

ProLFA: Representative Prototype Selection for Local Feature Aggregation

Xingxing Zhang^{a,b}, Zhenfeng Zhu^{a,b}, Yao Zhao^{a,b,*}

^a*Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China*

^b*Beijing Key Laboratory of Advanced Information Science and Network Technology*

Abstract

Given a set of hand-crafted local features, acquiring a global representation via aggregation is a promising technique to boost computational efficiency and improve task performance. Existing feature aggregation (FA) approaches, including Bag of Words and Fisher Vectors, usually fail to capture the desired information due to their pipeline mode. In this paper, we propose a generic formulation to provide a systematical solution (named **ProLFA**) to aggregate local descriptors. It is capable of producing compact yet interpretable representations by selecting representative prototypes from numerous descriptors, under relaxed exclusivity constraint. Meanwhile, to strengthen the discriminability of the aggregated representation, we rationally enforce the domain-invariant projection of bundled descriptors along a task-specific direction. Furthermore, ProLFA is also provided with a powerful generalization ability to deal flexibly with the semi-supervised and fully supervised scenarios in local feature aggregation. Experimental results on various descriptors and tasks demonstrate that the proposed ProLFA is considerably superior over currently available alternatives about feature aggregation.

Keywords: prototype selection, feature aggregation, block coordinate descent, domain-invariant projection.

*Corresponding author

Email addresses: zhangxing@bjtu.edu.cn (Xingxing Zhang), zhfhzhu@bjtu.edu.cn (Zhenfeng Zhu), yzhao@bjtu.edu.cn (Yao Zhao)

1. Introduction

A typical visual task (e.g. classification or retrieval), which uses hand-crafted features, consists of the following components: local feature extraction (e.g. SIFT [1] and DAISY [2]), local feature aggregation (e.g. Bag of Words [3] and Fisher Vectors [4]), and classification/retrieval/regression of the aggregated representations. This work focuses on the second component, *i.e.*, the produce of compact yet representative representations from the local features. The aggregated results can not only reduce the computation memory of post-processing [5], but also acquire the most valuable information of each sample. Furthermore, the performance of classification or retrieval task can be dramatically improved.

The problem of feature aggregation, generally referring to encoding and pooling of a series of local descriptors, has been well-studied in the literature [3, 4, 6]. Specifically, given a set of local descriptors, we first obtain a codebook by clustering all the descriptors, where the codebook is actually the set of cluster centers (also named codewords). Then, for each descriptor, we can find its most related codewords, and the statistics with respect to these codewords. Based on such information, each descriptor can be encoded as a new descriptor. Finally, we can obtain a global representation for the entire image by pooling the new descriptors belonging to that image. For example, Bag of Words (BoW) [3] firstly quantizes every local descriptor according to a codebook that is commonly learned with K-Means [7]. Then BoW represents each image as a histogram of codewords. The success of BoW aggregation prompted several extensions, including Fisher Vector (FV) coding [4], Super Vector (SV) coding [8], Locality-constrained Linear (LL) coding [9], Vector of Locally Aggregated Descriptors (VLAD) [10], Vector of Locally Aggregated Tensors (VLAT) [11], and spatial fisher vectors [12]. By these approaches, a set of local features can be aggregated into a single vector.

The success of these two-step methods, also called pipeline modes, naturally leads to the first question—are there better encoding/pooling alternatives? Second, the majority of existing methods completely rely on a visual codebook, which results in several attempts to improve the aggregation performance by improving the codebook. For example, [13] proposed a K-Means alternative that improves modelling of sparse regions of the local feature space. [14] made direct use of the class labels in order to improve the BoW representation using a classifier. Recently, a spatially efficient yet accurate feature aggregation method [15] called Sum of Sparse Binary codes

aggregation (SSB) is proposed, in which a set of sparse binary codes is aggregated by simple summing into a compact feature vector. Specifically, a family of local feature aggregation functions was defined in [16] for any task that can be expressed as a differentiable cost function minimization problem. However, they still suffer from imperfect performance due to the negligence of the intrinsic structure among local descriptors. Third, most of existing supervised aggregation methods are often particularly designed for a specific category recognition task, such as retrieval or classification [17, 18, 19, 20], thus limiting their applications [21]. We will address the mentioned three issues in following sections.

Additionally, to improve the interpretability of codebook, we consider the process of finding codebook as the task of **Prototype Selection (PS)**, which aims at finding exemplar samples from a feature collection. PS has been actively discussed in other fields [22], such as video summarization and product recommendation, since it holds several advantages over data storage, compression, synthesis and cleansing. Besides helping to reduce the computational time and memory of algorithms, due to working on several prototypical samples, PS has further improved performances of numerous applications. Compared to dictionary learning methods such as K-Means [7] and K-SVD [23, 24], that learn centers/atoms in the input-space, PS methods [25] choose centers/atoms from the given samples, such as Kmedoids [26] and Affinity Propagation [27].

In summary, the superiority of our prototype selection to conventional codebook learning lies in four key points. (i) Unlike those unsupervised codebook learning approaches (*e.g.*, BoW [3] and VLAD [10]), the prototypes selected by our ProLFA are directly related to ultimate tasks (*e.g.*, object recognition and image retrieval) by enforcing a task-specific projection. Thus, the aggregated features based on such prototypes are rich and discriminative enough to perform various tasks, yet compact to represent the entire image. (ii) Our prototype selection can work out not only in fully supervised feature aggregation scenarios like those supervised codebook learning approaches (*e.g.*, UniVCG [17] and DBoWs [18]), but also in semi-supervised scenarios. To the best of our knowledge, our work is the first attempt to obtain discriminative aggregated features in semi-supervised scenarios. (iii) More importantly, instead of the popular used clustering strategy for codebook learning, we design an algorithmic feature aggregation formulation, where the diversity and representativeness of selected prototypes are explicitly formulated as an exclusivity constraint. Consequently, we can guarantee the

quality of selected prototypes from the ultimate goal, which facilitates the interpretability of aggregated features on ultimate tasks. (iv) In particular, our ProLFA can alleviate the influence of class unbalance on aggregated features, since the quality of selected prototypes can be enforced even in class unbalanced case.

1.1. Contribution

To the best of our knowledge, our work describes the first attempt to jointly optimize the two key goals: FA interpretability and discrimination for final tasks. In summary, the main contributions of this work are highlighted as follows.

- i) We develop ProLFA: a **P**rototype selection induced **L**ocal **F**eature **A**ggregation (ProLFA) model, to aggregate a set of hand-crafted local features effectively for various tasks (*e.g.*, image search and recognition).
- ii) The most representative prototypes are selected from numerous local descriptors under relaxed exclusivity constraint, thus facilitating the interpretability of aggregated representations.
- iii) By enforcing the domain-invariant projection of bundled descriptors along a task-specific direction, we can strengthen the discriminability of aggregated representations. Additionally, our model can deal flexibly with the semi-supervised and fully supervised scenarios in local feature aggregation.
- iv) A composite Block Coordinate Descent (cBCD) algorithm is customized to effectively seek the optimal solution of ProLFA. Experimental results have demonstrated our method works better than most of aggregation methods on a variety of features and tasks.

2. Related Work

Depending on the order of statistics that connect codebook and local descriptors during encoding phase, feature aggregation approaches can be divided into two categories. The first group of approaches mainly leverages first-order information, and usually encodes a local descriptor with weighted linear sum of related codewords prior to aggregation. Representative approaches include BoW [3], DBoWs [18], SC [28], and LC-KSVD2 [24]. By contrast, the second group of approaches generally uses higher-order statistics

(*e.g.*, density, mean, and variances) that are computed from local descriptors and related codewords. Although this kind of approach, such as FV [4], SV [8], VLAD [10], and VLAT [11], takes more advantage, they still suffer from the influence of pipeline mode, where the codebook learning is independent from feature aggregation. To address this issue, γ -democratic [29] exploited the relationship between democratic pooling and spectral normalization in the context of second-order features, and then proposed an aggregation approach in an end-to-end manner. In addition, based on shallow aggregation approaches, several neural network based aggregation methods have also been proposed. FV+NN [30] proposed a hybrid architecture for image classification that took the advantages of FV [4] and deep convolutional neural network (CNN) pipelines. It dramatically improved over previous FV systems without incurring the high complexity with respect to CNNs. Likewise, inspired by VLAD [10], NetVLAD [31] was proposed that is pluggable into any CNN architecture for weakly supervised tasks. Our ProLFA can be considered as a combination of these two categories of FA approaches. This is due to: (i) Our ProLFA finds codebook by selecting prototypes from all local descriptors, where prototypes serve as dictionary. (ii) The density and mean of local descriptors are all involved in our ProLFA via the group term.

According to whether auxiliary information about each sample is involved during codebook produce, existing FA models are often learned in an unsupervised or a supervised manner. Early representative approaches, such as BoW [3] and VLAD [10], used unsupervised clustering algorithm to cluster the set of features and learn a dictionary. These unsupervised approaches achieved promising results and produced codebooks that were generic enough to be used for any tasks. However, learning a discriminative and task-oriented codebook is expected to perform significantly better. Therefore, some supervised aggregation approaches, such as $T_1(\cdot)$ [16] and EO-BoW [19] were proposed. By supervised dictionary learning, such approaches produce discriminative codebooks that are useful for the given classification or retrieval task. However, they cannot achieve the optimal performances simultaneously on all tasks. For instance, $T_1(\cdot)$ [16] could aggregate a highly discriminative representation for classification tasks, but it is not optimal for retrieval since it severely distorts the similarity between images in order to gain discriminability. It is worth noting that, our ProLFA is designed for any tasks (*e.g.*, image classification, retrieval, annotation, or question answering), but is able to deal flexibly with the semi-supervised and fully supervised scenarios in local feature aggregation.

3. The Proposed Model

In this section, we firstly develop a ProLFA model to produce a global representation from a set of local descriptors, and then derive the algorithm to solve ProLFA.

3.1. Model Formulation

Suppose that we have m samples $\{(\mathbf{X}_i, \mathbf{y}_i) : i = 1, \dots, m\}$ (*e.g.*, images or texts), where $\mathbf{X}_i \in \mathbb{R}^{d \times N_i}$ is the set of N_i local descriptors in \mathbb{R}^d extracted from the i^{th} sample, and $\mathbf{y}_i \in \mathbb{R}^c$ is the corresponding response vector of the i^{th} sample. Meanwhile, we denote $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$ as the set of $\{\mathbf{X}_i\}_{i=1}^m$, where $N = \sum_{i=1}^m N_i$. It is obvious that there exists much redundancy and irrelevance among these local descriptors. In order to improve the performance of final tasks (*e.g.*, classification or retrieval), and meanwhile save the computational time, producing a global representation $\bar{\mathbf{x}}_i \in \mathbb{R}^{\bar{d}}$ for the i^{th} sample is necessary. For this end, we propose an aggregation function $\Psi(\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_m)$, whose minimization over all possible aggregated representation set $\bar{\mathbf{X}} = \{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_m\}$, *i.e.*,

$$\min_{\{\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_m\}} \Psi(\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_m) \quad (1)$$

needs to achieve two goals of (i) maximizing the discrimination of $\bar{\mathbf{X}}$; (ii) enhancing the interpretability of $\bar{\mathbf{X}}$.

As shown in Figure 1, we consider a decomposition of the aggregation function Ψ into two functions Φ_{reg} and Φ_{gen} with respect to the two aforementioned goals, as

$$\Psi(\bar{\mathbf{x}}_1, \dots, \bar{\mathbf{x}}_m) := \sum_{i=1}^m \Phi_{\text{reg}}(\underbrace{\Phi_{\text{gen}}(\mathbf{X}_i, \mathbf{Z})}_{:=\bar{\mathbf{x}}_i}, \mathbf{y}_i; \mathbf{W}) + \Phi_{\text{con}}(\mathbf{W}, \mathbf{Z}), \quad (2)$$

where $\bar{\mathbf{x}}_i = \Phi_{\text{gen}}(\mathbf{X}_i, \mathbf{Z})$, \mathbf{Z} is a prototype selection matrix, and Φ_{gen} denotes the global representation *generation function* that aims to produce interpretable representations by selecting the most prototypical local descriptors. \mathbf{W} is a projection matrix from the feature space to the semantic space, and Φ_{reg} denotes the *regression function* that favors producing discriminative representations from \mathbf{X} by maximally minimizing the regression error. Φ_{con} represents the constraints imposed on \mathbf{Z} and \mathbf{W} . Next we study each function in (2).

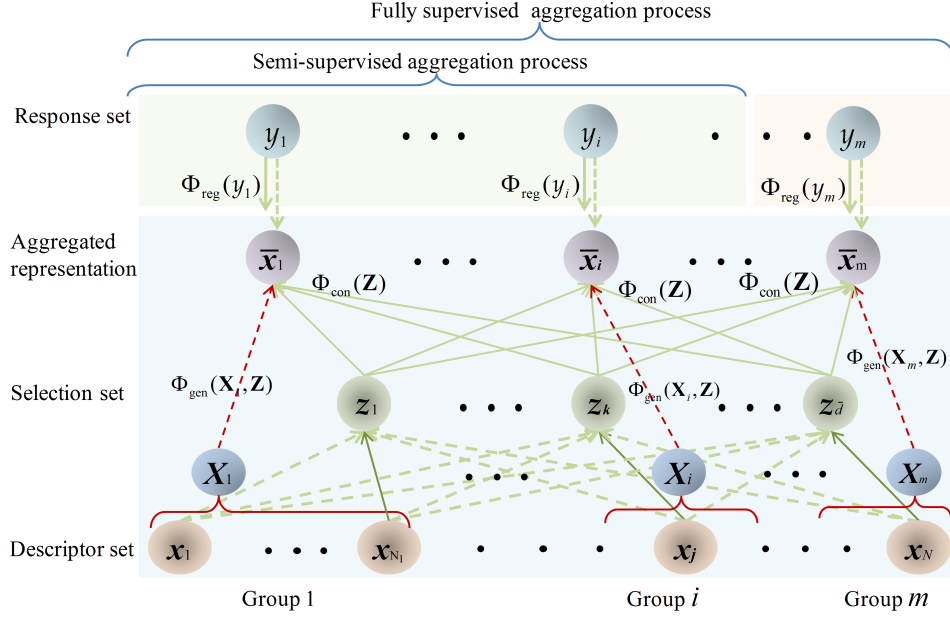


Figure 1: Illustration of the proposed ProLFA model. Given a collection of local descriptors, we can obtain the global representation for each sample by generation function Φ_{gen} . Specifically, representative prototypes are selected from numerous descriptors via selection matrix \mathbf{Z} , thus facilitating the interpretability of aggregated representations. Furthermore, we impose the response vector set on the aggregated representations via regression function Φ_{reg} to strengthen their discriminability.

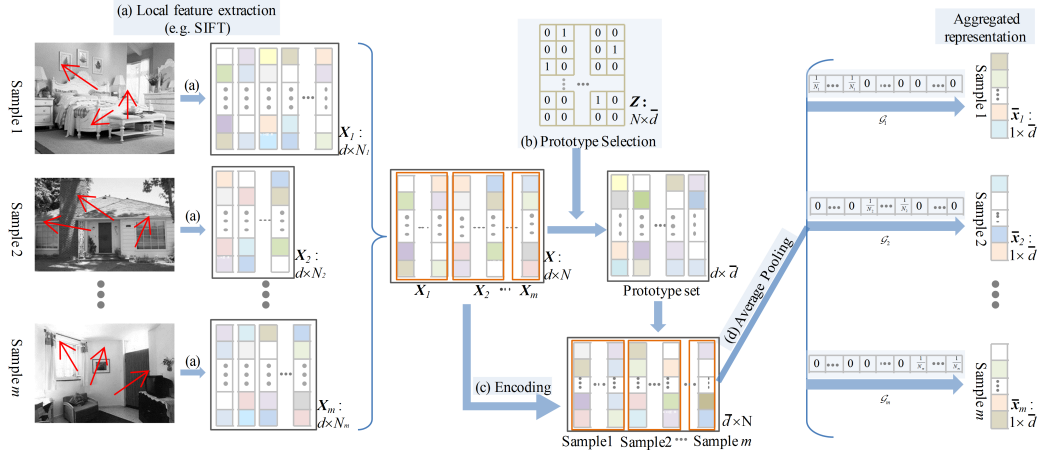


Figure 2: Framework of local feature aggregation by the function Φ_{gen} .

Generation Function. Instead of conventional codebook learning by clustering, we select the most prototypical descriptors to serve as codebook, where the diversity and representativeness of prototypes are guaranteed from the ultimate goal (*e.g.*, object recognition and image retrieval). This can provide a clear physical meaning (*i.e.*, interpretability) for aggregated features. For example, we can directly find their most related descriptors as well as locations, and meanwhile, aggregated features are representative enough to represent original images to perform the task. For this end, we integrate the codebook learning and feature encoding into a unified framework, instead of performing separately as in most existing aggregation approaches. Concretely, the generation function Φ_{gen} is cast into a three layers network that can be formulated as:

$$\bar{\mathbf{x}}_i = \Phi_{\text{gen}}(\mathbf{X}_i, \mathbf{Z}) = \underbrace{\mathcal{G}_i \mathbf{X}^T \underbrace{\mathbf{XZ}}_{\substack{\text{Prototypes} \\ \text{Encoding}}}}_{\text{Pooling}}, \quad (3)$$

where $\mathbf{Z} = [\mathbf{z}_1, \dots, \mathbf{z}_{\bar{d}}] \in \mathbb{R}^{N \times \bar{d}}$ denotes a selection matrix, $\bar{d} \ll N$, $\mathbf{Z} \in \{0, 1\}^{N \times \bar{d}}$ and $\mathbf{1}^T \mathbf{Z} = \mathbf{1}^T$.¹ $\mathcal{G}_i \in \mathbb{R}^N$ aims to exploit intrinsic structure of \mathbf{X} (including density and mean) by bundling the local descriptors in the i^{th} sample. Depends on the bundle strategy, the construction of \mathcal{G}_i involves soft and hard forms². In this work, we consider the hard one, *i.e.*, for any sample \mathbf{X}_i , and $j = 1, \dots, N$,

$$(\mathcal{G}_i)^j := \begin{cases} \frac{1}{N_i}, & \text{if } \mathbf{x}_j \in \mathbf{X}_i, \\ 0, & \text{otherwise.} \end{cases} \quad (4)$$

As shown in Figure 2, instead of clustering the set of local descriptors as in previous works [3], the \bar{d} selected prototypes with the properties of diversity and representativeness serve as codebook, on which a compact representation $\bar{\mathbf{x}}_i$ is produced via average pooling.

Regression Function. We introduce a domain-invariant projection, namely visual feature self reconstruction. Specifically, after projecting an

¹ $\mathbf{1}$ denotes a vector, of appropriate dimension, whose elements are all equal to one.

²Soft and hard forms imply each element in \mathcal{G}_i is in the range $[0, \frac{1}{N_i}]$ and $\{0, \frac{1}{N_i}\}$, respectively.

aggregated feature vector into a semantic embedding space, it should be able to be projected back in the reverse direction to reconstruct itself. Such a strategy, similar to that used in autoencoder, can improve the model generalization ability as demonstrated in other problems [32, 33]. Assuming that the forward and reverse projections have the same importance for feature aggregation, our regression function is then written as:

$$\Phi_{\text{reg}}(\bar{\mathbf{x}}_i, \mathbf{y}_i; \mathbf{W}) = \|\bar{\mathbf{x}}_i \mathbf{W} - \mathbf{y}_i\|_2^2 + \|\bar{\mathbf{x}}_i - \mathbf{y}_i \mathbf{W}^T\|_2^2, \quad (5)$$

where $\mathbf{W} \in \mathbb{R}^{\bar{d} \times c}$ denotes a projection matrix, and $\bar{\mathbf{x}}_i$ is the i^{th} normalized compact representation.

Our motivation can be explained as follows: (i) Adding the losses of the forward and reverse projections imposes a self-reconstruction constraint on our regression function, which can improve the model generalization ability. In our feature aggregation problem, this improved generalization ability makes the prototype selection matrix more applicable to the test samples. (ii) By regression instead of typical classifiers, our ProLFA can perform various tasks besides classification, with different definitions of response vector \mathbf{y} , such as *image annotation*, *question answering*, and *image classification* when \mathbf{y} represents *captions*, *answers*, and *labels*, respectively.

Constraint Function. The constraint function Φ_{con} is enforced both on the selection matrix \mathbf{Z} and projection matrix \mathbf{W} . First, to characterize the representativeness of selected prototypes, we formulate an orthogonality constraint $\mathbf{Z}^T \mathbf{Z} = \mathbf{I}$, *i.e.*, $\|\mathbf{z}_i \odot \mathbf{z}_j\|_0 = 0$ for $i, j \in \{1, \dots, \bar{d}\}$ and $i \neq j$, which leads to a diversified selection³. Second, to enhance the stability of solution and mitigate the scale issue, we formulate a ℓ_F^2 regularizer for \mathbf{W} , *i.e.*, $\|\mathbf{W}\|_F^2$.

Using all the functions defined above, we can rewrite the **Prototype** selection based **Local Feature Aggregation** model (ProLFA) in (1) as

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{W}} \quad & \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{Z} \mathbf{W} - \mathbf{y}_i\|_2^2 + \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{Z} - \mathbf{y}_i \mathbf{W}^T\|_2^2 + \lambda \|\mathbf{W}\|_F^2 \\ \text{s.t.} \quad & \mathbf{1}^T \mathbf{Z} = \mathbf{1}^T; \quad \mathbf{Z} \in \{0, 1\}^{N \times \bar{d}}; \quad \mathbf{Z}^T \mathbf{Z} = \mathbf{I}, \end{aligned} \quad (6)$$

where the regularization parameter $\lambda > 0$ sets the trade-off between the three terms in the objective function.

³ \odot designates the Hadamard product.

Due to the non-convexity and discontinuity of (6), we have the following relaxation

$$\begin{aligned}
& \min_{\mathbf{Z}, \mathbf{W}} \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{Z} \mathbf{W} - \mathbf{y}_i\|_2^2 + \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{Z} - \mathbf{y}_i \mathbf{W}^T\|_2^2 \\
& + 2\lambda_1 \underbrace{\sum_{j=1}^{\bar{d}} \sum_{i=1, i \neq j}^{\bar{d}} \|\mathbf{z}_i \odot \mathbf{z}_j\|_1}_{\text{Relaxed Exclusivity}} + \lambda_2 \|\mathbf{W}\|_F^2 \\
& s.t. \quad \mathbf{1}^T \mathbf{Z} = \mathbf{1}^T; \quad \mathbf{Z} \geq \mathbf{0},
\end{aligned} \tag{7}$$

where λ_1 and λ_2 are nonnegative regularization parameters. Compared (7) with (6), we first relax $\mathbf{Z} \in \{0, 1\}^{N \times \bar{d}}$ with $\mathbf{Z} \in [0, 1]^{N \times \bar{d}}$. Second, instead of directly employing the constraint $\mathbf{Z}^T \mathbf{Z} = \mathbf{I}$, we adopt the relaxed exclusivity constraint from a practical point of view, which is derived as follows:

$$\mathbf{Z}^T \mathbf{Z} = \mathbf{I} \Rightarrow \min \sum_{j=1}^{\bar{d}} \sum_{i=1, i \neq j}^{\bar{d}} \|\mathbf{z}_i \odot \mathbf{z}_j\|_0 \Rightarrow \min \sum_{j=1}^{\bar{d}} \sum_{i=1, i \neq j}^{\bar{d}} \|\mathbf{z}_i \odot \mathbf{z}_j\|_1. \tag{8}$$

Therefore, the task of local feature aggregation is converted into an optimization program with respect to \mathbf{Z} and \mathbf{W} .

3.2. Optimization Framework

In order to efficiently solve the proposed ProLFA model in (7), we further rewrite it as the following equivalent

$$\begin{aligned}
& \min_{\mathbf{Z}, \mathbf{W}} \mathcal{J}(\mathbf{Z}, \mathbf{W}) = \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{Z} \mathbf{W} - \mathbf{y}_i\|_2^2 + \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{Z} - \mathbf{y}_i \mathbf{W}^T\|_2^2 \\
& + \lambda_1 \left(\|\mathbf{Z}\|_{1,2}^2 - \|\mathbf{Z}\|_F^2 \right) + \lambda_2 \|\mathbf{W}\|_F^2 \\
& s.t. \quad \mathbf{1}^T \mathbf{Z} = \mathbf{1}^T; \quad \mathbf{Z} \geq \mathbf{0},
\end{aligned} \tag{9}$$

where the relaxed exclusivity term in (7) is replaced with the trick in Definition 1.

Definition 1.

$$\|\mathbf{Z}\|_{1,2}^2 := \sum_{i=1}^N \left(\sum_{j=1}^{\bar{d}} |z_{ij}| \right)^2 = \|\mathbf{Z}\|_F^2 + 2 \sum_{j=1}^{\bar{d}} \sum_{i=1, i \neq j}^{\bar{d}} \|\mathbf{z}_i \odot \mathbf{z}_j\|_1. \tag{10}$$

The objective function in (9) includes three convex terms and two mixed terms. Although not jointly convex in (\mathbf{Z}, \mathbf{W}) , it is convex with respect to each unknown when the other is fixed. This is why Block Coordinate Descent (BCD) on \mathbf{Z} and \mathbf{W} performs reasonably well [34], although not necessarily providing the global optimum. A composite BCD (cBCD) solver consists therefore of iterating between *Updating \mathbf{Z}* by fixing \mathbf{W} , and *Updating \mathbf{W}* by fixing \mathbf{Z} . Below are the solutions to two subproblems.

\mathbf{Z} subproblem: If \mathbf{W} is fixed, the subproblem in (9) with respect to \mathbf{Z} is written as

$$\begin{aligned} \min_{\mathbf{Z}} \quad & \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{Z} \mathbf{W} - \mathbf{y}_i\|_2^2 + \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{Z} - \mathbf{y}_i \mathbf{W}^T\|_2^2 \\ & + \lambda_1 \left(\|\mathbf{Z}\|_{1,2}^2 - \|\mathbf{Z}\|_F^2 \right) \\ \text{s.t.} \quad & \mathbf{1}^T \mathbf{Z} = \mathbf{1}^T; \quad \mathbf{Z} \geq \mathbf{0}. \end{aligned} \quad (11)$$

Considering the separability of both objective and constraints in (11), we employ the Alternating Direction Method of Multipliers (ADMM) framework to solve this subproblem. To do so, we introduce an auxiliary matrix $\mathbf{C} \in \mathbb{R}^{N \times \bar{d}}$ and consider the following equivalent optimization program

$$\begin{aligned} \min_{\mathbf{Z}, \mathbf{C}} \quad & \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{C} \mathbf{W} - \mathbf{y}_i\|_2^2 + \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{C} - \mathbf{y}_i \mathbf{W}^T\|_2^2 \\ & + \lambda_1 \left(\|\mathbf{Z}\|_{1,2}^2 - \|\mathbf{Z}\|_F^2 \right) \\ \text{s.t.} \quad & \mathbf{1}^T \mathbf{C} = \mathbf{1}^T; \quad \mathbf{C} \geq \mathbf{0}; \quad \mathbf{Z} = \mathbf{C}. \end{aligned} \quad (12)$$

Augmenting the last equality constraint of (12) to the objective function via the Lagrange multiplier matrix $\mathbf{\Lambda} \in \mathbb{R}^{N \times \bar{d}}$ and a positive penalty scalar μ , we can write the Lagrangian function as

$$\begin{aligned} \mathcal{L}(\mathbf{Z}, \mathbf{C}, \mathbf{\Lambda}) = & \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{C} \mathbf{W} - \mathbf{y}_i\|_2^2 + \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{C} - \mathbf{y}_i \mathbf{W}^T\|_2^2 \\ & + \frac{\mu}{2} \|\mathbf{Z} - \mathbf{C}\|_F^2 + \langle \mathbf{\Lambda}, \mathbf{Z} - \mathbf{C} \rangle + \lambda_1 \left(\|\mathbf{Z}\|_{1,2}^2 - \|\mathbf{Z}\|_F^2 \right). \end{aligned} \quad (13)$$

- Minimizing (13) with respect to \mathbf{Z} can be done using an effective iteratively

re-weighted algorithm [35]. Concretely, we have

$$\begin{aligned} \mathbf{Z}^{(t+1)} = & \operatorname{argmin}_{\mathbf{Z}} \lambda_1 \left(\|\mathbf{Z}\|_{1,2}^2 - \|\mathbf{Z}\|_F^2 \right) \\ & + \frac{\mu}{2} \|\mathbf{Z} - \mathbf{C}^{(t)}\|_F^2 + \langle \mathbf{\Lambda}^{(t)}, \mathbf{Z} - \mathbf{C}^{(t)} \rangle. \end{aligned} \quad (14)$$

As observed from (14), it can be split into N independent smaller optimization programs over the N rows of \mathbf{Z} . For each row vector $\mathbf{Z}_{\cdot j}$, we resolve the following equivalent objective:

$$\begin{aligned} \mathbf{Z}_{\cdot j}^{(t+1)} = & \operatorname{argmin}_{\mathbf{Z}_{\cdot j}} \lambda_1 \mathbf{Z}_{\cdot j} \mathbf{F} \mathbf{Z}_{\cdot j}^T + \frac{\mu}{2} \|\mathbf{Z}_{\cdot j} - \mathbf{C}_{\cdot j}^{(t)}\|_2^2 \\ & + \langle \mathbf{\Lambda}_{\cdot j}^{(t)}, \mathbf{Z}_{\cdot j} - \mathbf{C}_{\cdot j}^{(t)} \rangle, \end{aligned} \quad (15)$$

where $\mathbf{F} \in \mathbb{R}^{\bar{d} \times \bar{d}}$ is a diagonal matrix and formed by

$$\mathbf{F} := \operatorname{Diag} \left(\left[\frac{\|\mathbf{Z}_{\cdot j}\|_1}{|\mathbf{Z}_{\cdot j}(1)| + \epsilon} - 1, \dots, \frac{\|\mathbf{Z}_{\cdot j}\|_1}{|\mathbf{Z}_{\cdot j}(\bar{d})| + \epsilon} - 1 \right] \right), \quad (16)$$

where $\epsilon \rightarrow 0^+$ (in the experiments, we use 10^{-1}) is introduced to avoid zero denominators. With \mathbf{F} fixed, by equating the partial derivative of (15) with respect to $\mathbf{Z}_{\cdot j}$ to zero, we obtain

$$\mathbf{Z}_{\cdot j}^{(s+1)} = \left(\mu \mathbf{C}_{\cdot j}^{(s)} - \mathbf{\Lambda}_{\cdot j}^{(s)} \right) \left(\mu I + 2\lambda_1 \mathbf{F}^{(s)} \right)^{-1}. \quad (17)$$

Then $\mathbf{F}_{\cdot j}^{(s+1)}$ is updated using $\mathbf{Z}_{\cdot j}^{(s+1)}$ as in (16). In an iterative way, the optimal value $\mathbf{Z}_{\cdot j}^{(t+1)}$ is obtained.

- Minimizing (13) with respect to \mathbf{C} subject to the probability simplex constraints $\{\mathbf{1}^T \mathbf{C} = \mathbf{1}^T, \mathbf{C} \geq \mathbf{0}\}$ can be solved as follows:

$$\begin{aligned} \mathbf{C}^{(t+1)} = & \operatorname{argmin}_{\{\mathbf{1}^T \mathbf{C} = \mathbf{1}^T, \mathbf{C} \geq \mathbf{0}\}} \mathcal{L}(\mathbf{C}) \\ \approx & \operatorname{argmin}_{\{\mathbf{1}^T \mathbf{C} = \mathbf{1}^T, \mathbf{C} \geq \mathbf{0}\}} \left\| \mathbf{C} - \left(\mathbf{C} - \frac{1}{L} \frac{\partial \mathcal{L}(\mathbf{C})}{\partial \mathbf{C}} \right) \right\|_{\mathbf{C} = \mathbf{C}^{(t)}}^2, \end{aligned} \quad (18)$$

where $\mathcal{L}(\mathbf{C}) = \sum_{i=1}^m (\|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{C} \mathbf{W}^{(k)} - \mathbf{y}_i\|_2^2 + \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{C} - \mathbf{y}_i \mathbf{W}^{(k)T}\|_2^2) + \frac{\mu}{2} \|\mathbf{Z}^{(t+1)} - \mathbf{C}\|_F^2 + \langle \mathbf{\Lambda}^{(t)}, \mathbf{Z}^{(t+1)} - \mathbf{C} \rangle$. L is an upper bound of the Lipschitz

Algorithm 1: \mathbf{Z} and \mathbf{C} Solver using ADMM

Input: $\{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^m, \{\mathcal{G}_i\}_{i=1}^m, \mathbf{W}^{(k)}, \mathbf{Z}^{(k)}, \mathbf{C}^{(k)}, \mu$.
initialization $t \leftarrow 0; \mathbf{\Lambda}^{(t)}; \mathbf{Z}^{(t)} \leftarrow \mathbf{Z}^{(k)}; \mathbf{C}^{(t)} \leftarrow \mathbf{C}^{(k)}$;
while *not converged* **do**
 for $j = 0 : N$ **do**
 initialization $s \leftarrow 0$;
 while *not converged* **do**
 Update $\mathbf{F}^{(s+1)}$ via Eq. (16);
 Update $\mathbf{Z}_{:,j}^{(s+1)}$ via Eq. (17);
 $s \leftarrow s + 1$;
 end
 end
 $\mathbf{Z}^{(t+1)} \leftarrow \mathbf{Z}^{(s)}$;
 Update $\mathbf{C}^{(t+1)}$ via Eq. (18);
 Update $\mathbf{\Lambda}^{(t+1)}$ via Eq. (19);
 $t \leftarrow t + 1$;
end
Output: $\mathbf{Z}^{(k+1)} \leftarrow \mathbf{Z}^{(t)}, \mathbf{C}^{(k+1)} \leftarrow \mathbf{C}^{(t)}$.

constant of $\frac{\partial \mathcal{L}(\mathbf{C})}{\partial \mathbf{C}}$. By splitting (18) into \bar{d} independent smaller programs over the \bar{d} columns of \mathbf{C} , the algorithm⁴ in [36] can be employed to solve each subproblem with respect to each column vector $\mathbf{C}_i^{(t+1)}$.

- The multiplier matrix is updated by:

$$\mathbf{\Lambda}^{(t+1)} = \mathbf{\Lambda}^{(t)} + \mu (\mathbf{Z}^{(t+1)} - \mathbf{C}^{(t+1)}). \quad (19)$$

For clarity, the procedure of solving the subproblem in (11) is outlined in Algorithm 1. Convergence is achieved when we have $\|\mathbf{Z}^{(t+1)} - \mathbf{C}^{(t+1)}\|_\infty \leq \epsilon$ and $\|\mathbf{Z}^{(t+1)} - \mathbf{Z}^{(t)}\|_\infty \leq \epsilon$.

\mathbf{W} subproblem: Given \mathbf{C} , the subproblem in (9) with respect to \mathbf{W} is written as

$$\begin{aligned} \min_{\mathbf{W}} \quad & \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{C}^{(k)} \mathbf{W} - \mathbf{y}_i\|_2^2 + \sum_{i=1}^m \|\mathcal{G}_i \mathbf{X}^T \mathbf{X} \mathbf{C}^{(k)} - \mathbf{y}_i \mathbf{W}^T\|_2^2 \\ & + \lambda_2 \|\mathbf{W}\|_F^2. \end{aligned} \quad (20)$$

⁴The details are presented in Section 3 of [36].

Algorithm 2: ProLFA Implementation using cBCD

Input: $\{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^m, \{\mathcal{G}_i\}_{i=1}^m, \lambda_1, \lambda_2, \bar{d}$.
 initialization $k \leftarrow 0; \mathbf{W}^{(k)}; \mathbf{Z}^{(k)} = \mathbf{C}^{(k)}$;
while *not converged* **do**
 Update $\mathbf{Z}^{(k+1)}$ and $\mathbf{C}^{(k+1)}$ via Algorithm 1;
 Update $\mathbf{W}^{(k+1)}$ via Eq. (21);
 $k \leftarrow k + 1$;
end
Output: $\mathbf{Z}^* \leftarrow \mathbf{Z}^{(k)}, \mathbf{W}^* \leftarrow \mathbf{W}^{(k)}$.

By equating the partial derivative of (20) with respect to \mathbf{W} to zero, we obtain a linear equation as follows:

$$\mathbf{A}^{(k)} \mathbf{W}^{(k+1)} + \mathbf{W}^{(k+1)} \mathbf{B}^{(k)} = \mathbf{Q}^{(k)}, \quad (21)$$

where $\mathbf{A}^{(k)} = \sum_{i=1}^m \bar{\mathbf{x}}_i^{(k)\top} \bar{\mathbf{x}}_i^{(k)} + \lambda_2 \mathbf{I}$, $\mathbf{B}^{(k)} = \sum_{i=1}^m \mathbf{y}_i^\top \mathbf{y}_i$, $\mathbf{Q} = 2 \sum_{i=1}^m \bar{\mathbf{x}}_i^{(k)\top} \mathbf{y}_i$, and $\bar{\mathbf{x}}_i^{(k)} = \mathcal{G}_i \mathbf{X}^\top \mathbf{X} \mathbf{C}^{(k)}$. (20) is a Sylvester equation and it can be solved efficiently by the Bartels-Stewart algorithm [37].

In summary, Algorithm 2 shows the steps of the cBCD implementation of the ProLFA model in (9). The algorithm should not be terminated until the change of objective value is smaller than a pre-defined threshold (*e.g.*, 10^{-1}). For a new sample $\mathbf{X}_{\text{new}} \in \mathbb{R}^{d \times N_{\text{new}}}$, we finally obtain its corresponding global representation $\bar{\mathbf{x}}_{\text{new}}$ as follows

$$\bar{\mathbf{x}}_{\text{new}} = \Phi_{\text{gen}}(\mathbf{X}_{\text{new}}, \mathbf{Z}^*) = \mathcal{G}_{\text{new}} \mathbf{X}_{\text{new}}^\top \mathbf{X} \mathbf{Z}^*, \quad (22)$$

where \mathcal{G}_{new} is the weight information of N_{new} descriptors in this new sample.

4. Extension to Semi-supervised Aggregation

With the emergence of large-scale data, the available labeled (or annotated) samples are usually inadequate for some tasks. As shown in Figure 1, only n samples $\{\mathbf{X}_{\mathbb{I}_j} : j = 1, \dots, n; \mathbb{I}_j \in \{1, \dots, m\}\}$ among the whole dataset $\{\mathbf{X}_i : i = 1, \dots, m\}$ are labeled (or annotated) with the corresponding response vectors $\{\mathbf{y}_{\mathbb{I}_j}\}$, where $n \ll m$ generally. To deal flexibly with this semi-supervised scenario in local feature aggregation, our ProLFA model

in (9) is reformulated as follows

$$\begin{aligned}
\min_{\mathbf{Z}, \mathbf{W}} & \sum_{j=1}^n \|\mathcal{G}_{\mathbb{I}_j} \mathbb{X}^T \mathbf{X} \mathbf{Z} \mathbf{W} - \mathbf{y}_{\mathbb{I}_j}\|_2^2 + \sum_{j=1}^n \|\mathcal{G}_{\mathbb{I}_j} \mathbb{X}^T \mathbf{X} \mathbf{Z} - \mathbf{y}_{\mathbb{I}_j} \mathbf{W}^T\|_2^2 \\
& + \lambda_1 \left(\|\mathbf{Z}\|_{1,2}^2 - \|\mathbf{Z}\|_F^2 \right) + \lambda_2 \|\mathbf{W}\|_F^2 \\
s.t. & \quad \mathbf{1}^T \mathbf{Z} = \mathbf{1}^T; \quad \mathbf{Z} \geq \mathbf{0},
\end{aligned} \tag{23}$$

where \mathbb{X} and \mathbf{X} denote the sets of $\{\mathbf{X}_{\mathbb{I}_j}\}_{j=1}^n$ and $\{\mathbf{X}_i\}_{i=1}^m$, respectively. $\mathcal{K}(\mathbb{X}, \mathbf{X}) = \mathbb{X}^T \mathbf{X}$ is essentially a linear kernel matrix. $\mathcal{G}_{\mathbb{I}_j}$ is weight of each descriptor in the \mathbb{I}_j -th labeled (or annotated) sample $\mathbf{X}_{\mathbb{I}_j}$. By employing Algorithm 2, we can obtain the optimal solutions $\mathbf{Z}^* \in \mathbb{R}^{N \times \bar{d}}$ and $\mathbf{W}^* \in \mathbb{R}^{\bar{d} \times c}$. Furthermore, the corresponding global representation set $\bar{\mathbb{X}}_{\mathbb{U}} = \{\bar{\mathbf{x}}_{\mathbb{I}_j}\}_{j=n+1}^m$ for unlabeled or unannotated samples in embedding space is

$$\bar{\mathbf{x}}_{\mathbb{I}_j} = \Phi_{\text{gen}}(\mathbf{X}_{\mathbb{I}_j}, \mathbf{Z}^*) = \mathcal{G}_{\mathbb{I}_j} \mathbb{X}_{\mathbb{U}}^T \mathbf{X} \mathbf{Z}^*, \tag{24}$$

where $\mathbb{X}_{\mathbb{U}}$ is the set of unlabeled or unannotated samples $\{\mathbf{X}_{\mathbb{I}_j}\}_{j=n+1}^m$.

5. Discussions

The convergence, complexity, and scalability are analysed in this section. **Convergence Analysis.** The convergence behavior of our proposed cBCD algorithm is summarized as Theorem 1.

Theorem 1. *The sequence of $\{\mathcal{J}(\mathbf{Z}^{(k)}, \mathbf{W}^{(k)})\}$, i.e., the energy of the objective in (9), generated by the proposed cBCD optimizer (Algorithm 2) converges monotonically.*

Proof: In terms of energy, the optimization nature of BCD ensures that [38]:

$$\mathcal{J}(\mathbf{Z}^{(k)}, \mathbf{W}^{(k)}) \geq \mathcal{J}(\mathbf{Z}^{(k+1)}, \mathbf{W}^{(k)}) \geq \mathcal{J}(\mathbf{Z}^{(k+1)}, \mathbf{W}^{(k+1)}).$$

In other words, the energy gradually decreases as the involved two steps iterate. Further, the whole objective function (9) has a lower bound. Therefore, Algorithm 2 is guaranteed to converge monotonically.

Complexity Analysis. We consider using P parallel processing resources to solve the proposed optimization problem in (9). Thereby, updating each row of \mathbf{Z} takes $\mathcal{O}(\alpha_1 \bar{d})$ and $\mathcal{O}(\alpha_1 \bar{d}^2)$ for (16) and (17) respectively,

where α_1 is the (inner) iteration number in Algorithm 1. Specifically, due to the diagonalization of \mathbf{F} , the inverse operator in (17) only needs $\mathcal{O}(\bar{d})$. Updating \mathbf{C} takes $\mathcal{O}(N \lceil \bar{d}/P \rceil)$ for (18) using the randomized algorithm in [36]. Therefore, the cost of Algorithm 1 is $\mathcal{O}(\alpha_2(\alpha_1 \bar{d}^2 \lceil N/P \rceil + N \lceil \bar{d}/P \rceil))$, where α_2 is the number of (outer) iterations required to converge. Given \mathbf{A} , \mathbf{B} and \mathbf{Q} , solving \mathbf{W} via (21) spends $\mathcal{O}(\bar{d}^3 + c^3)$. The computation of $\bar{\mathbf{x}}_i$, \mathbf{A} and \mathbf{Q} has a time complexity of $\mathcal{O}(\bar{d}N)$, $\mathcal{O}(\bar{d}^2 m)$, and $\mathcal{O}(\bar{d}cm)$, respectively. Due to $N \gg \bar{d} > c$, Algorithm 2 has the complexity of $\mathcal{O}(\alpha_3(\alpha_2(\alpha_1 \bar{d}^2 \lceil N/P \rceil + N \lceil \bar{d}/P \rceil) + \bar{d}(\bar{d}^2 + \bar{d}m + N)))$, where α_3 is the number of iterations required to converge.

Scalability Analysis. As observed from the complexity analysis, the complexity of our algorithm shows a roughly linearly increasing timing result, which is a general case for some typical works in feature aggregation (*e.g.*, BoW). Thus, the proposed model has some limitations for directly dealing with a large set of local features. To alleviate this issue, we can resort to splitting the unlabeled (or unannotated) data \mathbb{X}_U into multiple batches, on which prototypes can be selected recursively. Ideally, its feasibility and effectiveness by this way can be expected.

6. Experimental Verification

6.1. Experimental Setup

For a given image, we apply the Hessian-affine detector [39] to detect multiple features. For each detected local feature, we then compute two types of local descriptors, SIFT [1] and DAISY [2]. The compared aggregation approaches include unsupervised ones (BoW [3], VLAD [10], FV [4], SC [28], $\phi_\Delta + \psi_d + \text{RN}$ [39], and DM [40]). Meanwhile, five supervised aggregation approaches, $T_1(\cdot)$ [16], UniVCG [17], LC-KSVD2 [24], DBoWs [18], and EO-BoW [19], are also employed for comparison. It is worth noting that the proposed model focuses on the semi-supervised and supervised feature aggregation. However, most of existing typical works rely on unsupervision. Thus, we additionally compared some unsupervised ones to verify the effectiveness of the introduced response information. Conducting such a comparison is also common in existing supervised aggregation works, *e.g.*, UniVCG [17] and EO-BoW [19]. For clarity, we denote our semi-supervised model in (23) as Semi-ProLFA. To avoid the influence of randomness, we average the results over 6 times of execution with different training set selections. Additionally,

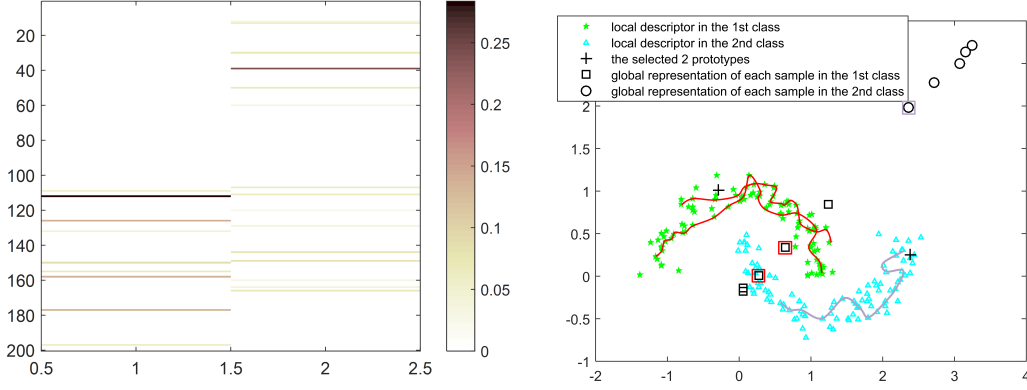


Figure 3: Left: selection matrix Z . Right: the aggregated linearly separable representations by ProLFA, where each curve represents 20 local descriptors included in each sample.

we have tried to yield better performances of all compared approaches by tuning the related parameters.

6.2. Synthetic Dataset

Figure 3 visualizes the selection matrix Z about 2 prototypes, and aggregated results by ProLFA on an artificial two-class dataset. Specifically, the 200 points (*i.e.*, local descriptors) in the dataset are first randomly grouped into 10 samples, and then we can obtain a global representation for each sample⁵. It can be observed that the exclusivity property of selection matrix enhances the representativeness of prototypes, thus promoting the discrimination of aggregated representations. Although the convergence and complexity of ProLFA have been theoretically provided, it would be more intuitive to see its empirical behavior. Thus, we have shown the training speed and time of this synthetic dataset in Figure 4, where $\bar{d} = 2$ and $P = 1$. Here, a roughly linearly increasing timing result is also consistent with that in Complexity Analysis.

6.3. Evaluation by Image Search

We assess the interpretability of aggregated representations by evaluating their search performance on **Oxford5k** [41] and **INRIA Holidays** [42] datasets. The statistics about all the datasets is provided in Table 1. we can find that such two datasets for search task are class balanced. Specially,

⁵The source code can be found at https://github.com/indusky8/demo_ProLFA.

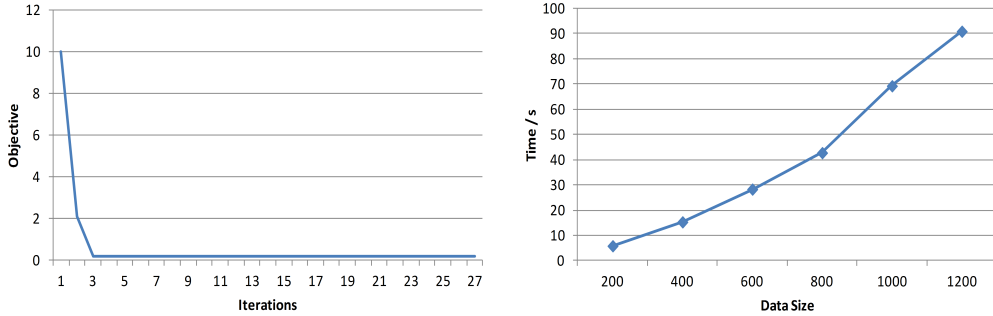


Figure 4: Left: convergence speed. Right: training time of ProLFA.

Table 1: Statistics for four datasets.

Dataset	# images	# classes	Class imbalance?	Size of images	# SIFT/image	# DAISY/image	d
Oxford5k	5,062	11	No	1024×768	90	300	2048
INRIA Holidays	1,491	500	No	1024×768	90	300	2048
Fifteen Scene Categories	4,485	15	Yes	300×250	90	150	4096
Pascal VOC 2007	2,989	20	Yes	300×300	90	150	4096

the downloaded images are all rescaled to 1024×768 , since the initial size is various, such as 1024×759 and 581×1024 . Specifically, Oxford5k consists of 5062 images of buildings and 55 query images corresponding to 11 distinct buildings in Oxford. The search quality is measured by the mean average precision (mAP) computed over the 55 queries. mAP is also the widespread use in evaluating image search system [42] since it evaluates the average performance on all classes. Holidays includes 1491 photos of different locations and objects, 500 of them being used as queries. The search quality is measured by mAP, with the query removed from the ranked list. Image search task is just used to evaluate the performance of the semi-supervised feature aggregation model in (23). Thus we annotated a small subset of Oxford5K (resp. Holidays), but not picking from another one. Such setup is also common in the scenario of semi-supervised image retrieval [43]. Consequently, we randomly select 20% of dataset in each class as the annotated samples. For each image in Oxford5K and Holidays, we take 90 and 300 descriptors as the input of each aggregation approach, respectively. Then our Semi-ProLFA model in (23) is employed to train the aggregation function, thus obtaining the global representations of dataset via (24). Table 2 presents the compared search results produced in two kinds of descriptors, where the prototype size is 2048. As expected, our Semi-ProLFA model performs better com-

Table 2: Impact of our method on search performance. The methods in I, II and III aim at aggregating local descriptors in unsupervised, fully supervised and semi-supervised scenarios, respectively.

Method ↓	mAP			
	Oxford5k		Holidays	
	SIFT	DAISY	SIFT	DAISY
I	BoW [3]	51.35±0.34	56.68±0.12	56.24±0.45
	VLAD [10]	58.31±0.40	59.33±0.18	56.34±0.62
	FV [4]	59.07±0.33	56.31±0.19	60.34±0.25
	SC [28]	63.69±0.55	61.12±0.21	59.31±0.44
	$\phi_\Delta + \psi_d + \text{RN}$ [39]	61.10±0.38	67.14±0.37	73.39±0.62
	DM [40]	59.66±0.18	55.64±0.14	59.69±0.32
II	$T_1(\cdot)$ [16]	62.64±0.26	60.33±0.37	66.01±0.18
	UniVCG [17]	53.11±0.58	51.03±0.56	54.86±0.29
	LC-KSVD2 [24]	56.82±0.43	56.11±0.36	62.81±0.10
	DBoWs [18]	53.30±0.45	55.90±0.50	59.31±0.58
	EO-BoW [19]	62.32±0.12	62.88±0.47	61.21±0.68
	ProLFA	63.40±0.19	61.22±0.34	66.89±0.66
III	Semi-ProLFA	69.90±0.63	70.70±0.30	75.33±0.28

pared with many representative unsupervised and fully supervised feature aggregation approaches. This is because in a semi-supervised aggregation way, representative prototypes can be selected with properties of diversity and discrimination. Thus, the aggregated representations are provided with more interpretability in search task. Additionally, our Semi-ProLFA model achieves more promising results compared with ProLFA, since the proportion of annotated samples is very small. Thus, semi-supervised approaches can take more advantages in this case.

In essence, $\phi_\Delta + \psi_d + \text{RN}$ [39] only obtains 1.23% improvement over the strongest competitor (*i.e.*, our Semi-ProLFA) on DAISY descriptor of Holidays dataset, while consistently performs worse on other cases. This is mainly due to the different data. Specifically, on such a small size dataset (Holidays), the advantage of semi-supervision is limited. In addition, we provide 300 DAISY descriptors for each photo, but 90 for another dataset. Consequently, $\phi_\Delta + \psi_d + \text{RN}$ [39], as an unsupervised method, can leverage more local information though no labels, thus performing best on this dataset even than ours.

6.4. Evaluation by Image Classification

Table 3: Impact of our method on classification task. The methods in I, II and III aim at aggregating local descriptors in unsupervised, fully supervised and semi-supervised scenarios, respectively.

Method ↓	Accuracy			
	Fifteen Scene		Pascal VOC 2007	
	SIFT	DAISY	SIFT	DAISY
I	BoW [3]	66.99 ±0.56	64.88±0.52	43.90±0.44
	VLAD [10]	71.93 ±0.50	69.41±0.36	46.69±0.21
	FV [4]	70.12 ±0.21	70.73±0.17	48.36±0.14
	SC [28]	72.20 ±0.15	74.69±0.21	49.08±0.14
	$\phi_\Delta + \psi_d + \text{RN}$ [39]	77.41±0.43	76.30±0.78	49.69±0.68
	DM [40]	69.36±0.57	71.21±0.56	41.63±0.23
II	$T_1(\cdot)$ [16]	73.74±0.21	74.86±0.11	46.26±0.43
	UniVCG [17]	72.94±0.20	68.07±0.50	47.41±0.21
	LC-KSVD2 [24]	78.35 ±0.19	75.17±0.42	47.83 ±0.33
	DBoWs [18]	73.49±0.17	72.10±0.24	47.29±0.59
	EO-BoW [19]	78.43±0.43	75.60±0.42	49.30±0.14
	ProLFA	80.88 ±0.39	80.33±0.21	52.50±0.48
III	Semi-ProLFA	76.10 ± 0.19	77.81 ±0.32	49.76±0.42

We assess the discrimination of aggregated representations by evaluating their classification performance on **Fifteen Scene Categories dataset** [44] and **Pascal VOC 2007** [45]. As shown in Table 1, the two datasets used for classification task are slightly imbalanced. Concretely, for Fifteen Scene Categories dataset, each category has 200 to 400 images, and average image size is 300×250 pixels. While for Pascal VOC 2007, each category has 96 to 600 images with the general size of 500×375 or 375×500 , and we rescale all images to 300×300 . Specifically, Scene dataset consists of both natural and man-made scenes, and has 4485 images in total. Pascal VOC 2007 is a widely used dataset for image classification. Here, a subset of its training and validation data that contains 2989 images from 20 object categories with one label is used as in [45]. Then, the evaluation protocol in [25] is used: 80% of dataset is sampled from each class to build the training set, and the rest is used for testing. For each image in Scene and Pascal, we take 90 and 150 descriptors as the input of each aggregation approach, respectively. Our ProLFA model in (9) is then employed to train the aggregation function on the training set, thus obtaining the global representations of testing set via (22). Finally, to evaluate the discrimination of aggregated representations, we choose the 1-Nearest Neighbor (1-NN) classifier with Euclidean

distance since it is parameter free and the results will be easily reproducible. Naturally, the classification accuracy by 1-NN classifier is also used to evaluate all compared methods for a fair comparison.

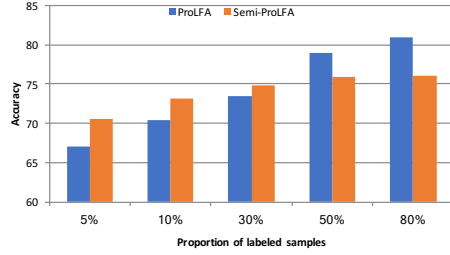
Table 3 presents the compared classification accuracy results produced in two kinds of descriptors, where the prototype size is 4096. As can be seen, our ProLFA outperforms many state-of-the-art unsupervised and fully supervised feature aggregation approaches. Additionally, our extension Semi-ProLFA can achieve comparable performance with that in fully supervised scenario. This is due to the fact that the proposed method improves the discrimination of aggregated representation by task-oriented prototype selection and domain-invariant projection. However, for this task, ProLFA still consistently outperforms Semi-ProLFA, where the improvements range from 2.52% to 4.78%. This is just because of the larger proportion (80%) of labeled data for each dataset. In this scenario, ProLFA, as a fully supervised method, takes more advantage than a semi-supervised method (Semi-ProLFA).

Next, we compare the performances of our ProLFA and Semi-ProLFA with respect to different proportions of labeled samples in Fifteen Scene Categories dataset. As shown in Figure 5, Semi-ProLFA performs better in the case of a few labeled samples, which means that it is suitable to perform semi-supervised local feature aggregation. While on sufficient labeled samples, ProLFA can perform better. In addition, as summarized in Table 1, the four datasets have covered all the possibilities, including class balance, slight and severe class imbalance. However, as reported in Table 2 and Table 3, our method (Semi-ProLFA and ProLFA) can outperform other competitors in most cases. This just show that the class imbalance may have no obvious impact on the advantage of our method.

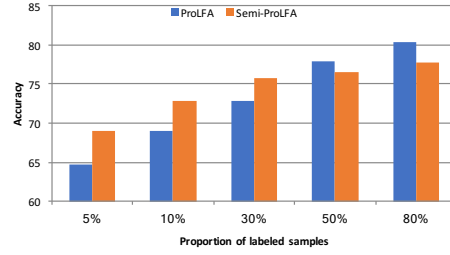
To present the efficiency of each method, we additionally report physical running time to generate global representations of Fifteen Scene Categories dataset, where SIFT and DAISY descriptors serve as input of each method. All the times we report are estimated using a single CPU of a 2.3 GHz Xeon machine with 32 GB of RAM. As shown in Table 4, we can find that the complexity of our method is a general case in feature aggregation, although is not the lowest. Besides, supervised methods task more time than unsupervised ones generally, since they need to additionally train a model on labeled data. In particular, the added overhead can be neglected in practice to the significant accuracy improvements achieved by our ProLFA.

Table 4: Physical running time (*sec.*) to generate global representations of Fifteen Scene Categories dataset by various methods.

Method	Unsupervised						
	BoW [3]	VLAD [10]	FV [4]	SC [28]	$\phi_\Delta + \psi_d + \text{RN}$ [39]	DM [40]	
SIFT	208.9	314.4	384.1	293.1	314.5	301.3	
DAISY	194.8	319.2	358.2	285.7	322.9	333.9	
Method	Supervised						Semi-supervised
	$T_1 (\cdot)$ [16]	UniVCG [17]	LC-KSVD2 [24]	DBoWs [18]	EO-BoW [19]	ProLFA	
SIFT	834.1	1042.2	688.2	1233.4	941.7	1023.3	1001.1
DAISY	794.1	1032.2	633.5	1183.4	978.6	984.5	1030.3

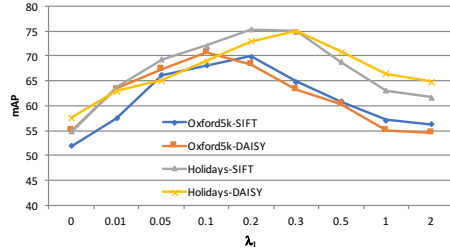


(a)

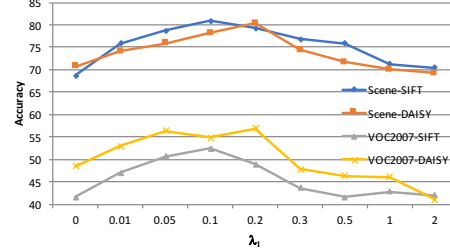


(b)

Figure 5: Impact of labeled proportion in Scene dataset by Semi-ProLFA and ProLFA. (a) on SIFT. (b) on DAISY.



(a)



(b)

Figure 6: Impact of λ_1 on the two tasks. (a) Image search by Semi-ProLFA. (b) Image classification by ProLFA.

6.5. Parameter Analysis

The main parameters in our ProLFA model are prototype size \bar{d} , and the regularization parameter λ_1 (λ_2 is relatively less important). The analysis of these parameters is shown in Figure 6 and Figure 7 for search and classification tasks. The conclusions drawn are identical on both tasks. For all datasets, the performance is an increasing function of the prototype size, but there indeed exists a turning point, which is around $N/6.5$ in scene category

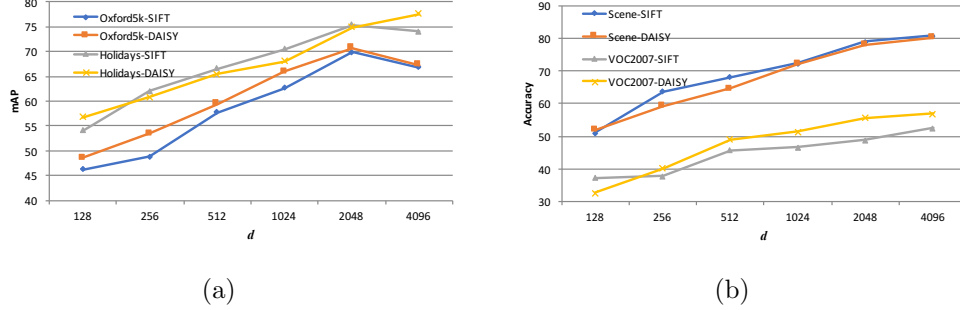
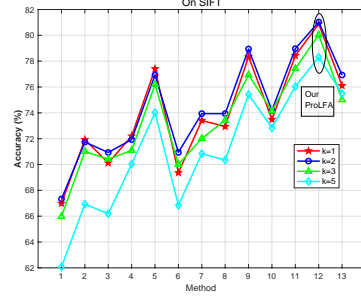


Figure 7: Impact of \bar{d} on the two tasks. (a) Image search by Semi-ProLFA. (b) Image classification by ProLFA.

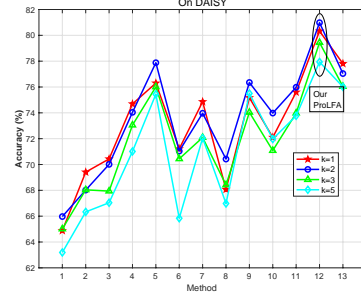
rization task. The general trend for other tasks needs to be further studied. In addition, as λ_1 grows, the mAP drops after growing. This is because firstly the prototypes are selected more discriminatively, and secondly less and less effort is put on fitting data.

It is worth noting that there only exist 6 times of execution for each method, and thus a Wilcoxon test is also necessary to claim our competitiveness against the methods compared. Concretely, at the default 5% significance level, we conduct a paired, two-sided Wilcoxon signed rank test on the six pairs of results about our approach and each compared aggregation approach using the exact method. For search task, we perform Wilcoxon test on our Semi-ProLFA and each competitor, while on ProLFA and each competitor for classification task. The statistical results in terms of p -value of the test on the results of two tasks are almost 0.0312, less than 0.05. Thus, there is enough statistical evidence to conclude that our method is indeed more competitive against all compared methods.

Furthermore, we conduct two additional studies about NN classifier on Fifteen Scene Categories dataset: (i) the effect of using different ‘ k ’ values (including $k = \{1, 2, 3, 5\}$) by using Euclidean distance in NN classifier; (ii) the effect of using different distances (including Euclidean, Cosine, Mahalanobis, and Minkowski Distances) in 1-NN classifier. The results are reported in Figure 8 and Figure 9, respectively. As expected, our ProLFA can achieve the best classification accuracy with different neighbours values and distance metrics. However, ‘ k ’ and distance metric have a slight effect on our ProLFA. For example, it can be found from Figure 8 and Figure 9 that $k > 3$ and Minkowski distance would degrade the classification performance. But it is worth noting that the superiority of aggregated representation by our ProLFA is not influenced by these setups.

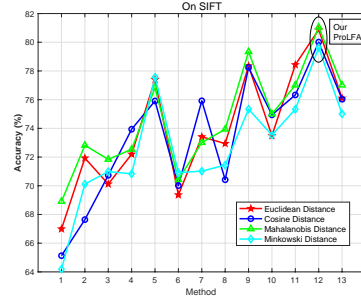


(a) On SIFT descriptors

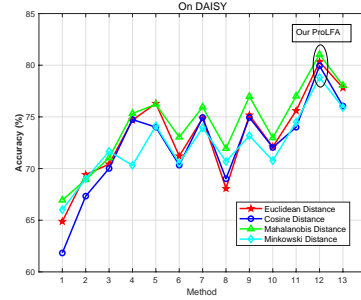


(b) On DAISY descriptors

Figure 8: Impact of ‘k’ (neighbours) values on classification with Fifteen Scene dataset, where the numbers 1-13 in horizontal axis denote BoW [3], VLAD [10], FV [4], SC [28], $\phi_{\Delta} + \psi_d + \text{RN}$ [39], DM [40], $T_1(\cdot)$ [16], UniVCG [17], LC-KSVD2 [24], DBoWs [18], EO-BoW [19], ProLFA, and Semi-ProLFA methods, respectively.



(a) On SIFT descriptors



(b) On DAISY descriptors

Figure 9: Impact of four distance metrics on classification with Fifteen Scene dataset.

6.6. Scalability Analysis

To show the scalability of the proposed ProLFA to large-scale problems, we further conduct image classification on a larger dataset (Caltech-UCSD Birds) [46], which contains 11,788 images from 200 bird species. Each species has approximately 30 images for training and 30 for testing. Totally, there are 5994 training images and 5794 test images. In light of the good performance of CNN, we aggregate the second-order features using fine-tuned VGG-16 [47] network. That is, we resize input images to 224×224 and aggregate the last convolutional layer features after ReLU activation. Thus, the

Table 5: Comparison with two aggregation approaches on Caltech-UCSD Birds.

Method	FV+NN [30]	γ -democratic [29]	ProLFA
Accuracy	77.2	82.3	80.5

size of local descriptors for each image is $28 \times 28 \times 512$ (*i.e.*, $N_i = 784$ for all i and $d = 512$). We first test the performance of our ProLFA as in Section 6.4, where the prototype size is 4096. Then, we compare our method with a recent approach named γ -**democratic** [29] and a neural network-based approach named **FV+NN** [30]. For γ -democratic [29], we directly adopt the aggregated features from their source code ⁶ with $\gamma = 0.5$, while FV+NN [30] is reproduced by following FV encoding layer with a Multi-Layer Perceptron without data augmentation but with bagging. Finally, 1-NN classifier is used to evaluate the discrimination of aggregated features by each method. Table 5 presents the compared results. It can be observed that our ProLFA achieves a 3.3 % improvement over FV+NN [30], while a 1.8 % deterioration over γ -democratic [29]. This is mainly because unlike γ -democratic [29] that performs end-to-end fine-tuning, both FV+NN [30] and our ProLFA directly adopt the local CNN descriptors extracted in advance. Extraction of local descriptors is also not the focus of this work. However, due to task-specific prototype selection in feature aggregation, our ProLFA takes more advantages over FV+NN [C] that still adopts conventional FV aggregation.

7. Conclusion and Future Work

Local feature aggregation is a fundamental problem for numerous applications. In this work, we have introduced, first, a domain-invariant prototype selection based feature aggregation approach (ProLFA) to produce compact representations with the properties of interpretability and discrimination, and second, a composite Block Coordinate Descend (cBCD) framework to efficiently solve the proposed optimization program. Third, by experiments on different local features and tasks, we showed that ProLFA improves the state of the art on the problem of local feature aggregation, even in semi-supervised scenario. Finally, in light of the good performance of parallel

⁶<http://vis-www.cs.umass.edu/o2dp>

optimization algorithms, ProLFA is provided with a potential scaling ability to very large datasets, which is also included in our ongoing research work.

Acknowledgments

This work was supported in part by the National Key Research and Development of China under Grant 2016YFB0800404, in part by the National Natural Science Foundation of China under Grants 61532005, 61332012, and 61572068, and in part by the Fundamental Research Funds for the Central Universities under Grant 2018JBZ001.

References

References

- [1] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *International Journal of Computer Vision* 60 (2) (2004) 91–110.
- [2] S. Winder, G. Hua, M. Brown, Picking the best daisy, in: *CVPR*, 2009, pp. 178–185.
- [3] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: *ECCV*, 2004, pp. 91–110.
- [4] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: *ECCV*, 2010, pp. 143–156.
- [5] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, *TPAMI* 34 (9) (2012) 1704–1716.
- [6] Y. Huang, Z. Wu, L. Wang, T. Tan, Feature coding in image classification: A comprehensive study, *TPAMI* 36 (3) (2014) 493–506.
- [7] J. A. Hartigan, M. A. Wong, A k-means clustering algorithm, *Applied Statistics* 28 (1) (1979) 100–108.
- [8] X. Zhou, K. Yu, T. Zhang, T. S. Huang, Image classification using super-vector coding of local image descriptors, in: *ECCV*, 2010, pp. 141–154.
- [9] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: *CVPR*, 2010, pp. 3360–3367.
- [10] H. Jégou, M. Douze, C. Schmid, P. Pérez, Aggregating local descriptors into a compact image representation, in: *CVPR*, 2010, pp. 3304–3311.
- [11] D. Picard, P. H. Gosselin, Improving image similarity with vectors of locally aggregated tensors, in: *ICIP*, 2011, pp. 669–672.
- [12] J. Krapac, J. Verbeek, F. Jurie, Modeling spatial layout with fisher vectors for image categorization, in: *ICCV*, 2011, pp. 1487–1494.

- [13] F. Jurie, B. Triggs, Creating efficient codebooks for visual recognition, in: ICCV, 2005, pp. 604–610.
- [14] M. Jiu, C. Wolf, C. Garcia, A. Baskurt, Supervised learning and codebook optimization for bag-of-words models, *Cognitive Computation* 4 (4) (2012) 409–419.
- [15] T. Furuya, R. Ohbuchi, Aggregating sparse binarized local features by summing for efficient 3d model retrieval, in: BigMM, 2016, pp. 314–321.
- [16] A. Katharopoulos, D. Paschalidou, C. Diou, A. Delopoulos, Learning local feature aggregation functions with backpropagation, in: EUSIPCO, 2017, pp. 748–752.
- [17] L. Yang, R. Jin, R. Sukthankar, F. Jurie, Unifying discriminative visual codebook generation with classifier training for object category recognition, in: CVPR, 2008, pp. 1–8.
- [18] A. Iosifidis, A. Tefas, I. Pitas, Discriminant bag of words based representation for human action recognition, *Pattern Recognition Letters* 49 (2014) 185–192.
- [19] N. Passalis, A. Tefas, Entropy optimized feature-based bag-of-words representation for information retrieval, *TKDE* 28 (7) (2016) 1664–1677.
- [20] N. Passalis, A. Tefas, Learning bag-of-embedded-words representations for textual information retrieval, *PR* 81 (2018) 254–267.
- [21] P. Wang, L. Liu, C. Shen, Z. Huang, A. van den Hengel, H. Shen, Multi-attention network for one shot learning, in: CVPR, 2017, pp. 22–25.
- [22] E. Elhamifar, G. Sapiro, S. S. Sastry, Dissimilarity-based sparse subset selection, *TPAMI* 38 (11) (2016) 2182–2197.
- [23] Z. Jiang, Z. Lin, L. S. Davis, Learning a discriminative dictionary for sparse coding via label consistent k-svd, in: CVPR, 2011, pp. 1697–1704.
- [24] Z. Jiang, Z. Lin, L. S. Davis, Label consistent k-svd: Learning a discriminative dictionary for recognition, *TPAMI* 35 (11) (2013) 2651–2664.
- [25] E. Elhamifar, C. D. P. M. Kaluza, Subset selection and summarization in sequential data, in: NIPS, 2017, pp. 1036–1045.

- [26] L. Kaufman, P. Rousseeuw, Clustering by means of medoids, in: Proc. Y. Dodge (Ed.) Statistical Data Anal. Based on the ℓ_1 -norm and Related Methods, 1987, pp. 405–416.
- [27] B. Frey, D. Dueck, Clustering by passing messages between data points, *Science* 315 (5814) (2007) 972–976.
- [28] T. Ge, Q. Ke, J. Sun, Sparse-coded features for image retrieval, in: BMVC, 2013.
- [29] T.-Y. Lin, S. Maji, P. Koniusz, Second-order democratic aggregation, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 620–636.
- [30] F. Perronnin, D. Larlus, Fisher vectors meet neural networks: A hybrid classification architecture, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3743–3752.
- [31] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, J. Sivic, Netvlad: Cnn architecture for weakly supervised place recognition, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 5297–5307.
- [32] S. C. AP, S. Lauly, H. Larochelle, M. Khapra, B. Ravindran, V. C. Raykar, A. Saha, An autoencoder approach to learning bilingual word representations, in: Advances in Neural Information Processing Systems, 2014, pp. 1853–1861.
- [33] P. Baldi, Autoencoders, unsupervised learning, and deep architectures, in: Proceedings of ICML workshop on unsupervised and transfer learning, 2012, pp. 37–49.
- [34] J. Mairal, J. Ponce, G. Sapiro, A. Zisserman, F. R. Bach, Supervised dictionary learning, in: Advances in neural information processing systems, 2009, pp. 1033–1040.
- [35] D. Kong, R. Fujimaki, J. Liu, F. Nie, C. Ding, Exclusive feature learning on arbitrary structures via $\ell_{1,2}$ -norm, in: NIPS, 2014, pp. 1655–1663.
- [36] J. Duchi, S. Shalev-Shwartz, Y. Singer, T. Chandra, Efficient projections onto the ℓ_1 ball for learning in high dimensions, in: ICML, 2008, pp. 272–279.

- [37] R. H. Bartels, G. W. Stewart, Solution of the matrix equation $ax + xb = c$ [f4], *Communications of the ACM* 15 (9) (1972) 820–826.
- [38] X. Guo, Z. Lin, Low-rank matrix recovery via robust outlier estimation, *TIP* 27 (11) (2018) 5316–5327.
- [39] H. Jégou, A. Zisserman, Triangulation embedding and democratic aggregation for image search, in: *CVPR*, 2014, pp. 3310–3317.
- [40] T. Furuya, R. Ohbuchi, Diffusion-on-manifold aggregation of local features for shape-based 3d model retrieval, in: *ICMR*, 2015, pp. 171–178.
- [41] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: *CVPR*, 2007, pp. 1–8.
- [42] H. Jégou, M. Douze, C. Schmid, Improving bag-of-features for large scale image search, *International journal of computer vision* 87 (3) (2010) 316–336.
- [43] J. Wang, S. Kumar, S.-F. Chang, Semi-supervised hashing for scalable image retrieval, in: *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on, 2010, pp. 3424–3431.
- [44] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, in: *CVPR*, 2006, pp. 2169–2178.
- [45] X. Li, Y. Guo, Adaptive active learning for image classification, in: *CVPR*, 2013, pp. 859–866.
- [46] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, *Caltech-ucsd birds* 200.
- [47] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: *ICLR*, 2015.