This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



# Contrastive Multi-view Kernel Learning

Jiyuan Liu, Xinwang Liu, *Senior Member, IEEE*, Yuexiang Yang, Qing Liao and Yuanqing Xia, *Senior Member, IEEE*

**Abstract**—Kernel method is a proven technique in multi-view learning. It implicitly defines a Hilbert space where samples can be linearly separated. Most kernel-based multi-view learning algorithms compute a kernel function aggregating and compressing the views into a single kernel. However, existing approaches compute the kernels *independently* for each view. This ignores complementary information across views and thus may result in a bad kernel choice. In contrast, we propose the *Contrastive Multi-view Kernel* — a novel kernel function based on the emerging contrastive learning framework. The Contrastive Multi-view Kernel implicitly embeds the views into a joint semantic space where all of them resemble each other while promoting to learn diverse views. We validate the method's effectiveness in a large empirical study. It is worth noting that the proposed kernel functions share the types and parameters with traditional ones, making them fully compatible with existing kernel theory and application. On this basis, we also propose a contrastive multi-view clustering framework and instantiate it with multiple kernel $k$-means, achieving a promising performance. To the best of our knowledge, this is the first attempt to explore kernel generation in multi-view setting and the first approach to use contrastive learning for a multi-view kernel learning.

**Index Terms**—Multi-view clustering, multiple kernel clustering, contrastive learning, kernel method, kernel function.

## 1 Introduction

Kernel technique is a fundamental paradigm in machine learning that has received considerable attention in real-world applications, such as image processing [1], [2], object detection [3], [4] and gene prediction [5]. To group the nonlinear-separable data, it defines an implicit kernel mapping which maps them into a high-dimensional Hilbert space where a clear decision boundary can be found [6]. Over the years, many kernel-based learning methods have been developed. The representatives are Kernel Support Vector Machines [6], Gaussian Processes [7] and Kernel $k$-means Clustering [8].

One obvious drawback of the methods mentioned above is that they can only handle data with a single kernel. However, in most practical settings, the data are collected from different sources/views. It would not make sense (nor would it be possible in most cases) to perform prediction without using all available information. For instance, lung patients are often diagnosed with a combination of nucleic acid test, blood test, and CT scan. In order to deal with these multi-view data problems, plenty of methods have been proposed [9], [10], with multiple kernel learning (MKL) being one of the most popular methodologies [11], [12]. MKL first computes one or several kernel matrices for each view and then aggregates the kernel matrices optimally for the learning task.

Current multiple kernel algorithms can be roughly grouped into three categories. Algorithms in the first category (known as *early-fusion* methods) directly learn a consensus kernel or graph for the subsequent clustering or classification process [11], [13], [14], [15], [16]. Frequently, both steps are unified into a single objective formulation, which can be solved using alternating optimization. For instance, Huang and Kloft et al. assume that the consensus kernel can be parameterized into a weighted linear combination of the pre-specified ones [13], [14], [17]. On this basis, Liu et al. propose a matrix-induced regularization to dynamically adjust the weights along with optimization, achieving satisfactory performance improvement [15]. Then, Liu et al. claim that the optimal kernel can be found in the proximity of the weighted kernel combination [16]. Meanwhile, some researchers propose to push the consensus kernel close to each pre-specified kernel [18], [19]. Since kernel matrix stores the pairwise similarities of the samples, it makes sense to transform the kernel matrix into a graph, in which a graph algorithm can be employed subsequently [20], [21]. Upon this assumption, Ren et al. compute candidate affinity graphs from pre-specified kernels and learn the consensus kernel and graph coherently [22]. Another category of MKL methods (called *late-fusion*) first imputes multiple base partitions from each kernel (e.g., using kernel $k$-means) and then integrates the partitions into a unified one [23]. For instance, Wang et al. maximize the alignment between the consensus partition and the weighted combination of base partitions [24]. In addition, we group the rest into the third category, in which a hierarchical method also achieves promising performance [25].

All the above methods concentrate on how to fuse pre-specified kernels, but ignore that the kernel quality is a performance bottleneck. In contrast, instead of using traditional kernel functions, we propose the Contrastive Multi-view Kernel (CMK), a novel unsupervised kernel generation paradigm to compute quality kernels by leveraging complementary information from the data views. It is inspired by the paradigm of contrastive learning, and the key idea is

- *J. Liu, X. Liu and Y. Yang are with the School of Computer, National University of Defense Technology, Changsha, Hunan, China, 410072. E-mail: {liujiyuan13, xinwangliu, yyx}@nudt.edu.cn*
- *Qing Liao is with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China, 518055. E-mail: liaoqing@hit.edu.cn.*
- *Yuanqing Xia is with School of Automation, Beijing Institute of Technology, China, 100081. E-mail: xia_yuanqing@bit.edu.cn.*
- *Corresponding author: Xinwang Liu.*

*Manuscript received May 10, 2022.*

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



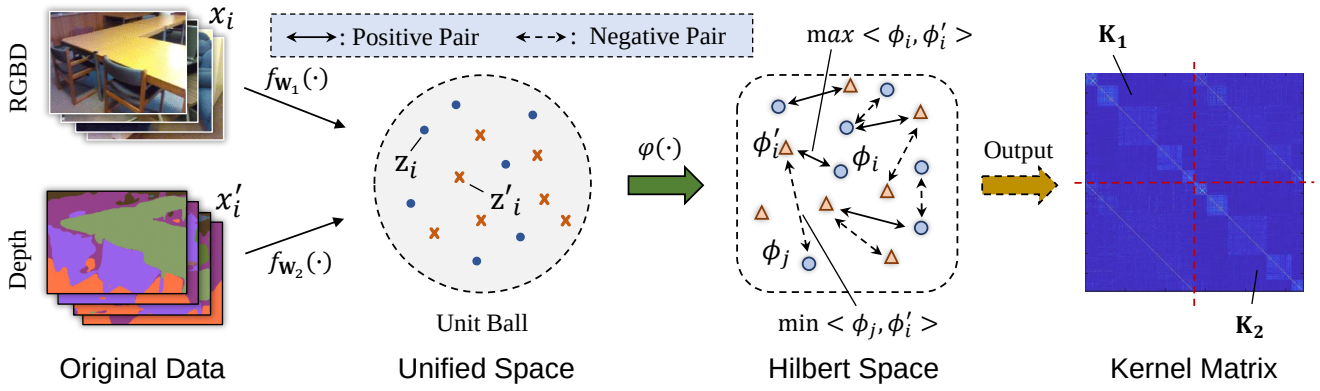



Fig. 1. Generation paradigm of the Contrastive Multi-view Kernel on images. For ease of expression, we consider only RGB and Depth images as the two data views. At the very beginning, the data $\mathbf{x}_i$ and $\mathbf{x}'_i$ are encoded into a unified space with two mapping functions $f_{\mathbf{W}_1}(\cdot)$ and $f_{\mathbf{W}_2}(\cdot)$. Then, they are projected into a Hilbert space with an implicit kernel mapping $\varphi(\cdot)$. Here, two representations of each data sample are considered as 'positive pairs' (for which we want a high kernel similarity $< \phi_i, \phi'_i >$) while disjointed data samples are considered as 'negative pairs' (for which we want low kernel similarity $< \phi_j, \phi'_i >$, thus promoting the diversity of the learned mappings). Note that the correlations between the samples are partially plotted for simplicity of the picture. Finally, the kernel matrices of each view can be obtained as $\mathbf{K}_1$ and $\mathbf{K}_2$.

to promote a high similarity across views for a given data sample while learning diverse and heterogeneous views. The approach is illustrated in Fig. 1. First, we separately encode the multi-view data into a unified (semantic) space using their respective learned mapping functions. Second, the obtained data representations are further projected into an implicit Hilbert space. Here, the representations of any two views of one (and the same) data sample are considered as positive pairs, so their kernel similarities are maximized. Meanwhile, the representations associated with two data samples are treated as negative pairs, so their kernel similarities are minimized. With updating the mapping functions, the proposed contrastive multi-view kernel function and corresponding kernel matrix of each data view can be obtained finally.

In a large experimental study, we compare the CMK with multiple types of traditional kernels, observing a promising performance improvement. Note that the proposed kernel functions share the types and parameters with traditional ones, making them fully compatible with existing kernel theory and applications. In other words, once the associated variables are optimized, the CMK is able to be applied in existing kernel methods, such as kernel SVM and kernel $k$-means, without any extra cost. Nevertheless, we find it can largely improve the performance of Multiple Kernel Clustering (MKC) algorithms to jointly optimize the CMK loss and theirs. On this basis, we propose a Contrastive Multi-view Clustering framework and instantiate it with the widely used Multiple Kernel $k$-means (MKKM), surpassing state-of-the-art methods in experiment. To the best of our knowledge, this is the first attempt at leveraging contrastive learning for multi-view kernel learning and of exploring kernel generation in a multi-view setting. Our work opens the door to new avenues in future research on using contrastive learning in multi-view and kernel learning.

The rest paper is organized as follows. Section 2 introduces two parts of closely related researches, including traditional kernel generation and contrastive learning. Section 3 presents the proposed CMK generation paradigm, its implementation, instance, complexity analysis and large-scale solution. In Section 4, we propose the Contrastive Multi-view Clustering framework and instantiate it with Multiple Kernel $k$-means. Experiment details, such as parameter setting, performance comparison and insights of model building, are introduced and analyzed in Section 5. At last, we make the conclusion in Section 7.

## 2 Related Work

Since the proposed CMK leverages contrastive learning on kernel generation, we briefly review the closely related researches of the two domains.

### 2.1 Kernel generation

For a set of data samples $\{\mathbf{x}_i\}_{i=1}^N$ drawn from a space $\mathcal{X} \subseteq \mathbb{R}^{d_\mathcal{X}}$, a kernel method encodes them into a Reproducing Kernel Hilbert Space $\mathcal{H} \subseteq \mathbb{R}^{d_\mathcal{H}}$ with an implicit kernel mapping $\varphi(\cdot)$. Since the dimension of Hilbert space $\mathcal{H}$ could be infinite, the mapping function $\varphi(\cdot)$ is hard to define explicitly, making it impossible to compute corresponding embeddings. Thanks to Mercer's theorem [26], we can measure the product of vectors in space $\mathcal{H}$ with the kernel function $k(\cdot,\cdot)$ in space $\mathcal{X}$ as

$$\mathbf{K}\,[i,j] = \varphi(\mathbf{x}_i)^\top \varphi(\mathbf{x}_j) = k(\mathbf{x}_i, \mathbf{x}_j), \tag{1}$$

where $[i,j]$ refers to the value in $i$-th row and $j$-th column of target kernel matrix $\mathbf{K}$. As a supplement, the widely used kernel functions are partially listed in Table 1.

TABLE 1
Representatives of traditional kernel function.

| Kernel Type | Formulation |
|---|---|
| Gaussian | $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)$ |
| Linear | $a\mathbf{x}_i^\top \mathbf{x}_j + c$ |
| Polynomial | $(a\mathbf{x}_i^\top \mathbf{x}_j + c)^d$ |
| Sigmoid | $\tanh(a\mathbf{x}_i^\top \mathbf{x}_j + c)$ |
| Cauchy | $(\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma + 1)^{-1}$ |

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



To deal with multi-view data, current multiple kernel methods generate one or more kernels on each data view. They usually focus on improving performance via exploring more effective way to fuse discriminative information from these kernels [11], [13], [14], [15], [16], but overlook the fact that kernel quality is a bottleneck. There are also some researches about how to choose the parameters in kernel functions, such as [27]. However, they are out of our scope, since the proposed CMK provides a new kernel generation paradigm and shares the same types and parameters with traditional kernel functions.

### 2.2 Contrastive learning

Contrastive learning is first proposed in [28] to address deep visual representation learning problem. By substantially promoting the representation capability of neural networks, it attracts lots of interest from industry and the research community [29], [30], [31], [32]. The idea beneath contrastive learning is to learn discriminative embeddings via maximizing the similarities between two random data augmentations.

For data $\{\mathbf{x}_i\}_{i=1}^N$, two separate data augmentation operators are randomly selected from an augmentation family $\mathcal{T}$. As a result, $2N$ augmented samples are derived. Then, a base encoder network $f(\cdot)$ is employed to map them into latent representations $\{\mathbf{h}_i\}_{i=1}^{2N}$. Subsequently, a projection head $g(\cdot)$, which only consists of multiple linear layers, is adopted to obtain $\{\mathbf{z}_i\}_{i=1}^{2N}$. Denoting $\mathbf{x}_{j(i)}$ as the augmentation of the $i$-th data sample, contrastive learning treats them as positive pair but the rest as negative pairs. By maximizing the similarities between positive pairs and minimizing those between negative pairs, it defines the Normalized Temperature-scaled Cross Entropy Loss (NT-Xent) as follows:

$$\ell_{i,j(i)} = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_{j(i)})/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k\neq i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}, \tag{2}$$

where $\mathbb{1}_{k\neq i} \in \{0, 1\}$ is the indicator function, $\tau$ denotes a temperature parameter and

$$\text{sim}(\mathbf{z}_i, \mathbf{z}_j) = \frac{\mathbf{z}_i^\top \mathbf{z}_j}{\|\mathbf{z}_i\|\|\mathbf{z}_j\|}. \tag{3}$$

Apart from NT-Xent, other types of loss function are tested but achieve worse performances. In addition, Yeh et al. propose to remove the positive pair in the denominator of Eq. (2) and achieve better results [33]. By introducing supervisory signals, Khosla et al. refine the loss in Eq. (2) by labeling samples of the same class and their augmentations as positive pairs [32].

## 3 Contrastive Multi-View Kernel

We leverage the contrastive learning paradigm to multi-view kernel learning. In the beginning, the CMK generation paradigm and its implementation are described. Then, we introduce five common instances of our general paradigm. Finally, CMK's complexity and large-scale solution are analyzed in detail.

### 3.1 Generation paradigm

Given a set of multi-view data $\{\mathbf{x}_i^v\}_{i,v=1}^{N,V}$ where $\mathbf{x}_i^v \in \mathbb{R}^{d_v}$, we first eliminate the dimension differences by encoding them into a unified latent space $\mathcal{X}_h \subseteq \mathbb{R}^d$ with mapping functions $f_{\mathbf{W}_v}(\cdot)$ via

$$\mathbf{h}_i^v = f_{\mathbf{W}_v}(\mathbf{x}_i^v) = \mathbf{x}_i^v \mathbf{W}_v, \tag{4}$$

in which $\mathbf{W}_v \in \mathbb{R}^{d_v \times d}$. Denoting $L_2(\cdot)$ as the $L_2$-norm of a vector, the normalized data representations can be obtained via

$$\mathbf{z}_i^v = f_N(\mathbf{h}_i^v) = \mathbf{h}_i^v \ / \ L_2(\mathbf{h}_i^v). \tag{5}$$

With $\varphi_z(\cdot)$ being an implicit but known kernel mapping, such as Gaussian mapping, we project the representations $\{\mathbf{z}_i^v\}_{i,v=1}^{N,V}$ into corresponding Hilbert space $\mathcal{H}$. As a consequence, the overall kernel mapping of the $v$-th view is obtained as

$$\varphi_c^v(\mathbf{x}_i^v) = \varphi_z(\mathbf{z}_i^v) = \varphi_z(f_N(f_{\mathbf{W}_v}(\mathbf{x}_i^v))), \tag{6}$$

in which the resulting kernel function is

$$\begin{aligned} k_c^v(\mathbf{x}_i^v, \mathbf{x}_j^v) &= k_z(\mathbf{z}_i^v, \mathbf{z}_j^v) \\ &= k_z(f_N(f_{\mathbf{W}_v}(\mathbf{x}_i^v)), f_N(f_{\mathbf{W}_v}(\mathbf{x}_j^v))), \end{aligned} \tag{7}$$

where $k_z(\cdot, \cdot)$ refers to the kernel function defined by kernel mapping $\varphi_z(\cdot)$, shown as

$$k_z(\mathbf{u}, \mathbf{v}) = \varphi_z(\mathbf{u})^\top \varphi_z(\mathbf{v}). \tag{8}$$

Also, it is obvious that the kernel matrix of $v$-th view should be

$$\mathbf{K}_c^v[i, j] = k_c^v(\mathbf{x}_i^v, \mathbf{x}_j^v). \tag{9}$$

As an arbitrary data view can be regarded as augmentation of the others semantically, we can naturally leverage contrastive learning loss on multi-view theory [34], [35]. Similar to the unsupervised setting in [28], $\{\mathbf{x}_i^v\}_{v=1}^V$ are regarded as a positive pair, leaving the remaining pairs as negative pairs. Thus, the loss of the $i$-th data sample in the $v$-th view can be written as

$$\ell_{i,v} = \frac{1}{V-1} \sum_{v'=1, v'\neq v}^{V} -\log \frac{\exp(k_z(\mathbf{z}_i^v, \mathbf{z}_i^{v'}))}{\sum_{j,v''\in\mathcal{A}_{i,v}} \exp(k_z(\mathbf{z}_i^v, \mathbf{z}_j^{v''}))}, \tag{10}$$

where

$$\mathcal{A}_{i,v} = \{1, 2, \cdots, N\} \times \{1, 2, \cdots, V\} \setminus \{(i, v)\}. \tag{11}$$

It can be observed that we measure the similarity of sample pairs with kernel function and directly maximize these of positive pairs while minimize the rest. The overall loss is implemented as

$$\ell_c = \frac{1}{NV} \sum_{i,v=1}^{N,V} \ell_{i,v}. \tag{12}$$

By minimizing the loss in Eq. (12), only variables $\{\mathbf{W}_v\}_{i=1}^V$ will be optimized, determining a unique kernel mapping $\varphi_v(\cdot)$ in Eq. (6) and kernel matrix $\mathbf{K}^v$ in Eq. (9) for the $v$-th view. Besides, we visualize the generation paradigm of CMK in Fig. 1.

It is worth to note that the CMK refers to a unified kernel paradigm and differs from each other by adopting a different kernel mapping $\varphi_z(\cdot)$ in Eq. (6) and kernel

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



**Algorithm 1** The Contrastive Multi-view Kernel Generation Paradigm

**Input**: Data $\{\mathbf{x}_i^v\}_{i,v=1}^{N,V}$
**Output**: Kernel mapping $\varphi_c^v(\cdot)$ and kernel function $k_c^v(\cdot,\cdot)$

```
Initialize the mapping weights {W_v}_{v=1}^V randomly;
t = 0;
while t < epochs do
    # forward
    Compute the loss value ℓ_c in Eq. (12);
    # back propagation
    Compute the loss gradients ∂ℓ_c/∂W_v via Eq. (13);
    Update {W_v}_{v=1}^V via Eq. (23);
    t = t + 1;
end while
Obtain the updated weights {W_v}_{v=1}^V;
```

function $k_z(\cdot,\cdot)$ in Eq. (7). For example, a Gaussian CMK can be obtained when instantiating $\varphi_z(\cdot)$ and $k_z(\cdot,\cdot)$ with Gaussian kernel. More instantiation details are thoroughly described in Section 3.4. Due to this pairwise correlation, the CMK is proposed to compete with traditional kernels correspondingly, such as Gaussian CMK v.s. Gaussian kernel. Therefore, it can be utilized into a large set of kernel methods to improve their performance, enjoying a promising application prospect.

### 3.2 Implementing the critic

In order to optimize the proposed model, we adopt the Gradient Descent (GD) algorithm and compute the gradients on variables $\{\mathbf{W}_v\}_{v=1}^V$ with chain rule as

$$\frac{\partial \ell_c}{\partial \mathbf{W}_v} = \sum_{i=1}^{N}\left(\frac{\partial \ell_c}{\partial \mathbf{z}_i^v}\cdot\frac{\partial \mathbf{z}_i^v}{\partial \mathbf{h}_i^v}\cdot\frac{\partial \mathbf{h}_i^v}{\partial \mathbf{W}_v}\right). \tag{13}$$

By utilizing Eq. (12), the gradient of $\ell_c$ with respect to $\mathbf{z}_i^v$ can be decomposed into

$$\frac{\partial \ell_c}{\partial \mathbf{z}_i^v} = \frac{1}{NV}\sum_{i',v'=1}^{N,V}\frac{\partial \ell_{i',v'}}{\partial \mathbf{z}_i^v}. \tag{14}$$

For any target $i_0$ and $v_0$, we separate the sub-losses of Eq. (12) into three groups, including $\ell_{i_0,v_0}$, $\{\ell_{i_0,v}\}_{v=1,v\neq v_0}^V$ and $\{\ell_{i,v}\}_{i=1,v=1,i\neq i_0}^{N,V}$. Correspondingly, Eq. (14) can be rewritten as

$$\frac{\partial \ell_c}{\partial \mathbf{z}_{i_0}^{v_0}} = \frac{1}{NV}\left(\frac{\partial \ell_{i_0,v_0}}{\partial \mathbf{z}_{i_0}^{v_0}} + \sum_{v=1,v\neq v_0}^{V}\frac{\partial \ell_{i_0,v}}{\partial \mathbf{z}_{i_0}^{v_0}} + \sum_{i=1,v=1,i\neq i_0}^{N,V}\frac{\partial \ell_{i,v}}{\partial \mathbf{z}_{i_0}^{v_0}}\right). \tag{15}$$

Denoting

$$\mathcal{B}_{i,v} = \sum_{i',v'\in\mathcal{A}_{i,v}}\exp(k_z(\mathbf{z}_i^v,\mathbf{z}_{i'}^{v'})), \tag{16}$$

Each item of Eq. (15) can be computed as follows:

1) For $\partial \ell_{i_0,v_0}/\partial \mathbf{z}_{i_0}^{v_0}$, it holds that

$$\begin{aligned}\frac{\partial \ell_{i_0,v_0}}{\partial \mathbf{z}_{i_0}^{v_0}} &= -\frac{1}{V-1}\sum_{v=1,v\neq v_0}^{V}\frac{\partial k_z(\mathbf{z}_{i_0}^{v_0},\mathbf{z}_{i_0}^{v})}{\partial \mathbf{z}_{i_0}^{v_0}} \\ &+ \sum_{i,v\in\mathcal{A}_{i_0,v_0}}\frac{\exp(k_z(\mathbf{z}_{i_0}^{v_0},\mathbf{z}_i^v))}{\mathcal{B}_{i_0,v_0}}\cdot\frac{\partial k_z(\mathbf{z}_{i_0}^{v_0},\mathbf{z}_i^v)}{\partial \mathbf{z}_i^v}\end{aligned} \tag{17}$$

2) For $\partial \ell_{i_0,v}/\partial \mathbf{z}_{i_0}^{v_0}$, we can get

$$\begin{aligned}\frac{\partial \ell_{i_0,v}}{\partial \mathbf{z}_{i_0}^{v_0}} &= -\frac{1}{V-1}\cdot\frac{\partial k_z(\mathbf{z}_{i_0}^{v},\mathbf{z}_{i_0}^{v_0})}{\partial \mathbf{z}_{i_0}^{v_0}} \\ &+ \frac{\exp(k_z(\mathbf{z}_{i_0}^{v},\mathbf{z}_{i_0}^{v_0}))}{\mathcal{B}_{i_0,v}}\cdot\frac{\partial k_z(\mathbf{z}_{i_0}^{v},\mathbf{z}_{i_0}^{v_0})}{\partial \mathbf{z}_{i_0}^{v_0}}\end{aligned} \tag{18}$$

3) For $\partial \ell_{i,v}/\partial \mathbf{z}_{i_0}^{v_0}$, it is obvious that

$$\frac{\partial \ell_{i,v}}{\partial \mathbf{z}_{i_0}^{v_0}} = \frac{\exp(k_z(\mathbf{z}_i^{v},\mathbf{z}_{i_0}^{v_0}))}{\mathcal{B}_{i,v}}\cdot\frac{\partial k_z(\mathbf{z}_i^{v},\mathbf{z}_{i_0}^{v_0})}{\partial \mathbf{z}_{i_0}^{v_0}} \tag{19}$$

Furthermore, denoting $z_{j'}$ and $h_{i'}$ as the $j'$-th and $i'$-th element of $\mathbf{z}_i^v$ and $\mathbf{h}_i^v$, we can obtain

$$\frac{\partial \mathbf{z}_i^v}{\partial \mathbf{h}_i^v} = \left[\sum_{j'=1}^{d}\frac{\partial z_{j'}}{\partial h_1},\cdots\sum_{j'=1}^{d}\frac{\partial z_{j'}}{\partial h_{i'}},\cdots,\sum_{j'=1}^{d}\frac{\partial z_{j'}}{\partial h_d}\right], \tag{20}$$

where

$$\frac{\partial z_{j'}}{\partial h_{i'}} = \mathbb{1}_{i'=j'}\left(\sum_{k=1}^{d}h_k^2\right)^{-1/2} + h_{i'}h_{j'}\left(\sum_{k=1}^{d}h_k^2\right)^{-3/2} \tag{21}$$

Additionally, the gradient of $\mathbf{h}_i^v$ on $\mathbf{W}_v$ can be computed as

$$\frac{\partial \mathbf{h}_i^v}{\partial \mathbf{W}_v} = \mathbf{x}_i^{v\top} \tag{22}$$

By setting the learning rate to $\alpha$, the updating of $\mathbf{W}_v$ is written as

$$\mathbf{W}_v = \mathbf{W}_v - \alpha\frac{\partial \ell_c}{\partial \mathbf{W}_v}. \tag{23}$$

In summary, we present an overview of the CMK generation paradigm in Algorithm 1.

### 3.3 Complexity and large-scale solution

In this section, we analyze the computation complexity of the proposed CMK. Since the weights $\{\mathbf{W}_v\}_{v=1}^V$ are optimized with the GD algorithm, the complexity is only dependent on the gradient computation. To compute the gradient $\partial \ell_c/\partial \mathbf{W}_v$, one should compute Eq. (13), (15), (20) and (22). Note that, the computation complexity of Eq. (20) and (22) are only related to the dimension $d$ of latent representations, and therefore are ignored here. At the very beginning, we pre-compute and store $\{\mathcal{C}_{i,v}\}_{i,v=1}^{N,V}$ with each being

$$\mathcal{C}_{i,v} = \sum_{i',v'=1}^{N,V}\exp(k_z(\mathbf{z}_i^v,\mathbf{z}_{i'}^{v'})). \tag{24}$$

Corresponding complexity is $\mathcal{O}(V^2N^2)$. It is obvious that

$$\mathcal{B}_{i,v} = \mathcal{C}_{i,v} - \exp(k_z(\mathbf{z}_i^v,\mathbf{z}_i^v)), \tag{25}$$

which prevents from duplicated computation of $\mathcal{B}_{i,v}$ in Eq. (17), (18) and (19). In this way, the computation of Eq. (15) by utilizing Eq. (17), (18) and (19) only requires $\mathcal{O}(VN)$ complexity. Since $\partial \ell_c/\partial \mathbf{W}_v$ of Eq. (13) is the sum of $N$ items with each consisting of Eq. (15), its complexity is $\mathcal{O}(VN^2)$. To optimize the whole model, one needs to compute $\{\partial \ell_c/\partial \mathbf{W}_v\}_{v=1}^V$, resulting in the $\mathcal{O}(V^2N^2)$ complexity. Considering the aforementioned pre-computation, the overall complexity is $\mathcal{O}(2V^2N^2)$ which can be rewritten as $\mathcal{O}(V^2N^2)$.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



The aforementioned quadratic complexities prevent CMK from handling with large-scale data. A direct and effective solution is to employ the Stochastic Gradient Descent (SGD) strategy in the optimization where data are split into batches. In specific, given a random batch of multi-view data $\{\mathbf{x}_i^v\}_{i,v=1}^{N_b,V}$, corresponding loss can be accumulated in Eq. (10) and (12) in which

$$\mathcal{A}_{i,v} = \{1, 2, \cdots, N_b\} \times \{1, 2, \cdots, V\} \setminus \{(i, v)\}. \quad (26)$$

In this way, the computation complexity for each data batch is $\mathcal{O}(V^2 N_b^2)$. With $t$ denoting the number of epochs, $tN/N_b$ data batches will be used in the model training. Therefore, the overall complexity is $\mathcal{O}(tV^2 N_b N) = \mathcal{O}(V^2 N_b^2 \cdot tN/N_b)$. Since $N_b << N$ in most neural network researches, we can train the CMK model within a linear time.

Moreover, two techniques will also help mitigate the large-scale problem. First, several deep learning packages (including PyTorch and TensorFlow) can accelerate the Gradient Descent algorithm using GPU computations. Second, we can separate data into two splits and use one part to train CMK's parameters and the other part or all of them to compute the kernel matrices.

### 3.4 Instantiation

It is obvious from Eq. (6) and (7) that the proposed CMK mapping function $\varphi_c^v(\cdot)$ and kernel function $k_c^v(\cdot,\cdot)$ are defined on the given $\varphi_z(\cdot)$ and $k_z(\cdot,\cdot)$ which can be instantiated with the widely-used traditional kernels. Here, five common ones are concerned, including Gaussian, Linear, Polynomial, Sigmoid and Cauchy. Due to the implicit property of mapping function, only the kernel function definitions are presented in Table 1. Taking the Gaussian kernel as an example, the instantiated CMK kernel function of $v$-th view is

$$k_c^v(\mathbf{x}_i^v, \mathbf{x}_j^v) = \exp\left(\frac{-\|f_N(f_{\mathbf{W}_v}(\mathbf{x}_i^v)) - f_N(f_{\mathbf{W}_v}(\mathbf{x}_j^v))\|^2}{2\sigma^2}\right). \quad (27)$$

For the computation of Eq. (17), (18) and (19), we also list gradients of the five kernel types in the following.

1) Gaussian:

$$\frac{\partial k_z(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma^2)}{\sigma^2} \cdot (\mathbf{x}_j - \mathbf{x}_i) \quad (28)$$

2) Linear:

$$\frac{\partial k_z(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} = \mathbf{x}_j \quad (29)$$

3) Polynomial:

$$\frac{\partial k_z(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} = a(a\mathbf{x}_i^\top \mathbf{x}_j + c)^{d-1} \cdot \mathbf{x}_j \quad (30)$$

4) Sigmoid:

$$\frac{\partial k_z(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} = a(1 - \tanh^2(a\mathbf{x}_i^\top \mathbf{x}_j + c)) \cdot \mathbf{x}_j \quad (31)$$

5) Cauchy:

$$\frac{\partial k_z(\mathbf{x}_i, \mathbf{x}_j)}{\partial \mathbf{x}_i} = \frac{2}{\sigma(\|\mathbf{x}_i - \mathbf{x}_j\|^2/\sigma + 1)^2} \cdot (\mathbf{x}_i - \mathbf{x}_j) \quad (32)$$

## 4 Contrastive Multi-view Clustering

Apart from the kernel generation paradigm, we propose to unify the CMK generation into downstream kernel tasks for the sake of improving their performance. Here, one considers the Multiple Kernel Clustering (MKC) setting. At the beginning, the proposed framework is introduced. Then, we instantiate it with the widely used Multiple Kernel $k$-means. Finally, an alternate strategy is designed to optimize the resultant problem.

### 4.1 Framework

Existing MKC methods assumes the kernel matrices are computed in advance and fixed during the clustering process. Denote $m$ ready-made kernels $\{\mathbf{K}_p\}_{p=1}^m$, they prefer to minimize a loss like

$$\ell_{\mathbf{K}} = g_{\mathbf{\Theta}}(\{\mathbf{K}_p\}_{p=1}^m,\ \mathbf{F}),\ \ s.t.\ g_{\mathbf{\Theta}} \in \mathcal{G}, \quad (33)$$

where $\mathcal{G}$ is a class of objective functions and $\mathbf{\Theta}$ represents the extra temporary variables. Meanwhile, $\mathbf{F}$ is the target hard label ($\mathbb{R}^{N\times 1}$) [36] or soft label ($\mathbb{R}^{N\times k}$ with $k$ being the number of class) [15], [19] and obtained via optimization. Here, we propose to perform kernel clustering along with CMK generation by defining the overall loss as

$$\ell \ = \ \ell_c + \lambda \ell_{\mathbf{K}}. \quad (34)$$

In Eq. (34), the two processes contribute to each other, i.e. the CMK paradigm generates kernel matrices for the latter MKC model to achieve a better performance; as a feedback, a better MKC model motivates the generation of more specific CMK matrices. In the experiments, we will show this unified learning mode outperforms the separated one.

### 4.2 Instantiation

Without loss of generality, we instantiate the aforementioned contrastive multi-view clustering framework with Multiple Kernel $k$-means (MKKM) [37], whose objective function $g_{\mathbf{\Theta}}$ should be

$$\sum_{p=1}^{m} \beta_p \mathrm{Tr}\left[\mathbf{K}_p(\mathbf{I}_N - \mathbf{F}\mathbf{F}^\top)\right],\ \ s.t.\ \mathbf{F}^\top\mathbf{F} = \mathbf{I}_k, \quad (35)$$

in which $\beta_p$ is the weight of $p$-th kernel, $\mathbf{I}_k$ refers to the identity matrix of size $k$ and $\mathbf{F} \in \mathbb{R}^{N\times k}$ is the target soft label. Therefore, we can obtain the model, named Contrastive Multiple Kernel $k$-means (CMKKM) for brevity, as

$$\begin{aligned} \ell \ &= \ \frac{1}{NV}\sum_{i,v=1}^{N,V} \ell_{i,v} + \lambda \frac{1}{N}\sum_{v=1}^{V} \beta_v \mathrm{Tr}\left[\mathbf{K}_c^v(\mathbf{I}_N - \mathbf{F}\mathbf{F}^\top)\right] \\ &s.t.\ \mathbf{F}^\top\mathbf{F} = \mathbf{I}_k, \end{aligned} \quad (36)$$

where $\ell_{i,v}$ represents the loss of Gaussian CMK[1]. Here, we consider the kernel function learning and Multiple Kernel $k$-means task as two equally important parts and set $\lambda = 1$. Also, $\beta_v$ is globally set to $1/V$ in order to balance the kernel information from each data view. Note that, a large set of MKC algorithms, apart from Multiple Kernel $k$-means, can be unified in the proposed framework, showing its generality.

1. In the following, we use Gaussian CMK in CMKKM by default.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



**Algorithm 2** Contrastive Multiple Kernel $k$-means

**Input**: Data $\{\mathbf{x}_i^v\}_{i,v=1}^{N,V}$
**Output**: Data cluster assignment $\mathbf{Y}$

Obtain the updated weights $\{\mathbf{W}_v\}_{v=1}^V$ via Algorithm 1;
$t = 0$;
**while** $t < epochs$ **do**
  # *kernel clustering*
  Compute the kernel matrix $\{\mathbf{K}_c^v\}_{v=1}^V$ via Eq. (9);
  Update the soft label $\mathbf{F}$ via Eq. (40);
  # *kernel function learning*
  Compute the loss value $\ell$ in Eq. (36);
  Compute the loss gradients $\partial\ell/\partial\mathbf{W}_v$ via Eq. (37);
  Update $\{\mathbf{W}_v\}_{v=1}^V$ via Eq. (38);
  $t = t + 1$;
**end while**
Obtain the soft label $\mathbf{F}$;
Obtain $\mathbf{Y}$ by performing $k$-means on $\mathbf{F}$;

### 4.3 Optimization

In the optimization problem of Eq. (36), there are two independent sets of variables, i.e. the weights $\{\mathbf{W}_v\}_{v=1}^V$ in kernel functions $\{k_c^v(\cdot,\cdot)\}_{v=1}^V$ and the target soft label $\mathbf{F}$. To solve them, we design an alternate strategy in which one variable is computed while the others are fixed.

For $\{\mathbf{W}_v\}_{v=1}^V$ with fixed $\mathbf{F}$, Gradient Descent (GD) algorithm is adopted, where their gradients are computed with chain rule as

$$\frac{\partial \ell}{\partial \mathbf{W}_v} = \sum_{i=1}^{N} \frac{\partial \ell}{\partial \mathbf{z}_i^v} \cdot \frac{\partial \mathbf{z}_i^v}{\partial \mathbf{h}_i^v} \cdot \frac{\partial \mathbf{h}_i^v}{\partial \mathbf{W}_v}. \tag{37}$$

Due to the space limit, we omit the detailed derivation here. With setting the learning rate to $\alpha$, the updating is shown as

$$\mathbf{W}_v = \mathbf{W}_v - \alpha \frac{\partial \ell}{\partial \mathbf{W}_v}. \tag{38}$$

Once fixing $\{\mathbf{W}_v\}_{v=1}^V$, the CMK matrices $\{\mathbf{K}_c^v\}_{v=1}^V$ are available and the problem can be transformed to

$$\max_{\mathbf{F}} \operatorname{Tr}\left[\sum_{v=1}^{V} \mathbf{K}_c^v \mathbf{F}\mathbf{F}^\top\right], \quad s.t.\ \mathbf{F}^\top\mathbf{F} = \mathbf{I}_k. \tag{39}$$

Suppose $\mathbf{u}_i$ and $\sigma_i$ be the $i$-th pairwise eigen-vector and eigen-value of matrix $\sum_{v=1}^V \mathbf{K}_c^v$, the solution of $\mathbf{F}$, by following [25], should be the horizontal concatenation of $k$ eigen-vectors as

$$\begin{aligned} \mathbf{F}^* &= [\mathbf{u}_{i'_1}; \mathbf{u}_{i'_2}; \cdots; \mathbf{u}_{i'_k}] \\ &s.t.\ \{i'_t\}_{t=1}^k \subset \{1, 2, \cdots, N\}, \end{aligned} \tag{40}$$

where the corresponding $\{\sigma_{i'_t}\}_{t=1}^k$ are the $k$ largest out of $N$ eigen-values. Moreover, we present the overall optimization strategy in Algorithm 2.

## 5 Experiment

In the following, we first introduce the used datasets and then design experiments to validate effectiveness of the proposed CMK generation paradigm and CMKKM algorithm.

TABLE 2
Details of the used datasets.

| Dataset | Type | Number of Samples | Views | Clusters |
|---|---|---|---|---|
| BBC | multi-feature | 2012 | 2 | 5 |
| BBCSport | multi-feature | 554 | 2 | 5 |
| CiteSeer | multi-modal | 3312 | 4 | 6 |
| Cora | multi-modal | 2708 | 4 | 7 |
| Movies | multi-modal | 617 | 2 | 17 |
| AwA | multi-feature | 30475 | 2 | 50 |
| CCV | multi-modal | 6773 | 3 | 20 |
| NusWide | multi-feature | 23953 | 5 | 31 |
| YtVideo | multi-modal | 101499 | 3 | 31 |
| CropLand | multi-modal | 325834 | 2 | 7 |

### 5.1 Datasets

At the very beginning, we roughly define the two types of multi-view data mentioned in the question as follows:

1) **Multi-feature**: This kind of data originates from a single modality of target samples. In most cases, they are extracted by designing multiple features. For example, Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradient (HOG) features (two data views) can be extracted from a RGB image (one modality).
2) **Multi-modal**: This kind of data consists of multiple modalities of target samples. In most cases, they are collected from multiple sensors, sources, etc., such as the data comprised of x-rays plus blood tests which is mentioned in this question.

Note that, apart from *multi-modal*, *multi-feature* is an another classical type of *multi-view* concept. This can be proved by the wide application of multi-feature datasets [10], [38], [39] in multi-view literature.

In the following, we briefly introduce the view meanings of the used datasets:

1) **BBC**[2] [40] (*multi-feature*) is processed from documents of the BBC news website corresponding to stories in five topics, i.e. business, entertainment, politics, sport and technology. Each data view corresponds to a segment of the documents[3].
2) **BBCSport**[2] [40] (*multi-feature*) is processed from documents of the BBC Sport website corresponding to sport news in five topics, i.e. athletics, cricket, football, rugby and tennis. Each data view corresponds to a segment of the documents[3].
3) **CiteSeer**[4] [41] (*multi-modal*) contains four views, i.e. content, inbound, outbound and cites, of the documents. There are 3707 words in content view and 4732 links in the other views.
4) **Cora**[4] [41] (*multi-modal*) contains four views, i.e. content, inbound, outbound and cites, of the documents. There are 1433 words in content view and 5492 links in the other views.

2. http://mlg.ucd.ie/datasets/bbc.html
3. http://mlg.ucd.ie/datasets/segment.html
4. https://lig-membres.imag.fr/grimal/data.html

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



TABLE 3
Accuracy comparison of traditional kernel and CMK on kernel $k$-means. *Trad.* is the abbreviation of *Traditional* and *Avg.* refers to the average kernel of all data views. At the same time, the best results are marked in bold.

| Dataset | View | Gaussian | | Linear | | Polynomial | | Sigmoid | | Cauchy | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Trad. | CMK | Trad. | CMK | Trad. | CMK | Trad. | CMK | Trad. | CMK |
| BBC | 1 | 86.48 | **93.99** | 86.63 | **94.43** | 86.53 | **93.79** | 86.88 | **93.24** | 86.33 | **93.89** |
| | 2 | 86.23 | **93.89** | 86.38 | **93.79** | 86.23 | **93.04** | 86.48 | **92.20** | 85.88 | **94.14** |
| | Avg. | 91.45 | **93.99** | 91.00 | **94.23** | 91.50 | **93.99** | 91.30 | **93.89** | 91.35 | **93.89** |
| BBCSport | 1 | 89.89 | **93.01** | **90.44** | 82.35 | **89.71** | 88.60 | **92.46** | 82.17 | 89.15 | **94.30** |
| | 2 | 73.53 | **93.01** | 87.50 | **91.54** | 74.08 | **88.60** | **87.13** | 81.99 | 86.03 | **94.49** |
| | Avg. | 90.07 | **93.01** | 90.07 | **91.54** | **89.89** | 88.60 | **91.36** | 81.80 | 90.44 | **94.49** |
| CiteSeer | 1 | 26.54 | **52.69** | 34.42 | **55.34** | 27.11 | **53.32** | 27.72 | **51.33** | 26.75 | **51.75** |
| | 2 | 44.96 | **54.53** | 45.11 | **57.04** | 45.11 | **53.56** | 44.69 | **52.60** | 45.08 | **53.96** |
| | 3 | 22.68 | **38.62** | 23.88 | **38.44** | 22.13 | **38.50** | 22.52 | **37.80** | 21.35 | **37.71** |
| | 4 | 27.39 | **43.21** | 28.77 | **42.33** | 27.32 | **43.63** | 25.09 | **39.22** | 24.18 | **41.03** |
| | Avg. | 27.32 | **54.05** | 43.45 | **56.37** | 45.86 | **53.29** | 43.90 | **53.02** | 23.16 | **53.53** |
| Cora | 1 | 32.79 | **56.06** | 44.24 | **58.27** | 32.75 | **59.34** | 46.05 | **63.66** | 32.90 | **57.16** |
| | 2 | 34.31 | **61.48** | 34.45 | **66.06** | 34.34 | **54.73** | 34.49 | **64.03** | 34.31 | **61.30** |
| | 3 | 31.09 | **41.36** | 30.06 | **41.47** | 31.28 | **38.40** | 31.94 | **36.34** | 24.00 | **42.10** |
| | 4 | 32.98 | **55.02** | 37.11 | **51.26** | 33.09 | **57.53** | 39.25 | **44.09** | 32.94 | **51.77** |
| | Avg. | 28.66 | **62.11** | 45.20 | **58.83** | 43.54 | **54.65** | 47.97 | **64.73** | 28.40 | **61.41** |
| Movies | 1 | 27.07 | **29.01** | **28.53** | 26.74 | 26.58 | **29.98** | 27.88 | **30.31** | 28.53 | **29.50** |
| | 2 | 19.94 | **29.34** | 20.26 | **26.74** | 20.75 | **29.50** | 22.69 | **27.07** | 20.42 | **29.66** |
| | Avg. | 26.26 | **29.82** | 25.45 | **29.34** | 26.09 | **31.28** | 27.71 | **29.17** | 28.36 | **29.98** |
| AwA | 1 | 6.59 | **7.26** | 6.63 | **6.69** | 6.65 | **7.42** | 6.43 | **6.47** | **6.61** | 6.51 |
| | 2 | 6.22 | **7.07** | 6.09 | **6.53** | 6.14 | **7.22** | 6.00 | **6.00** | **6.62** | 6.37 |
| | Avg. | 6.72 | **7.79** | 6.42 | **7.06** | 6.42 | **7.74** | 6.77 | **6.96** | **6.77** | 6.63 |
| CCV | 1 | 19.40 | **21.66** | 18.13 | **19.15** | **19.09** | 17.58 | 16.89 | **17.82** | 19.19 | **23.68** |
| | 2 | 21.38 | **23.45** | 20.40 | **22.86** | 20.79 | **21.70** | 20.91 | **24.02** | 21.51 | **26.03** |
| | 3 | 18.19 | **23.52** | **17.89** | 17.73 | 17.89 | **18.96** | 16.85 | **17.84** | 17.76 | **23.08** |
| | Avg. | 24.36 | **25.93** | 24.38 | **24.58** | **24.39** | 22.65 | **23.87** | 23.43 | 23.53 | **27.52** |
| NusWide | 1 | 12.55 | **18.40** | 11.09 | **12.66** | 12.67 | **13.79** | **11.13** | 10.98 | **12.77** | 12.32 |
| | 2 | 11.28 | **20.23** | 9.91 | **10.69** | 10.58 | **11.84** | 9.54 | **9.99** | 10.94 | **11.26** |
| | 3 | 10.72 | **17.01** | 9.96 | **11.14** | 10.09 | **11.92** | 9.74 | **10.88** | 10.95 | **11.15** |
| | 4 | 11.47 | **22.15** | 10.95 | **11.40** | 11.08 | **12.14** | 10.39 | **12.01** | **11.33** | 10.27 |
| | 5 | 10.32 | **10.60** | 9.90 | **10.96** | 9.94 | **11.59** | 9.41 | **11.32** | **10.09** | 9.91 |
| | Avg. | **14.35** | 14.11 | **13.64** | 12.85 | 13.91 | **14.17** | 13.36 | **14.07** | **14.01** | 12.21 |
| YtVideo | 1 | 10.81 | **20.98** | 10.85 | **22.94** | 10.49 | **31.18** | 7.73 | **14.94** | 11.23 | **18.57** |
| | 2 | 52.70 | **61.44** | 53.63 | **61.82** | 52.17 | **64.29** | 37.06 | **54.90** | 38.72 | **62.43** |
| | 3 | 11.50 | **30.48** | 11.65 | **33.65** | 11.63 | **40.43** | 11.21 | **23.43** | 11.63 | **23.90** |
| | Avg. | 32.06 | **55.50** | 37.61 | **52.47** | 33.29 | **67.67** | 8.87 | **54.04** | 27.92 | **52.71** |
| CropLand | 1 | 43.59 | **57.97** | 43.60 | **58.11** | 43.58 | **67.59** | **43.12** | 37.65 | 43.57 | **57.35** |
| | 2 | 50.20 | **58.41** | 50.20 | **62.07** | 50.21 | **59.08** | 40.22 | **59.93** | 50.20 | **56.91** |
| | Avg. | 59.19 | **68.03** | 58.90 | **70.90** | 59.09 | **72.09** | 47.62 | **66.39** | 59.92 | **63.55** |

5) **Movies**[4] [41] (*multi-modal*) is extracted form IMDb[5] to have two data matrices with the first describing the movie keywords while the second describing the movie actors.
6) **AwA**[6] [42] (*multi-feature*) consists of animal images with pre-extracted feature representations. Note we only use the Color Histogram and Local Self-Similarity features here.
7) **CCV**[7] [43] (*multi-modal*) consists of three popular data features, including SIFT, Spatial-Temporal Interest Points (STIP), and Mel-Frequency Cepstral Coefficients (MFCC), where the first two are extracted from visual modality and the last is from audio modality.
8) **NusWide**[8] [44] (*multi-feature*) extracts five features, including Color Histogram (CH), Color Correlo-

5. https://www.imdb.com
6. https://cvml.ist.ac.at/AwA/
7. https://www.ee.columbia.edu/ln/dvmm/CCV/
8. https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



gram (CORR), Edge Direction Histogram (EDH), Wavelet Texture (WT) and Color Moments (CM), from a large set of web images.

9) **YtVideo**[9] [45] (*multi-modal*) extracts a set of visual, audio and text features from Youtube game videos. Here, we use HOG (visual), MFCC (audio), Latent Dirichlet Allocation (LDA) [46] (text) features.
10) **CropLand**[10] [47] (*multi-modal*) collects a large volume of images by RapidEye satellites (optical) and the Unmanned Aerial Vehicle Synthetic Aperture Radar (UAVSAR) system (Radar) over an agricultural region near Winnipeg, Manitoba, Canada on 2012. Correspondingly, two features are extracted.

Overall, we summarize the dataset specifications in Table 2.

### 5.2 CMK: kernel quality improvement

To evaluate the kernel quality, we adopt three common clustering metrics by applying standard kernel $k$-means on the generated kernels. The metrics are Accuracy (ACC), Normalized Mutual Information (NMI) and Purity. Their definitions are detailed in Appendix. We also generate traditional kernels and CMK with the same set of parameters as shown in Table 4 to ensure the fairness of comparison. Additionally, the dimension of latent representations $d$ and the learning rate $\alpha$ are set to $128$ and $1.0$ globally [11].

TABLE 4
Parameters of traditional kernel and CMK.

| Kernel Type | Parameter setting |
|---|---|
| Gaussian | $\sigma = 1$ |
| Linear | $a = 1,\ c = 0$ |
| Polynomial | $a = 1,\ c = 1,\ d = 2$ |
| Sigmoid | $a = 1,\ c = 1$ |
| Cauchy | $\sigma = 1$ |

For the small-scale datasets, including BBC, BBCSport, CiteSeer, Cora and Movies, we adopt the GD optimization strategy, while SGD strategy with batch size $1024$ is employed on AwA, CCV, NusWide, YtVedio and Cropland. In experiment, we apply kernel $k$-means on both traditional kernel and CMK of the ten datasets. Note that, for YtVideo and CropLand, Nyström technique [48] is employed to prevent from memory error. In specific, corresponding accuracy comparison is presented in Table 3, where the best results are marked in bold. We make the following observations.

1) CMK generation paradigm improves the kernel quality to a large extent. For example, it promotes about 5%, 6%, 20%, 23%, 3%, 1%, 3%, 5%, 15% and 10% accuracy of Gaussian kernel on ten datasets, respectively.
2) The results on some datasets and settings decrease, especially for the small-scale datasets, i.e. BBCSport and Movies. This may be caused by the random initialization of mapping weights $\mathbf{W}_v$ in Eq. (4). In the optimization, we use the Gradient Descent algorithm which may stop on bad local minimums. This problem can be eased by adopting a more robust optimization strategy, such as Adam [49]. Moreover, this may also be affected by the over-fit problem as discussed in section 5.4. Some performance decreases are also observed on Cauchy CMKs on AwA and NusWide. But it can be observed that they are much smaller than the improvements in other settings.
3) Accuracies of Gaussian and Linear CMKs consistently outperform those of traditional kernels, while the others are not. We leave this in future research.
4) The average kernel (average of several traditional kernels) has often been observed to be a simple yet tough baseline in kernel learning [35]. The reason is that averaging kernels integrates cross-view information. Nevertheless, CMK outperforms the traditional average kernel in most cases.

Nevertheless, the NMI and Purity values follow a similar trend and are shown in Appendix. Overall, we can conclude that the proposed CMK generation paradigm can improve kernel quality compared with traditional kernel generation approaches.

### 5.3 CMK: downstream task

Since the proposed method is a kernel generation paradigm, we also validate its effectiveness via comparing the performances of multiple kernel methods on CMKs and traditional kernels. The competing methods are:

1) **MKKM** [13] extends the well-known fuzzy $c$-means algorithm with multiple kernel learning framework, where the weights among kernels are adjusted automatically.
2) **RMKC** [18] proposes to clean the noise of input kernels and then aggregates them into a robust and low-rank consensus one.
3) **RMKKM** [50] performs robust $k$-means with an appropriate consensus kernel which is learned from a linear combination of input kernels. Meanwhile, all the variables are encapsulated by the non-smooth $L_{2,1}$-norm.
4) **MKCMR** [15] proposes a matrix-induced regularization to reduce the redundancies among kernels and improve the kernel diversity.
5) **ONKC** [51] finds that the representation capability of consensus kernel is limited by decomposing it into a linear weighted kernel combination. Thus, it locates the optimal kernel in the neighborhood area.
6) **LFAM** [24] first computes the base partition of each data view, then aligns them with a consensus partition, at last applies $k$-means to obtain the labels.

In addition, we use the codes which are publicly available on authors' websites. Also, corresponding parameters are grid-searched in the recommended ranges, and the best results are reported. Moreover, we inherit the settings in section 5.2 to generate kernel matrices.

9. https://archive.ics.uci.edu/ml/datasets/YouTube+Multiview+Video+Games+Dataset
10. https://archive.ics.uci.edu/ml/datasets/Crop+mapping+using+fused+optical-radar+data+set
11. We do not tune any parameter in experiment for practicality.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



TABLE 5
Accuracy comparison of traditional kernel and CMK on classical multiple kernel methods. *Trad.* is the abbreviation of *Traditional*. At the same time, the best results are marked in bold.

| Dataset | Alg. | Gaussian | | Linear | | Polynomial | | Sigmoid | | Cauchy | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Trad. | CMK | Trad. | CMK | Trad. | CMK | Trad. | CMK | Trad. | CMK |
| BBC | MKKM | 91.45 | **93.99** | 91.00 | **94.23** | 91.55 | **93.99** | 91.30 | **93.84** | 91.35 | **93.89** |
| | RMKC | 91.45 | **93.99** | 91.00 | **94.23** | 91.55 | **93.99** | 91.30 | **93.89** | 91.35 | **94.14** |
| | RMKKM | 91.80 | **92.94** | 91.90 | **93.89** | 91.95 | **93.69** | 90.76 | **92.64** | 92.54 | **93.39** |
| | MKCMR | 91.45 | **94.04** | 91.00 | **94.23** | 91.55 | **93.99** | 91.40 | **93.99** | 91.35 | **93.94** |
| | ONKC | 91.60 | **94.04** | 91.00 | **94.23** | 91.55 | **93.99** | 91.40 | **93.99** | 91.35 | **93.94** |
| | LFAM | 91.80 | **94.23** | 91.25 | **94.23** | 91.55 | **93.99** | 91.35 | **93.89** | 91.35 | **93.89** |
| BBCSport | MKKM | 90.07 | **93.01** | 90.07 | **91.36** | **89.89** | 88.60 | **91.36** | 81.80 | 90.44 | **94.49** |
| | RMKC | 90.07 | **93.01** | 90.07 | **91.54** | **89.89** | 88.60 | **92.46** | 82.17 | 90.44 | **94.49** |
| | RMKKM | 88.42 | **96.32** | 95.59 | **96.32** | 90.07 | **96.51** | **97.24** | 94.85 | 77.02 | **96.32** |
| | MKCMR | 90.07 | **93.01** | 90.07 | **91.54** | **90.07** | 88.60 | **91.36** | 81.80 | 90.44 | **94.49** |
| | ONKC | 90.26 | **93.01** | 90.07 | **91.54** | **90.07** | 88.60 | **91.36** | 81.80 | 90.44 | **94.49** |
| | LFAM | 90.81 | **93.01** | 90.81 | **91.54** | **90.26** | 88.60 | **91.54** | 81.80 | 90.44 | **94.49** |
| CiteSeer | MKKM | 23.22 | **54.05** | 47.89 | **57.19** | 44.72 | **53.41** | 43.42 | **53.35** | 23.16 | **53.05** |
| | RMKC | 45.35 | **54.53** | 43.33 | **56.37** | 45.92 | **53.35** | 43.93 | **53.08** | 49.82 | **54.11** |
| | RMKKM | 25.69 | **58.27** | **58.33** | 55.68 | 49.12 | **52.81** | **59.78** | 52.11 | 23.22 | **58.15** |
| | MKCMR | 46.62 | **54.44** | 47.86 | **57.40** | 45.65 | **53.35** | 44.69 | **53.44** | 46.04 | **53.80** |
| | ONKC | 34.72 | **54.50** | 49.40 | **57.49** | 49.37 | **53.32** | 49.82 | **54.35** | 27.87 | **54.08** |
| | LFAM | 43.45 | **54.05** | 43.60 | **56.37** | 46.59 | **53.35** | 43.96 | **53.26** | 33.06 | **53.74** |
| Cora | MKKM | 28.43 | **57.39** | 46.20 | **66.62** | 44.13 | **65.14** | 48.45 | **64.44** | 28.58 | **59.08** |
| | RMKC | 34.49 | **58.97** | 45.90 | **59.01** | 43.61 | **59.27** | 47.78 | **64.81** | 36.41 | **57.20** |
| | RMKKM | 30.10 | **67.43** | 46.05 | **68.32** | 46.20 | **70.72** | 41.88 | **66.25** | 29.69 | **61.41** |
| | MKCMR | 44.24 | **61.89** | 46.16 | **66.62** | 43.39 | **66.17** | 48.97 | **64.51** | 42.76 | **60.93** |
| | ONKC | 35.56 | **62.78** | 51.62 | **66.58** | 46.82 | **66.06** | 54.84 | **65.77** | 33.27 | **63.00** |
| | LFAM | 30.98 | **62.19** | 45.68 | **66.43** | 43.57 | **66.29** | 48.26 | **65.69** | 32.75 | **61.71** |
| Movies | MKKM | 28.20 | **29.34** | **29.50** | 28.53 | 27.88 | **28.04** | 27.39 | **28.69** | 25.77 | **30.47** |
| | RMKC | 27.23 | **30.31** | 25.45 | **29.98** | 28.36 | **31.12** | 25.93 | **31.12** | 27.23 | **31.28** |
| | RMKKM | 25.28 | **32.58** | 27.71 | **34.85** | 25.77 | **30.96** | **31.93** | 31.28 | 25.93 | **33.55** |
| | MKCMR | 26.58 | **30.63** | 28.04 | **30.15** | 28.04 | **29.98** | 29.17 | **31.60** | 28.36 | **30.47** |
| | ONKC | 29.01 | **31.77** | 29.01 | **31.60** | 28.53 | **33.39** | 30.63 | **31.28** | 28.85 | **32.09** |
| | LFAM | 26.09 | **30.15** | 26.26 | **30.47** | 26.74 | **30.96** | 26.26 | **30.96** | 28.20 | **31.12** |

Since the aforementioned multiple kernel methods are of cubic complexity, we only test them with the traditional kernels and CMKs of BBC, BBCSport, CiteSeer, Cora and Movies. Corresponding accuracies are presented in Table 5. Three observations can be obtained as follows:

1) Multiple kernel methods on Gaussian, Linear and Cauchy CMKs consistently outperform those on traditional kernels. For Gaussian CMKs, about 2%, 4%, 21%, 26% and 3% accuracy improvements are observed, demonstrating its effectiveness.
2) Although CMK shows weaker performances on a few settings (such as *BBCSport + Polynomial*), the gaps are relatively small. Meanwhile, CMK exceeds the traditional kernels in most Polynomial and Sigmoid settings.
3) The results of multiple kernel methods establish a similar tendency with kernel quality evaluation in Table 3, especially for the decreases of Linear CMK on Movies, Polynomial CMK on BBCSport, and Sigmoid CMK on BBCSport. This may be improved by adopting other optimization strategies.

Overall, CMK generation paradigm can effectively promote the performance of downstream tasks. Furthermore, the NMI and Purity values follow a similar trend and are presented in Appendix due to space limit.

### 5.4 CMK: insights of model building

In this section, we explore two extra properties of the CMK generation paradigm in the kernel learning process. Specifically, we apply kernel $k$-means on the generated kernels and record corresponding performances by epoch. For ease of expression, performance of the average Gaussian CMK on BBC, along with the loss value, is shown on the left of Fig. 2. It can be observed that the loss value continuously decreases in the training process. Meanwhile, accuracy, NMI, and purity increase with an opposite tendency. We can conclude that minimizing the loss function helps improve the kernel quality, demonstrating the consistency between loss design and our motivation.

We also visualize the differences among kernels and latent representations at each epoch in the middle of Fig. 2. Similarly, Gaussian CMK on BBC is taken for an instance. We can discover that their differences dramatically drop

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



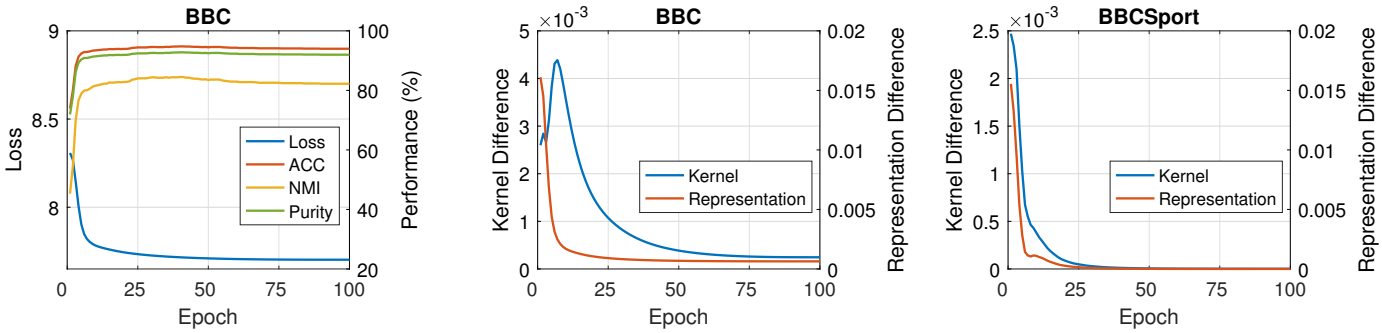



Fig. 2. Temporary measurements in model building, including loss value, performances, kernel difference and representation difference, by the example of Gaussian CMK on BBC and BBCSport, respectively.

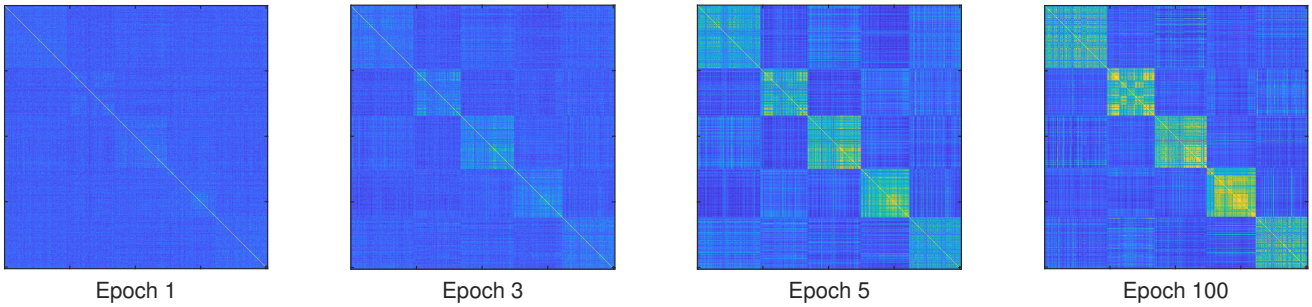



Fig. 3. Visualization of the Gaussian CMK on the 1st view of BBC dataset in the learning process before 100 epochs.

TABLE 6
Accuracy comparison among classical MKC methods, CMK$^+$ and CMKKM. Here, CMK$^+$ refers to conducting kernel $k$-means on average Gaussian CMK. At the same time, the best results are marked in bold.

| Dataset | MKKM | RMKC | RMKKM | MKCMR | ONKC | LFAM | CMK | CMKKM |
|---|---|---|---|---|---|---|---|---|
| BBC | 91.45 | 91.45 | 91.80 | 91.45 | 91.60 | 91.80 | 93.99 | **95.08** |
| BBCSport | 90.07 | 90.07 | 88.42 | 90.07 | 90.26 | 90.81 | 93.01 | **96.88** |
| CiteSeer | 23.22 | 45.35 | 25.69 | 46.62 | 34.72 | 43.45 | 54.05 | **60.93** |
| Cora | 28.43 | 34.49 | 30.10 | 44.24 | 35.56 | 30.98 | 62.11 | **68.28** |
| Movies | 28.20 | 27.23 | 25.28 | 26.58 | 29.01 | 26.09 | 29.82 | **33.06** |

from the top, then remain stable at constants, which can be explained in two-folds:

1) The decrease illustrates that minimizing the CMK loss motivates mapping functions to push the latent representations of different data views towards a consensus.
2) The stability at constants demonstrates that the learned latent representations keep view-specific information.

The decrease and stability are two consistent, instead of opposite, concepts in multi-view learning. In the fusion of multi-view data, we expect to not only enhance discriminative information of the shared part, but also encourage each view to hold view-specific information as a supplement for the shared. Results on BBC in Fig. 2 well achieve this expectation, indicating an effective learning state.

One potential problem of CMK may be that CMK can overfit on small-scale datasets. To further analyze this risk, we plot the kernel and representation differences of Gaussian CMK on BBCSport (shown on the right of Fig. 2). We observe that the differences decrease to zero. This means that the mapping functions encode multiple data views into the same latent representations, failing to balance the learning of shared information and the preservation of view-specific information, as discussed in section 5.4. But it is noteworthy that CMK outperforms traditional kernels even in this setting, as shown in Tables 3 and 5. We leave the more detailed study of this problem to future work.

We also visualize the Gaussian CMKs on the 1st view of BBC dataset before 100 epochs in Fig. 3. Since the element of CMK measures the similarity between latent data representations, we can see that the cluster structure of data samples are gradually enhanced along with model learning, verifying the effectiveness of CMK design.

### 5.5 CMKKM: performance improvement

To validate the benefits of jointly conducting CMK generation and kernel clustering ($\ell_c$ and $\ell_{\mathbf{K}}$ in Eq. (34)), we compare the accuracies between CMKKM and CMK$^+$ in Table 6. Note that CMK$^+$ refers to conducting kernel k -means on average Gaussian CMK, which differs from CMKKM only at whether MKKM loss $\ell_{\mathbf{K}}$ are employed

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



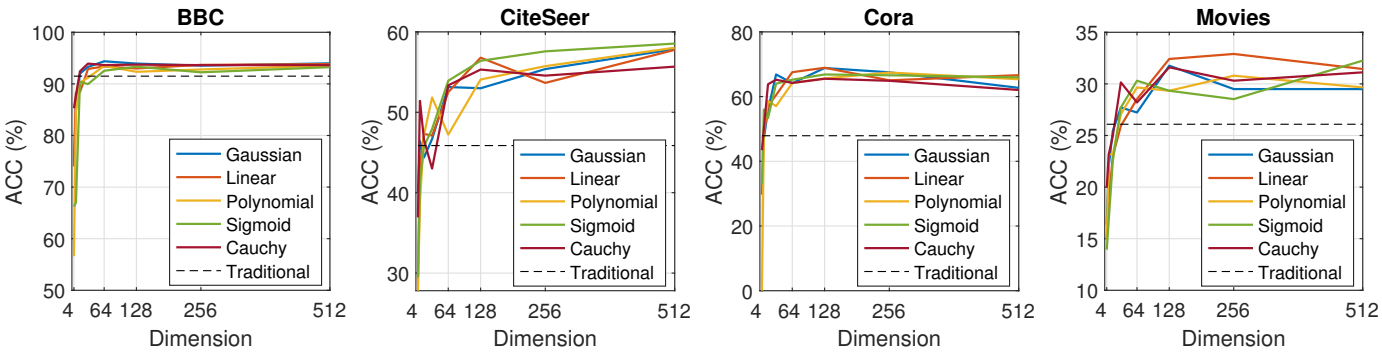



Fig. 4. Accuracy variations respect to the dimension $d$ of latent representations. The solid line represents different CMK types, while the black dotted line refers to the best result of traditional kernels.

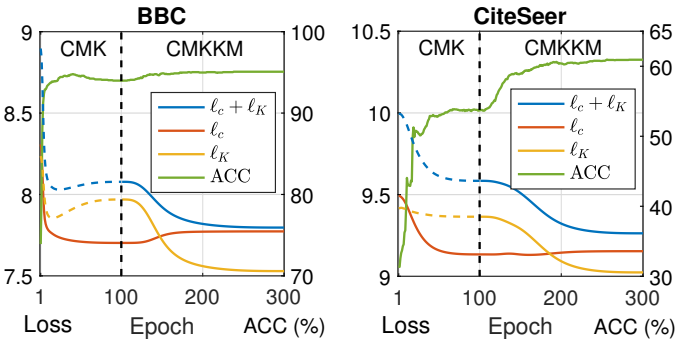



Fig. 5. Temporary measurements in model building, including loss value and performances, by the example of Contrastive Multiple Kernel $k$-means with Gaussian CMK on BBC and CiteSeer, respectively. The dashed line indicates corresponding loss is plotted for better understanding but not employed in the optimization.

in optimization. It can be seen that CMKKM outperforms CMK$^+$ by 1.09%, 3.87%, 6.88%, 6.17% and 3.24% on five benchmark datasets, respectively. This ablation study well verifies our proposal of CMKKM and contrastive multi-view clustering framework.

Nevertheless, we compare the experiment results of CMKKM with that of six representative MKC methods. It can be observed in Table 6 that CMKKM exceeds the best of comparative methods by 3.28%, 6.07%, 14.31%, 22.04% and 4.05%, demonstrating its effectiveness. By the way, CMK$^+$ also achieves promising results compared with the nine methods, which well illustrates the quality improvement of CMK.

### 5.6 CMKKM: insights of model building

By exploring the phenomenons in CMKKM model building, we, in the following, try to explain why unifying CMK generation with MKC task can help improve the clustering performance. In specific, two temporary measurements, including loss value and accuracy, are recorded in CMKKM optimization and the ones on BBC and CiteSeer are shown in Fig. 5. The model building is separated into two stages, i.e. CMK generation of the first 100 epochs (minimizing only $\ell_c$ in Eq. (34)) and CMKKM model learning of the last 200 epochs (minimizing both $\ell_c$ and $\ell_{\mathbf{K}}$in Eq. (34)), corresponding to line 1 and line 2-12 in Algorithm 2, respectively.

From the subplot of BBC, we can see CMK loss $\ell_c$ decreases dramatically at first, and then keep stable in the first 100 epochs. At the same time, the accuracy rises to the top at around 40-th epoch. These validate our proposal that minimizing the CMK loss can help improve the quality of resultant kernels. Also, two more observations are obtained: 1) The accuracy decreases from 40-th to 100-th epoch; 2) The MKKM loss $\ell_{\mathbf{K}}$ first falls at a large scale but then increases gradually. The two points illustrate that minimizing the CMK loss blindly would result in the kernel quality loss. When imposing MKKM loss on optimization in the last 200 epochs, it drops quickly. Meanwhile, the CMK loss rises slowly and keeps stable at last. So does the accuracy, indicating the MKKM loss can help improve kernel quality. Moreover, the results on CiteSeer share similar observations with that on BBC, but one can observe the accuracy improvement resulting from MKKM loss more clearly.

Overall, it can be concluded that CMK generation (minimizing CMK loss $\ell_c$) and MKC task (minimizing MKKM loss $\ell_{\mathbf{K}}$) are two independent but supplementary processes to each other. Jointly optimizing them in a unified framework would achieve an ideal learning state, leading to promising performance.

### 5.7 Parameter analysis

We conduct an ablation study on the dimension of the latent representation $\mathbf{z}_i^v$ to explore its effect on kernel quality. Keeping the learning rate constant (i.e. $\alpha = 1.0$), we vary the dimension from $2^2$ to $2^9$. As a result, performances on five types of CMK are obtained and the average accuracies are plotted in Fig. 4. Note that the black dotted line represents the best result achieved by traditional kernels. It can be seen that the accuracy starts increasing from a relatively low position. Especially, Polynomial CMK gets an error on Cora when the dimension is set to 4. This is caused by kernel $k$-means only separating the data into less than 7 clusters, contradictory to the ground truth. Meanwhile, all types of CMK increase dramatically and then stay relatively stable at wide ranges. Nevertheless, we observe that CMK outperforms the best traditional kernel when the dimension is larger than 32. Therefore, it can be concluded that the proposed paradigm is able to generate kernels of high quality even with a large dimension of the latent representation $\mathbf{h}_i^v$. We recommend setting the dimension to 128 or larger. At the

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



TABLE 7
Accuracy comparison of CMK (evaluating with kernel $k$-means) on the 1st view of BBC dataset. Note that, "-" indicates that the optimization reports an error, while 'Norm.' is the short for "Normalization".

| Type | Norm. | Learning Rate | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | 1e-5 | 1e-4 | 1e-3 | 0.01 | 0.1 | 1.0 | 10 | 100 |
| Gaussian | False | 66.45 | 67.15 | 71.72 | - | - | - | - | - |
| | True | 73.91 | 74.25 | 77.53 | 91.60 | 94.09 | 93.99 | 93.74 | 92.25 |
| Linear | False | 69.93 | 70.83 | 81.06 | - | - | - | - | - |
| | True | 72.37 | 73.21 | 81.01 | 93.04 | 94.28 | 94.18 | 93.29 | 92.59 |
| Polynomial | False | - | - | - | - | - | - | - | - |
| | True | 73.96 | 75.60 | 88.42 | 94.28 | 59.99 | 93.99 | 92.89 | 91.60 |
| Sigmoid | False | 72.17 | 73.71 | 86.93 | - | - | - | - | - |
| | True | 72.07 | 74.01 | 87.52 | 93.04 | 91.40 | 93.89 | 92.59 | 92.59 |
| Cauchy | False | - | - | - | - | - | - | - | - |
| | True | 74.35 | 74.40 | 76.59 | 90.01 | 93.79 | 93.89 | 93.84 | 92.79 |

TABLE 8
Accuracy comparison of CMK (evaluating with kernel $k$-means) and CMKKM on BBC dataset.

| Method | Epoch | Learning Rate | | | | |
|---|---|---|---|---|---|---|
| | | 0.01 | 0.1 | 1.0 | 10 | 100 |
| CMK | 50 | 85.69 | 93.44 | 92.20 | 93.29 | 92.15 |
| | 100 | 91.60 | 94.09 | 93.99 | 93.74 | 92.25 |
| | 150 | 92.74 | 94.43 | 94.48 | 93.94 | 92.45 |
| CMKKM | 150 | 92.35 | 94.09 | 94.48 | 93.84 | 92.35 |
| | 300 | 93.49 | 94.28 | 95.08 | 93.99 | 92.74 |
| | 450 | 93.84 | 94.53 | 95.03 | 94.23 | 93.29 |

same time, CMK establishes a stable quality improvement on traditional kernels, verifying its effectiveness again.

By grid-searching the epoch number and learning rate, we present the accuracy results in Table 8. It can be seen that both the CMK and CMKKM models achieve better performances with a larger training epoch number. Meanwhile, a large or small learning rate results in a visible performance decrease. The NMI and Purity results follow a similar trend and are shown in Appendix. Therefore, we recommend setting the learning rate, the epoch number of CMK and CMKKM models to $1.0$, $150$ and $450$.

## 6 Discussion

In this section, we first discuss the connection and differences between the proposed CMK loss and the widely-used contrastive loss [28], [35] as follows.

**Connection**. *Disregarding of the generation method of latent representation $\mathbf{z}_i^v$, the contrastive loss in the Normalized Temperature-scaled Cross Entropy (NT-Xent) form is a special case of the proposed Linear CMK loss*, where both of them intend to maximize the similarities between positive pairs and minimize those between negative pairs. When limiting the view number $V$ of the proposed CMK loss in Eq. (10) to 2, it is obvious that

$$\begin{aligned}\text{Eq. (10)} &= -\log \frac{\exp(k_z(\mathbf{z}_i^v, \mathbf{z}_i^{v'}))}{\sum_{j,v'' \in \mathcal{A}_{i,v}} \exp(k_z(\mathbf{z}_i^v, \mathbf{z}_j^{v''}))} \\ &= -\log \frac{\exp(k_z(\mathbf{z}_i, \mathbf{z}_{j(i)}))}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(k_z(\mathbf{z}_i, \mathbf{z}_k))},\end{aligned} \tag{41}$$

where the first column adopts $\mathbf{z}_i^v$ and $\mathbf{z}_i^{v'}$ to represent the positive sample pair, while the second uses $\mathbf{z}_i$ and $\mathbf{z}_{j(i)}$ to do so. Nevertheless, from Table 1 of the manuscript, $k_z(\mathbf{x}_i, \mathbf{x}_j) = a\mathbf{x}_i^\top \mathbf{x}_j + c$ for the linear CMK, resulting in

$$\begin{aligned}\text{Eq. (41)} &= -\log \frac{\exp(a\mathbf{z}_i^\top \mathbf{z}_{j(i)} + c)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(a\mathbf{z}_i^\top \mathbf{z}_{j(i)} + c)} \\ &= -\log \frac{\exp(a\mathbf{z}_i^\top \mathbf{z}_{j(i)})}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(a\mathbf{z}_i^\top \mathbf{z}_{j(i)})}.\end{aligned} \tag{42}$$

By setting $a = 1/\tau$, we can get

$$\begin{aligned}\text{Eq. (42)} &= -\log \frac{\exp(\mathbf{z}_i^\top \mathbf{z}_{j(i)}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(\mathbf{z}_i^\top \mathbf{z}_{j(i)}/\tau)} \\ &= -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_{j(i)})/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{k \neq i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}\end{aligned} \tag{43}$$

where $\text{sim}(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\top \mathbf{x}_j / \|\mathbf{x}_i\| \|\mathbf{x}_j\|$, and the last step holds for $\mathbf{z}_i$ is normalized in the proposed CMK. We can see Eq. (43) is exactly the contrastive loss of [28], [35].

**Difference**. We identify the novelty of the proposed CMK loss from three aspects:

1) Motivation. The contrastive loss of [28], [35] is designed to learn discriminative representations of images, while CMK tends to improve the kernel quality of multi-view data (the output is corresponding kernel matrices), which is novel in kernel learning but ignored by existing researches.
2) Loss design. The method [28], [35] is partially limited by the finite loss functions, such as NT-Xent, Margin Triplet, etc. Meanwhile, the proposed CMK loss is more flexible, where all types of kernel functions can be integrated by simply instancing $k_z(\cdot, \cdot)$.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



This also makes it compatible with the literature of kernel theory, such as kernel learning, kernel approximation, etc.

3) Encoding structure. Contrastive learning proposes to encoding images with an encoder $f(\cdot)$ and subsequent projection head $g(\cdot)$. However, it is based on images and not practical for data of vectors. Therefore, CMK simplifies the encoding design and projects multi-view data with $V$ independent weights $\{\mathbf{W}_v\}_{v=1}^{V}$.

Nevertheless, we explore the necessity of the normalization of latent representations in Eq. (5). By removing the normalization, we obtain the experiment results in Table 7. It can be observed that the CMK generation paradigm without normalization often reports an error, especially when learning rate is bigger than $0.01$ or Polynomial and Cauchy kernel functions are adopted. In such cases, we find the CMK's values are always "***NaN***" or "***Inf***", illustrating a trivial solution. Meanwhile, for Gaussian, Linear and Sigmoid CMK generation paradigms with learning rate smaller than $0.01$, the accuracies decrease rapidly once the normalization is removed. In sum, the normalization is essential in the proposed CMK.

## 7 Conclusion

Current multiple kernel learning methods compute kernels independently for each data view, ignoring the complementary information across views. We propose the Contrastive Multi-view Kernel generation paradigm, which integrates the views into a quality kernel with a high concordance across views while ensuring their diversity and heterogeneity. The experiments show that CMK generates more quality kernels than traditional methods. We also propose a Contrastive Multi-view Clustering framework and instantiate it with Multiple Kernel $k$-means, achieving promising performance. To our best knowledge, this is the first attempt to explore kernel generation and contrastive learning in multi-view setting, providing a new direction for future research.

## Acknowledgments

This work was supported by the National Key R&D Program of China 2020AAA0107100 and the National Natural Science Foundation of China (project no. 61922088, 61976196 and 62276271).

## References

[1] K. I. Kim, M. O. Franz, and B. Schölkopf, "Iterative kernel principal component analysis for image modeling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 9, pp. 1351–1366, 2005. [Online]. Available: https://doi.org/10.1109/TPAMI.2005.181

[2] N. Perveen, D. Roy, and C. K. Mohan, "Facial expression recognition in videos using dynamic kernels," *IEEE Trans. Image Process.*, vol. 29, pp. 8316–8325, 2020. [Online]. Available: https://doi.org/10.1109/TIP.2020.3011846

[3] H. Wang, Q. Wang, M. Gao, P. Li, and W. Zuo, "Multi-scale location-aware kernel representation for object detection," in *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, 2018, pp. 1248–1257. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Wang_Multi-Scale_Location-Aware_Kernel_CVPR_2018_paper.html

[4] Y. Chai, P. Sun, J. Ngiam, W. Wang, B. Caine, V. Vasudevan, X. Zhang, and D. Anguelov, "To the point: Efficient 3d object detection in the range image with graph convolution kernels," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 16 000–16 009. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Chai_To_the_Point_Efficient_3D_Object_Detection_in_the_Range_CVPR_2021_paper.html

[5] C. S. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: A string kernel for SVM protein classification," in *Proceedings of the 7th Pacific Symposium on Biocomputing, PSB 2002, Lihue, Hawaii, USA, January 3-7, 2002*, R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, Eds., 2002, pp. 566–575. [Online]. Available: http://psb.stanford.edu/psb-online/proceedings/psb02/leslie.pdf

[6] B. Schölkopf and A. J. Smola, *Learning with Kernels: support vector machines, regularization, optimization, and beyond*, ser. Adaptive computation and machine learning series. MIT Press, 2002. [Online]. Available: https://www.worldcat.org/oclc/48970254

[7] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning.*, ser. Adaptive computation and machine learning. MIT Press, 2006.

[8] S. Y. Kung, *Kernel methods for cluster analysis*. Cambridge University Press, 2014, p. 178–218.

[9] Z. Han, C. Zhang, H. Fu, and J. T. Zhou, "Trusted multi-view classification," in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. [Online]. Available: https://openreview.net/forum?id=OOsR8BzCnl5

[10] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, and D. Xu, "Generalized latent multi-view subspace clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 1, pp. 86–99, 2020. [Online]. Available: https://doi.org/10.1109/TPAMI.2018.2877660

[11] S. Sonnenburg, G. Rätsch, C. Schäfer, and B. Schölkopf, "Large scale multiple kernel learning," *J. Mach. Learn. Res.*, vol. 7, pp. 1531–1565, 2006. [Online]. Available: http://jmlr.org/papers/v7/sonnenburg06a.html

[12] X. Liu, X. Zhu, M. Li, L. Wang, E. Zhu, T. Liu, M. Kloft, D. Shen, J. Yin, and W. Gao, "Multiple kernel $k$-means with incomplete kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 5, pp. 1191–1204, 2020. [Online]. Available: https://doi.org/10.1109/TPAMI.2019.2892416

[13] H. Huang, Y. Chuang, and C. Chen, "Multiple kernel fuzzy clustering," *IEEE Trans. Fuzzy Syst.*, vol. 20, no. 1, pp. 120–134, 2012. [Online]. Available: https://doi.org/10.1109/TFUZZ.2011.2170175

[14] M. Kloft, U. Rückert, and P. L. Bartlett, "A unifying view of multiple kernel learning," in *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part II*, ser. Lecture Notes in Computer Science, J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, Eds., vol. 6322. Springer, 2010, pp. 66–81. [Online]. Available: https://doi.org/10.1007/978-3-642-15883-4_5

[15] X. Liu, Y. Dou, J. Yin, L. Wang, and E. Zhu, "Multiple kernel $k$-means clustering with matrix-induced regularization," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, D. Schuurmans and M. P. Wellman, Eds. AAAI Press, 2016, pp. 1888–1894. [Online]. Available: http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12115

[16] J. Liu, X. Liu, J. Xiong, Q. Liao, S. Zhou, S. Wang, and Y. Yang, "Optimal neighborhood multiple kernel clustering with adaptive local kernels," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 2020. [Online]. Available: https://doi.org/10.1109/TKDE.2020.3014104

[17] Z. Ren and Q. Sun, "Simultaneous global and local graph structure preserving for multiple kernel clustering," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 5, pp. 1839–1851, 2021. [Online]. Available: https://doi.org/10.1109/TNNLS.2020.2991366

[18] P. Zhou, L. Du, L. Shi, H. Wang, and Y. Shen, "Recovery of corrupted multiple kernels for clustering," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. J. Wooldridge, Eds. AAAI Press, 2015, pp. 4105–4111. [Online]. Available: http://ijcai.org/Abstract/15/576

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211




[19] A. Kumar, P. Rai, and H. D. III, "Co-regularized multi-view spectral clustering," in *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger, Eds., 2011, pp. 1413–1421. [Online]. Available: https://proceedings.neurips.cc/paper/2011/hash/31839b036f63806cba3f47b93af8ccb5-Abstract.html

[20] J. Wen, H. Sun, L. Fei, J. Li, Z. Zhang, and B. Zhang, "Consensus guided incomplete multi-view spectral clustering," *Neural Networks*, vol. 133, pp. 207–219, 2021. [Online]. Available: https://doi.org/10.1016/j.neunet.2020.10.014

[21] J. Wen, Z. Zhang, Y. Xu, and Z. Zhong, "Incomplete multi-view clustering via graph regularized matrix factorization," in *Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part IV*, ser. Lecture Notes in Computer Science, L. Leal-Taixé and S. Roth, Eds., vol. 11132. Springer, 2018, pp. 593–608. [Online]. Available: https://doi.org/10.1007/978-3-030-11018-5_47

[22] Z. Ren, S. X. Yang, Q. Sun, and T. Wang, "Consensus affinity graph learning for multiple kernel clustering," *IEEE Trans. Cybern.*, vol. 51, no. 6, pp. 3273–3284, 2021. [Online]. Available: https://doi.org/10.1109/TCYB.2020.3000947

[23] E. Bruno and S. Marchand-Maillet, "Multiview clustering: a late fusion approach using latent models," in *Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2009, Boston, MA, USA, July 19-23, 2009*, 2009, pp. 736–737. [Online]. Available: https://doi.org/10.1145/1571941.1572103

[24] S. Wang, X. Liu, E. Zhu, C. Tang, J. Liu, J. Hu, J. Xia, and J. Yin, "Multi-view clustering via late fusion alignment maximization," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, 2019, pp. 3778–3784. [Online]. Available: https://doi.org/10.24963/ijcai.2019/524

[25] J. Liu, X. Liu, Y. Yang, S. Wang, and S. Zhou, "Hierarchical multiple kernel clustering," in *Proceedings of the Thirty-fifth AAAI Conference on Artificial Intelligence, (AAAI-21), Virtually, February 2-9, 2021*, 2021.

[26] H. Q. Minh, P. Niyogi, and Y. Yao, "Mercer's theorem, feature maps, and smoothing," in *Learning Theory, 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006, Proceedings*, ser. Lecture Notes in Computer Science, G. Lugosi and H. U. Simon, Eds., vol. 4005. Springer, 2006, pp. 154–168. [Online]. Available: https://doi.org/10.1007/11776420_14

[27] M. Bouafia, D. Benterki, and A. Yassine, "An efficient parameterized logarithmic kernel function for linear optimization," *Optim. Lett.*, vol. 12, no. 5, pp. 1079–1097, 2018. [Online]. Available: https://doi.org/10.1007/s11590-017-1170-5

[28] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 1597–1607. [Online]. Available: http://proceedings.mlr.press/v119/chen20j.html

[29] Y. Lin, Y. Gou, Z. Liu, B. Li, J. Lv, and X. Peng, "COMPLETER: incomplete multi-view clustering via contrastive prediction," in *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*. Computer Vision Foundation / IEEE, 2021, pp. 11 174–11 183. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2021/html/Lin_COMPLETER_Incomplete_Multi-View_Clustering_via_Contrastive_Prediction_CVPR_2021_paper.html

[30] Y. Li, P. Hu, J. Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 8547–8555. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/17037

[31] B. Du, X. Gao, W. Hu, and X. Li, "Self-contrastive learning with hard negative sampling for self-supervised point cloud learning," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 3133–3142.

[32] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," in *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., 2020. [Online]. Available: https://proceedings.neurips.cc/paper/2020/hash/d89a66c7c80a29b1bdbab0f2a1a94af8-Abstract.html

[33] C. Yeh, C. Hong, Y. Hsu, T. Liu, Y. Chen, and Y. LeCun, "Decoupled contrastive learning," *CoRR*, vol. abs/2110.06848, 2021. [Online]. Available: https://arxiv.org/abs/2110.06848

[34] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XI*, ser. Lecture Notes in Computer Science, A. Vedaldi, H. Bischof, T. Brox, and J. Frahm, Eds., vol. 12356. Springer, 2020, pp. 776–794. [Online]. Available: https://doi.org/10.1007/978-3-030-58621-8_45

[35] J. Xu, H. Tang, Y. Ren, X. Zhu, and L. He, "Contrastive multi-modal clustering," *CoRR*, vol. abs/2106.11193, 2021. [Online]. Available: https://arxiv.org/abs/2106.11193

[36] X. Cai, F. Nie, and H. Huang, "Multi-view k-means clustering on big data," in *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, F. Rossi, Ed. IJCAI/AAAI, 2013, pp. 2598–2604. [Online]. Available: http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6979

[37] L. Chen, C. L. P. Chen, and M. Lu, "A multiple-kernel fuzzy c-means algorithm for image segmentation," *IEEE Trans. Syst. Man Cybern. Part B*, vol. 41, no. 5, pp. 1263–1274, 2011. [Online]. Available: https://doi.org/10.1109/TSMCB.2011.2124455

[38] M. Yin, J. Gao, S. Xie, and Y. Guo, "Multiview subspace clustering via tensorial t-product representation," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 3, pp. 851–864, 2019. [Online]. Available: https://doi.org/10.1109/TNNLS.2018.2851444

[39] H. Wang, Y. Yang, and B. Liu, "GMC: graph-based multi-view clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 6, pp. 1116–1129, 2020. [Online]. Available: https://doi.org/10.1109/TKDE.2019.2903810

[40] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, ser. ACM International Conference Proceeding Series, W. W. Cohen and A. W. Moore, Eds., vol. 148. ACM, 2006, pp. 377–384. [Online]. Available: https://doi.org/10.1145/1143844.1143892

[41] G. Bisson and C. Grimal, "Co-clustering of multi-view datasets: A parallelizable approach," in *12th IEEE International Conference on Data Mining, ICDM 2012, Brussels, Belgium, December 10-13, 2012*, M. J. Zaki, A. Siebes, J. X. Yu, B. Goethals, G. I. Webb, and X. Wu, Eds. IEEE Computer Society, 2012, pp. 828–833. [Online]. Available: https://doi.org/10.1109/ICDM.2012.93

[42] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*. IEEE Computer Society, 2009, pp. 951–958. [Online]. Available: https://doi.org/10.1109/CVPR.2009.5206594

[43] Y. Jiang, G. Ye, S. Chang, D. P. W. Ellis, and A. C. Loui, "Consumer video understanding: a benchmark database and an evaluation of human and machine performance," in *Proceedings of the 1st International Conference on Multimedia Retrieval, ICMR 2011, Trento, Italy, April 18 - 20, 2011*, F. G. B. D. Natale, A. D. Bimbo, A. Hanjalic, B. S. Manjunath, and S. Satoh, Eds. ACM, 2011, p. 29. [Online]. Available: https://doi.org/10.1145/1991996.1992025

[44] T. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: a real-world web image database from national university of singapore," in *Proceedings of the 8th ACM International Conference on Image and Video Retrieval, CIVR 2009, Santorini Island, Greece, July 8-10, 2009*, S. Marchand-Maillet and Y. Kompatsiaris, Eds. ACM, 2009. [Online]. Available: https://doi.org/10.1145/1646396.1646452

[45] O. Madani, M. Georg, and D. A. Ross, "On using nearly-independent feature families for high precision and confidence," *Mach. Learn.*, vol. 92, no. 2-3, pp. 457–477, 2013. [Online]. Available: https://doi.org/10.1007/s10994-013-5377-0

[46] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, no. null, p. 993–1022, mar 2003.


© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.

This article has been accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2023.3253211



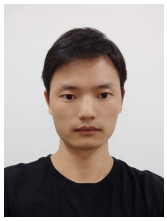
[47] I. Khosravi and S. K. Alavipanah, "A random forest-based framework for crop mapping using temporal, spectral, textural and polarimetric observations," *International Journal of Remote Sensing*, vol. 40, no. 18, pp. 7221–7251, 2019. [Online]. Available: https://doi.org/10.1080/01431161.2019.1601285

[48] D. Oglic and T. Gärtner, "Nyström method with kernel k-means++ samples as landmarks," in *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 2017, pp. 2652–2660. [Online]. Available: http://proceedings.mlr.press/v70/oglic17a.html

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[50] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y. Shen, "Robust multiple kernel k-means using l21-norm," in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Q. Yang and M. J. Wooldridge, Eds. AAAI Press, 2015, pp. 3476–3482. [Online]. Available: http://ijcai.org/Abstract/15/489

[51] X. Liu, S. Zhou, Y. Wang, M. Li, Y. Dou, E. Zhu, and J. Yin, "Optimal neighborhood kernel clustering with multiple kernels," in *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, S. P. Singh and S. Markovitch, Eds. AAAI Press, 2017, pp. 2266–2272. [Online]. Available: http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14761



**Jiyuan Liu** is a PhD student in National University of Defense Technology (NUDT), China. His current research interests include multi-view clustering, deep clustering and anomaly detection. He has published papers in journals and conferences such as IEEE T-KDE, IEEE T-NNLS, ICML, CVPR, ICCV, ACMMM, AAAI, IJCAI, etc. He serves as program committee member and reviewer on IEEE T-KDE, IEEE T-NNLS, NeurIPS, ICML, CVPR, ICCV, ACMMM, AAAI, IJCAI, etc. More information can be found at https://liujiyuan13.github.io/.

**Xinwang Liu** received his PhD degree from National University of Defense Technology (NUDT), China. He is now Professor at School of Computer, NUDT. His current research interests include kernel learning and unsupervised feature learning. Dr. Liu has published 60+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-KDE, IEEE T-IP, IEEE T-NNLS, IEEE T-MM, IEEE T-IFS, NeurIPS, CVPR, ICCV, AAAI, IJCAI, etc. More information can be found at https://xinwangliu.github.io/.

**Yuexiang Yang** received the B.S. degree in Mathematics from Xiangtan University, Xiangtan, China, in 1986, the M.S. degree in Computer Application and the PHD degree in Computer Science and Technology from National University of Defense Technology, Changsha, China, in 1989 and 2008, respectively. His research interests include information retrieval, network security and data analysis. He is the executive director of the Information Branch of China Higher Education Association. He has co-authored more than 100 papers in international journals and conference or workshop proceedings. He has been serving as reviewer and program committee member of various conferences and journals.

**Qing Liao** received her Ph.D. degree in computer science and engineering in 2016 supervised by Prof. Qian Zhang from the Department of Computer Science and Engineering of the Hong Kong University of Science and Technology. She is currently a professor with School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. Her research interests include artificial intelligence and data mining.

**Yuanqing Xia** received the Ph.D. degree in control theory and control engineering from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2001. He is currently the Dean of the School of Automation, Beijing Institute of Technology. He has published eight monographs in Springer, Wiley, and CRC, and over 100 papers in international scientific journals. His current research interests include networked control systems, robust control and signal processing, active disturbance rejection control and flight control. Dr. Xia was a recipient of the National Science Foundation for Distinguished Young Scholars of China in 2012, the Second Award of the Beijing Municipal Science and Technology (No. 1) in 2010 and 2015, the Second National Award for Science and Technology (No. 2) in 2011, and the Second Natural Science Award of the Ministry of Education (No. 1) in 2012. He is a Deputy Editor of the Journal of Beijing Institute of Technology, an Associate Editor of Acta Automatica Sinica, Control Theory and Applications, the International Journal of Innovative Computing, Information and Control, and the International Journal of Automation and Computing. In 2016, he was honored as the Yangtze River Scholar Distinguished Professor and was supported by National High Level Talents Special Support Plan (Million People Plan) by the Organization Department of the CPC Central Committee.

© 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
Authorized licensed use limited to: National Univ of Defense Tech. Downloaded on March 07,2023 at 02:07:46 UTC from IEEE Xplore. Restrictions apply.