa+1}(\mathbf{M}_{\tau+1}, \boldsymbol{\alpha}_{\tau+1}, w_{\tau,\kappa+1}, \psi_{\tau,\kappa+1}, \theta_{\tau,\kappa+1}, \mathbf{N}_{\tau,\kappa+1}) \\ &\leq J_{\tau,\kappa}(\mathbf{M}_{\tau+1}, \boldsymbol{\alpha}_{\tau+1}, w_{\tau,\kappa}, \psi_{\tau,\kappa}, \theta_{\tau,\kappa}, \mathbf{N}_{\tau,\kappa}), \end{aligned} \tag{62}$$

while $J_{\tau,0}(\mathbf{M}_{\tau+1}, \boldsymbol{\alpha}_{\tau+1}) = J_{\tau,K}$ when Steps 2a and 2b start. Thus, there exists integer $K'$ such that $\|\mathbf{N}_{\tau,\kappa+1} - \mathbf{N}_{\tau,\kappa}\|_F \leq \epsilon_1$ for all $\kappa \geq K'$. From (60) and (62) the cost function values iterates $J_\tau(\mathbf{M}_\tau, \mathbf{N}_\tau, \boldsymbol{\alpha}_\tau)$ for (4) form a decreasing sequence, since the cost is lower bounded from below the cost iterates converge to a finite value. The continuity of cost (4), implies that a sufficiently large $T$ exists such that $\|\mathbf{M}_{\tau+1} - \mathbf{M}_\tau\|_F + \|\mathbf{N}_{\tau+1} - \mathbf{N}_\tau\|_F + \|\boldsymbol{\alpha}_{\tau+1} - \boldsymbol{\alpha}_\tau\|_F < \epsilon_2$ for $\tau > T$. □

## Appendix D. Proof of Proposition 4

From the equality constraints in the last two lines of (18) and (19) notice that $\mathbf{M}^{j'}_{j,:} = \mathbf{M}_{j,:}$, $\mathbf{N}^{j'}_{j,:} = \mathbf{N}_{j,:}$ and $\alpha^j_b = \alpha_b$ for all $j' \in \mathcal{N}_j$ and $j = 1, \ldots, P$ and $b = 1, \ldots, B$. Next, we simplify notation accordingly. Using similar arguments as in the last two paragraphs of Apdx. B, it follows that the set of local LMIs $[\mathbf{H}^2_\tau(\mathbf{N}_\tau)]_{\mathcal{N}_j} \succeq w_j \cdot \mathbf{I}_Q$ for $j \in \mathcal{S}_Q$ for $w_j > 0$ in (18) ensures that a minimizing factor $\mathbf{M}^*$ will satisfy

$$0.5 \cdot \textstyle\sum_{j' \in \mathcal{N}_j \cup \{j\}} [(\mathbf{N}^j_{\tau,j',:})^T \mathbf{M}^*_{j',:} + (\mathbf{M}^*_{j',:})^T \mathbf{N}^j_{\tau,j',:}] \succeq w_j \cdot \mathbf{I}_Q \succ \mathbf{0},$$

which further implies that $(|\mathcal{N}_j|+1) \times Q$ factor submatrix $\mathbf{M}^*_{\mathcal{N}_j} := \left[(\mathbf{M}^*_{j,:})^T, (\mathbf{M}^*_{j',:})^T, \ldots, (\mathbf{M}^*_{j'',:})^T\right]^T$, with $j, j', j'' \in \mathcal{N}_j$, has rank equal to $Q$, thus $\text{rank}(\mathbf{M}^*) = Q$. Similarly the third LMI in (19) guarantees that a minimizing $\mathbf{N}^*$ will also have rank equal to $Q$.

The local LMI constraint $\mathbf{G}_{1,j} \succeq 0$ can be rewritten as $\sum_{b=1}^B \alpha_b [\mathbf{A}^b_x]_{\mathcal{N}_j} - \mathbf{M}_{\mathcal{N}_j} \cdot (\mathbf{M}_{\mathcal{N}_j})^T \succeq \mathbf{0}$ which for an optimal set of variables $\mathbf{M}^*$ and $\{\alpha^*_b\}_{b=1}^B$ produces

$$\begin{aligned} &\sum_{b=1}^B \alpha^*_b [\mathbf{A}^b_x]_{\mathcal{N}_j} - \mathbf{M}^*_{\mathcal{N}_j} \cdot (\mathbf{M}^*_{\mathcal{N}_j})^T \\ &= \mathbf{E}_{\mathcal{N}_j} \left[ \sum_{b=1}^B \alpha^*_b [\mathbf{A}^b_x] - \mathbf{M}^* \cdot (\mathbf{M}^*)^T \right] \mathbf{E}^T_{\mathcal{N}_j} \succeq \mathbf{0}, \end{aligned} \tag{63}$$

where $\mathbf{E}_{\mathcal{N}_j}$ is a $(|\mathcal{N}_j| + 1) \times P$ matrix where each row has a single nonzero entry equal to one. The column indices for those nonzero entries in $\mathbf{E}_{\mathcal{N}_j}$ are in the set $\mathcal{N}_j \cup \{j\}$. The second row in (63) is indicating that $\sum_{b=1}^B \alpha^*_b [\mathbf{A}^b_x]_{\mathcal{N}_j} - \mathbf{M}^*_{\mathcal{N}_j} \cdot (\mathbf{M}^*_{\mathcal{N}_j})^T$ is a submatrix of $\sum_{b=1}^B \alpha^*_b [\mathbf{A}^b_x] - \mathbf{M}^* \cdot (\mathbf{M}^*)^T$. Given that $\sum_{b=1}^B \alpha^*_b [\mathbf{A}^b_x]_{\mathcal{N}_j} - \mathbf{M}^*_{\mathcal{N}_j} \cdot (\mathbf{M}^*_{\mathcal{N}_j})^T \succeq \mathbf{0}$ and $\text{rank}(\mathbf{M}^*_{\mathcal{N}_j} \cdot (\mathbf{M}^*_{\mathcal{N}_j})^T) = Q$, then it follows that $\text{rank}(\sum_{b=1}^B \alpha^*_b [\mathbf{A}^b_x]_{\mathcal{N}_j}) \geq Q$ which ensures that $\text{rank}(\sum_{b=1}^B \alpha^*_b [\mathbf{A}^b_x]) \geq Q$ since $\sum_{b=1}^B \alpha^*_b [\mathbf{A}^b_x]_{\mathcal{N}_j}$ is a submatrix of $\sum_{b=1}^B \alpha^*_b [\mathbf{A}^b_x]$.

If the kernel dictionary $\mathcal{D}$ contains a unique subset of kernels whose convex combination is block diagonal with $Q$ diagonal blocks of rank 1, then there exists kernel coefficients $\{\alpha^*_b\}_{b=1}^B$ and

$P \times Q$ factors $\mathbf{M}^*, \mathbf{N}^*$ such that $\sum_{b=1}^{B} \alpha_b^* [\mathbf{A}_x^b] = \mathbf{M}^* (\mathbf{N}^*)^T$ from which it follows that for sufficiently large $\omega$, $\mathbf{G}_{\tau,3,j}(\mathbf{N}^*) = \mathbf{0}$ which further implies that $\psi_j^* = 0$ for $j = 1, \ldots, P$ in (18). Similarly, for sufficiently large $\omega$ in (19) , $\mathbf{G}_{\tau,3,j}(\mathbf{M}^*) = \mathbf{0}$ which further implies that $\psi_j^* = 0$ in (19). The block diagonal structure of $\sum_{b=1}^{B} \alpha_b^* [\mathbf{A}_x^b]$ implies that $\mathbf{M}^*$ and $\mathbf{N}^*$ can have at most one nonzero entry per row. For sufficiently large $v$ and DCA iterations $K$, if $\mathbf{M}_{\tau,\kappa} = \text{diag}(c_1, c_2, \ldots, c_p)\mathbf{M}^*$ for $\kappa \geq K$, with $\{c_i\}_{i=1}^{P}$ fixed arbitrary scalars that results

$$\begin{aligned} &\left[ \|\mathbf{M}_{\ell,:}^*\|_1 - \frac{\mathbf{M}_{\tau,\kappa,\ell,:}}{\|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2} (\mathbf{M}_{\ell,:}^*)^T \right] \\ &= \|\mathbf{M}_{\ell,:}^*\|_1 - \frac{c_\ell \cdot \mathbf{M}_{\ell,:}^* \cdot (\mathbf{M}_{\ell,:}^*)^T}{c_\ell \cdot \|\mathbf{M}_{\ell,:}^*\|_2} = \|\mathbf{M}_{\ell,:}^*\|_1 - \|\mathbf{M}_{\ell,:}^*\|_2 = 0, \end{aligned} \tag{64}$$

where $\ell = 1, \ldots, P$, the first equality stems from $\mathbf{M}_{\ell,:}^{\tau,\kappa} = c_\ell \cdot \mathbf{M}_{\ell,:}^*$ whereas the second inequality from the property that $\mathbf{M}^*$ has at most one nonzero entry per row. Thus, for sufficiently large $v$ the fourth summation term in the cost in (18) will be zero. Similarly, it can shown that the fourth summation term in (19) will also be zero. Further, $w_j^*$ in (18) will attain the maximum possible value coinciding with the $Q$th eigevalue of the local matrix $[\mathbf{H}_\tau^2(\mathbf{N}^*)]_{\mathcal{N}_j}$ formed using the submatrix $\mathbf{M}_{\mathcal{N}_j}^*$. Thus, $\mathbf{M}^*$ and $\{\alpha_b^*\}$ are minimizers of (18), and $\mathbf{N}^*$ minimizer of (19) for sufficiently large $v$ and $\omega$.

Similarly, for sufficiently large $\omega$ and $v$, $\mathbf{M}^*, \mathbf{N}^*$ and $\{\alpha_b^*\}_{b=1}^{B}$ result $\psi^* = 0$ following from the third LMI in (6) and second LMI in (8) being equal to a zero matrix, while

$$\begin{aligned} &\left[ \|\mathbf{M}_{\ell,:}^*\|_1 - \frac{\mathbf{M}_{\tau,\kappa,\ell,:}}{\|\mathbf{M}_{\tau,\kappa,\ell,:}\|_2} (\mathbf{M}_{\ell,:}^*)^T \right] = \\ &\left[ \|\mathbf{N}_{\ell,:}^*\|_1 - \frac{\mathbf{N}_{\tau,\kappa,\ell,:}}{\|\mathbf{N}_{\tau,\kappa,\ell,:}\|_2} (\mathbf{N}_{\ell,:}^*)^T \right] = 0, \end{aligned} \tag{65}$$

for $\ell = 1, \ldots, P$ and $\kappa \geq K$ when $\mathbf{M}_{\tau,\kappa} = \text{diag}(c_1, c_2, \ldots, c_P)\mathbf{M}^*$ and $\mathbf{N}_{\tau,\kappa} = \text{diag}(c_1', c_2', \ldots, c_P')$ $\cdot \mathbf{N}^*$, with $\{c_i, c_i'\}_{i=1}^{P}$ fixed arbitrary scalars. Again $w^*$ will attain the maximum possible value coinciding with the $Q$th eigevalue of $\mathbf{H}_{\tau,\kappa}^2(\mathbf{M}^*)$. Thus, $\mathbf{M}^*, \mathbf{N}^*$ and $\{\alpha_b^*\}_{b=1}^{B}$ are also minimizers of (6)-(8) for sufficiently large $v$ and $\omega$. Thus, both the central and separable formulations in (6) and (18) may not be equivalent but they share optimal solutions under the assumptions of Prop. 4 [similarly for (8) and (19)]. □

## Appendix E. Proof of Proposition 5

We will utilize the convergence claims in (He and Yuan, 2015). First we establish that the constraint set of (18) is strictly feasible (Slater's condition). Consider a set of nonnegative kernel coefficients values $\{\alpha_b^*\}_{b=1}^{B}$ such that $\alpha_b^j = \beta_b^{j,j'} = \alpha_b^*$ for $j' \in \mathcal{N}_j$ and $j = 1, \ldots, P$ that satisfy $\sum_{b=1}^{B} \alpha_b^* = 1$, as well as the equality constraints involving $\alpha_b^j$ and $\beta_b^{j,j'}$ in (18).

Let the eigenvalue decomposition $\sum_{b=1}^{B} \alpha_b^* \mathbf{A}_x^b = \mathbf{U}_{x,B} \mathbf{\Lambda}_B \mathbf{U}_{x,B}^T$ and consider factor matrix $\mathbf{M}^* = \mathbf{U}_{x,B}[:, 1:Q] \mathbf{\Lambda}_{B,Q}^{1/2}$, where $\mathbf{U}_{x,B}[:, 1:Q]$ corresponding to the first $Q$ principal eigenvectors in $\mathbf{U}_B$ with $\mathbf{\Lambda}_{B,Q}$ containing the corresponding principal eigenvalues. Note that $\sum_{b=1}^{B} \alpha_b^* \mathbf{A}_x^b - \mathbf{M}^* (\mathbf{M}^*)^T = \sum_{b=1}^{B} \alpha_b^* \mathbf{A}_x^b - \mathbf{U}_{x,B}[:, 1:Q] \mathbf{\Lambda}_{B,Q} \mathbf{U}_{x,B}[:, 1:Q]^T$ is a positive semidefinite matrix,

thus $\sum_{b=1}^{B} \alpha_b^* \mathbf{A}_x^b \succeq \mathbf{M}^*(\mathbf{M}^*)^T$ which further implies that $\mathbf{G}_{1,j} \succeq \mathbf{0}$ is satisfied for $j = 1, \ldots, P$ in (18). Further, setting $\mathbf{Z}_{j,j'} = \mathbf{M}_{j,:}^{j} = \mathbf{M}_{j,:}^{j'} = \mathbf{M}_{j,:}^{*}$ for $j = 1, \ldots, P$ and $j' \in \mathcal{N}_j$ ensures that the last two equalities in (18) are satisfied.

For the coefficient values $\{\alpha_b^*\}_{b=1}^{B}$ and factor $\mathbf{M}^*$ we can choose values $\psi_j^*, \theta_j^*$ that satisfy $\mathbf{G}_{\tau,2,j}(\mathbf{N}_\tau) \succeq \mathbf{0}$ and $\mathbf{G}_{\tau,3,j}(\mathbf{N}_\tau) \succeq \mathbf{0}$ for $j = 1, \ldots, P$. By setting $w_j^* = \{$Qth largest eigenvalue of $[\mathbf{M}_{\mathcal{N}_j}^*(\mathbf{M}_{\mathcal{N}_j}^*)^T\} \geq 0$ and sufficiently small $\theta_j^*$ the LMI $[\mathbf{H}_\tau^2(\mathbf{N}_\tau)]_{\mathcal{N}_j} \succeq w_j^* \cdot \mathbf{I}_Q$ for $j \in \mathcal{S}_Q$ is satisfied. Thus, the solution set of (18) is non-empty.

The $x$ variables in (He and Yuan, 2015, Eqn. (1.1)) correspond to factor vectors $\mathbf{M}_{j,:}^{j}, \{\mathbf{M}_{j,:}^{j'}\}_{j' \in \mathcal{N}_j}$, coefficients $\{\alpha_b^j\}_{b=1}^{B}$, variables $w_j, \psi_j$ and $\theta_j$ for $j = 1, \ldots, P$. The $y$ variables in (Eqn. (1.1) He and Yuan, 2015) include the auxiliary variates $\mathbf{Z}_{j,j'}$ and $\beta_b^{j,j'}$ for $j = 1, \ldots, p$ and $j' \in \mathcal{N}_j$. The equality constraint $A \cdot x + B \cdot y = 0$ in (Eqn. (1.1) He and Yuan, 2015) corresponds to the last four equality constraints in (18). Further, convex set $\mathcal{X}$ corresponds to the convex LMIs in (18) and scalar inequality constraints, as well as convex linear equality $\sum_b \alpha_b^j = 1$ for $j = 1, \ldots, P$. Set $\mathcal{Y}$ in (Eqn. (1.1) He and Yuan, 2015) is the entire vector space $\mathbb{R}^{Q \times 1}$ in which the $\mathbf{Z}_{j,j'}$ vectors lie which by definition is convex. Finally, $\theta_1(x)$ in (Eqn. (1.1) He and Yuan, 2015) corresponds to the convex cost function in (18), and $\theta_2(y) = 0$.

Thus, the convergence claims follow as a direct application of (Thm. 6.1 He and Yuan, 2015) since (18) was proved to be a special case of the family of problems considered in (Eqn. (1.1) He and Yuan, 2015). The exact same line of arguments can be applied for (19) to show that is a special case of the formulation in (Eqn. (1.1) He and Yuan, 2015). □

## References


Hyperspectral remote sensing scenes. Available: `http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes`, 2021.

Dimitri Bertsekas and John Tsitsiklis. *Parallel and distributed computation: numerical methods*. Athena Scientific, 2015.

Pantelis Bouboulis, Symeon Chouvardas, and Sergios Theodoridis. Online distributed learning over networks in rkh spaces using random fourier features. *IEEE Transactions on Signal Processing*, 66(7):1920–1932, 2017.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

Stephen P Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1548–1560, 2010.

Jiecao Chen, He Sun, David Woodruff, and Qin Zhang. Communication-optimal distributed clustering. *Advances in Neural Information Processing Systems*, 29, 2016.

Corinna Cortes, Mehryar Mohri, and Afshin Rostamizadeh. Algorithms for learning kernels based on centered alignment. *The Journal of Machine Learning Research*, 13(1):795–828, 2012.

Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz Kandola. On kernel-target alignment. *Advances in neural information processing systems*, 14, 2001.

Pouya M Ghari and Yanning Shen. Online multi-kernel learning with graph-structured feedback. In *International Conference on Machine Learning*, pages 3474–3483. PMLR, 2020.

Michael Grant and Stephen Boyd. Cvx: Matlab software for disciplined convex programming, version 2.1, 2014.

Bingsheng He and Xiaoming Yuan. On non-ergodic convergence rate of douglas–rachford alternating direction method of multipliers. *Numerische Mathematik*, 130(3):567–577, 2015.

Steven CH Hoi, Rong Jin, Peilin Zhao, and Tianbao Yang. Online multiple kernel classification. *Machine learning*, 90:289–316, 2013.

Songnam Hong and Jeongmin Chae. Distributed online learning with multiple kernels. *IEEE Transactions on neural networks and learning systems*, 2021.

Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

Kejun Huang, Nicholas D Sidiropoulos, and Ananthram Swami. Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition. *IEEE Transactions on Signal Processing*, 62(1):211–224, 2013.

Rong Jin, Steven CH Hoi, and Tianbao Yang. Online multiple kernel learning: Algorithms and mistake bounds. In *Algorithmic Learning Theory: 21st International Conference, ALT 2010, Canberra, Australia, October 6-8, 2010. Proceedings 21*, pages 390–404. Springer, 2010.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Shi Li and Xiangyu Guo. Distributed $k$-clustering for data with heavy noise. *Advances in Neural Information Processing Systems*, 31, 2018.

Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2): 129–137, 1982.

Akshay Malhotra and Ioannis D Schizas. On unsupervised simultaneous kernel learning and data clustering. *Pattern Recognition*, 108:107518, 2020.

Daniela Micucci, Marco Mobilio, and Paolo Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017.

Yuichi Motai. Kernel association for classification and prediction: A survey. *IEEE transactions on neural networks and learning systems*, 26(2):208–223, 2014.

Klaus-Robert Müller, Sebastian Mika, Koji Tsuda, and Koji Schölkopf. An introduction to kernel-based learning algorithms. In *Handbook of neural network signal processing*, pages 4–1. CRC Press, 2018.

Gabriele Oliva, Roberto Setola, and Christoforos N Hadjicostis. Distributed k-means algorithm. *arXiv preprint arXiv:1312.4176*, 2013.

Zhengxiao Peng, Yun Li, and Gang Hao. The research on distributed fusion estimation based on machine learning. *IEEE Access*, 8:38174–38184, 2020.

Jiahu Qin, Weiming Fu, Huijun Gao, and Wei Xing Zheng. Distributed $k$-means algorithm and fuzzy $c$-means algorithm for sensor networks based on multiagent consensus theory. *IEEE transactions on cybernetics*, 47(3):772–783, 2016.

Zhenwen Ren and Quansen Sun. Simultaneous global and local graph structure preserving for multiple kernel clustering. *IEEE transactions on neural networks and learning systems*, 32(5): 1839–1851, 2020.

Zhenwen Ren, Mithun Mukherjee, Mehdi Bennis, and Jaime Lloret. Multikernel clustering via non-negative matrix factorization tailored graph tensor over distributed networks. *IEEE Journal on Selected Areas in Communications*, 39(7):1946–1956, 2020.

Ban-Sok Shin, Masahiro Yukawa, Renato Luis Garrido Cavalcante, and Armin Dekorsy. Distributed adaptive learning with multiple kernels in diffusion networks. *IEEE Transactions on Signal Processing*, 66(21):5505–5519, 2018.

Pham Dinh Tao and LT Hoai An. Convex analysis approach to dc programming: theory, algorithms and applications. *Acta mathematica vietnamica*, 22(1):289–355, 1997.

George Trigeorgis, Konstantinos Bousmalis, Stefanos Zafeiriou, and Björn W Schuller. A deep matrix factorization method for learning attribute representations. *IEEE transactions on pattern analysis and machine intelligence*, 39(3):417–429, 2016.

Nikolaos Tsapanos, Anastasios Tefas, Nikolaos Nikolaidis, and Ioannis Pitas. A distributed framework for trimmed kernel k-means clustering. *Pattern recognition*, 48(8):2685–2698, 2015.

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475, 2001.

Lieven Vandenberghe, V Ragu Balakrishnan, Ragnar Wallin, Anders Hansson, and Tae Roh. Interior-point algorithms for semidefinite programming problems derived from the kyp lemma. *Positive polynomials in control*, pages 195–238, 2005.

Saeed V Vaseghi. *Advanced digital signal processing and noise reduction*. John Wiley & Sons, 2008.

Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *IEEE Transactions on knowledge and data engineering*, 25(6):1336–1353, 2012.

Penghang Yin, Yifei Lou, Qi He, and Jack Xin. Minimization of $\ell_{1-2}$ for compressed sensing. *SIAM Journal on Scientific Computing*, 37(1):A536–A563, 2015.