Pattern Recognition 124 (2022) 108397

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Infinite-dimensional feature aggregation via a factorized bilinear model

Jindou Dai, Yuwei Wu*, Zhi Gao, Yunde Jia

Laboratory of Intelligent Information Technology, School of Computer Science, Beijing Institute of Technology (BIT), Beijing, 100081, China

ARTICLE INFO

Article history:
Received 17 August 2020
Revised 8 July 2021
Accepted 19 October 2021
Available online 20 November 2021

Keywords:
Feature aggregation
Infinite-dimensional features
Non-approximate method
Second-order statistics

ABSTRACT

Aggregating infinite-dimensional features has demonstrated superiority compared with their finite-dimensional counterparts. However, most existing methods approximate infinite-dimensional features with finite-dimensional representations, which inevitably results in approximation error and inferior performance. In this paper, we propose a non-approximate aggregation method that directly aggregates infinite-dimensional features rather than relying on approximation strategies. Specifically, since infinite-dimensional features are infeasible to store, represent and compute explicitly, we introduce a factorized bilinear model to capture pairwise second-order statistics of infinite-dimensional features as a global descriptor. It enables the resulting aggregation formulation to only involve the inner product in an infinite-dimensional space. The factorized bilinear model is calculated by a Sigmoid kernel to generate informative features containing infinite order statistics. Experiments on four visual tasks including the fine-grained, indoor scene, texture, and material classification, demonstrate that our method consistently achieves the state-of-the-art performance.

1. Introduction

Feature aggregation, aiming to aggregate features into a global descriptor, is one of the important topics in the machine learning and computer vision communities [1–4]. Many efforts have been made to develop powerful feature aggregation methods and achieve certain success, such as capturing second-order statistics [5–7], and modeling non-linear relationships [8,9]. These methods mainly focus on aggregating *finite-dimensional features* that are easy to calculate but have limited expressive power.

Recent works [6,8,10] have demonstrated that *infinite-dimensional features* can carry richer information than finite-dimensional features, and descriptors from infinite-dimensional feature aggregation are more discriminative than their finite-dimensional counterparts. From the perspective of kernel learning, it comes from the fact that if the inputs are mapped into a high (possibly infinite) dimensional space, non-linear structure underlying the inputs can be captured by operating on the mapped representations. It is proved that non-linear information is essential for a generic descriptor [9,11,12]. From the perspective of information theory, the dimensionality of infinite-dimensional representations is much higher than that of finite-dimensional representations, that is beneficial to encode more information [6,13]. Based on above observations, many attempts have been made to enhance the discriminative power of aggregated descriptors by mapping features to an infinite-dimensional reproducing kernel Hilbert space (RKHS).

Infinite-dimensional features are usually obtained by kernel mapping. A typical kernel learning setting depends on computing Gram matrix that has a quadratic complexity of the sample number, making it intractable for large-scale data [13,14]. To tackle this problem, several works [6,10,15] exploit approximation strategies on infinite-dimensional features. Instead of using the infinite-dimensional mapping, they construct a finite-dimensional mapping and adopt the resulting finite-dimensional representations to approximate infinite-dimensional features. In this way, the approximated representations can share the advantages of infinite-dimensional features without calculating the Gram matrix.

However, approximation-based methods inevitably suffer from approximation error caused by the difference between the exact infinite-dimensional features and the approximated finite-dimensional representations. This deteriorates the performance of aggregated descriptors [16,17]. In addition, approximation-based methods are often at the cost of high-dimensional descriptors. For example, in the case of image classification, Wang et al. [6] approximated infinite-dimensional features with three times the dimensionality of the input features (*e.g.*, 512). It leads to a $(512 \times 3)^2 \approx 2.4 \times 10^6$ dimensional approximated infinite-dimensional global

* Corresponding author.
E-mail address: wuyuwei@bit.edu.cn (Y. Wu).

https://doi.org/10.1016/j.patcog.2021.108397
0031-3203/

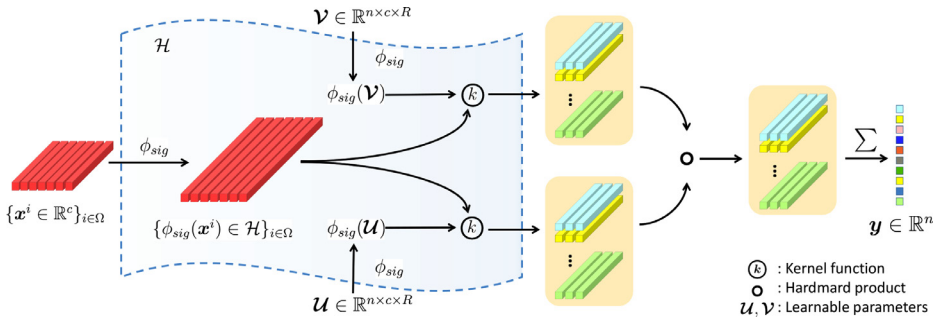


Fig. 1. Illustration of our non-approximate infinite-dimensional feature aggregation method. The original features $\{\mathbf{x}^i \in \mathbb{R}^c\}_{i\in\Omega}$ are first mapped into an $\mathcal{H}$ via the mapping function $\phi_{sig}(\cdot)$ induced by the Sigmoid kernel to construct infinite-dimensional features $\{\phi_{sig}(\mathbf{x}^i) \in \mathcal{H})\}_{i\in\Omega}$. We introduce a factorized bilinear model to capture the second-order statistics of infinite-dimensional features. The learnable parameters $\mathcal{U}$ and $\mathcal{V}$ are also mapped by $\phi_{sig}(\cdot)$ into the same $\mathcal{H}$ so that we can aggregate infinite-dimensional without approximation error via a kernel function and tackle the issue that the infinite-dimensional mapping $\phi_{sig}(\cdot)$ cannot be calculated explicitly.

descriptor, which is a heavy burden for subsequent processing. Therefore, infinite-dimensional feature aggregation without approximation error while generating a compact descriptor remains a challenging problem.

In this paper, we propose a non-approximate infinite-dimensional feature aggregation method that directly aggregates infinite-dimensional features to generate a discriminative and compact global descriptor. Specifically, we introduce a bilinear model to capture pairwise second-order statistics of infinite-dimensional features as the global descriptor. It is non-trivial to directly obtain the global descriptor since infinite-dimensional features are infeasible to store, represent and compute explicitly. To this end, we factorize the parameter of the bilinear model so that the proposed aggregation scheme only involves the inner product between pairs of infinite-dimensional representations, where the inner product is readily calculated by the Sigmoid kernel function. The mapped infinite-dimensional features induced by the Sigmoid kernel are informative due to the infinite order information of inputs. The illustration of our method is presented in Fig. 1. Different from methods in the typical kernel learning setting [13,14], our method can be trained in an end-to-end fashion without calculating the Gram matrix of the whole data. Compared with approximation-based methods, our method directly aggregates infinite-dimensional features instead of resorting to finite-dimensional approximations thus avoids approximation error. In addition, the factorized aggregation strategy enables our method to generate a compact descriptor with a large reduction of parameters.

The contributions of this work are two-fold.

- We propose a non-approximate aggregation method that directly aggregates infinite-dimensional features. Our method can capture the second-order statistics of infinite-dimensional features to generate a both discriminative and compact descriptor.
- The mapping function induced by the Sigmoid kernel can generate informative features as it contains the infinite order statistics of inputs.

2. Related work

2.1. Infinite-dimensional feature aggregation

Infinite-dimensional features have demonstrated superior performance compared with finite-dimensional features [13,14,18,19], which is inspired by kernel-based learning algorithms (*e.g.*, kernel SVM) possessing the capability to efficiently capture non-linear structure of data. A natural idea for feature aggregation methods is to map finite-dimensional features to an infinite-dimensional RKHS and aggregate infinite-dimensional features into a global descriptor followed by a classifier. In practice, since the exact mapping to RKHS is unknown and global descriptors cannot be computed explicitly, several researchers convert this framework into a formulation that relies on the metric between infinite-dimensional global descriptors. Harandi et al. [13] derived several Bregman divergences to compare the infinite-dimensional descriptors in RKHS, and Quang et al. [14] introduced a Log-Hilbert-Schmidt metric on a Hilbert space, which is a generalization of the Log-Euclidean metric. However, the two methods in [13] and [14] depend on Gram matrices having a quadratic complexity of the sample number, in which the computational cost and memory requirements may become intractable and infeasible with larger and larger datasets. Besides, they can only utilize low-dimensional hand-crafted features as it is computationally prohibitive for them to combine the informative convolutional neural network features.

To solve the problem, several works construct approximated finite-dimensional mapping to approximate infinite-dimensional mapping via kernel approximation techniques such as random Fourier transformation and Nyström method. Recently, Wang et al. [6] exploited two approximated additive kernel functions, explicit mappings of the Helinger's kernel and the χ^2 kernel, to approximate infinite-dimensional Gaussian descriptors. Cui et al. [8] introduced a Taylor series kernel to approximate infinite-dimensional mapping corresponding to the Gaussian kernel, and Cavazza et al. [10] approximated the radial basis function (RBF) kernel with the Kronecker products. The methods mentioned above avoid explicit representations of infinite-dimensional features via finite-dimensional feature approximations and share the advantages of infinite-dimensional features. However, these methods inevitably result in approximation error that deteriorates the performance of the aggregated descriptors [16,17]. Besides, the aggregated descriptors are often high-dimensional with expensive storage and computational cost. Compared with the approximation-based methods, our method directly aggregates infinite-dimensional features in the RKHS, thus avoiding the approximation error. In addition, our factorization scheme is able to generate a compact global descriptor with a small number of parameters. The comparisons of infinite-dimensional feature aggregation methods are presented in Table 1.

Table 1
Comparisons with infinite-dimensional descriptors.

Methods	Kernels	Gram matrix	Kernel approximation
Log-HS [14]	RBF	Yes	No
Harandi et al. [13]	RBF	Yes	No
KP [8]	RBF	No	Yes
RAID-G [6]	Hellinger's & χ^2	No	Yes
Ours	Sigmoid	No	No

2.2. Second-order statistic model

Statistics-based methods, such as bag-of-words, vector of locally aggregated descriptors and its variants [2,20], once played an important role in pattern recognition community. Second-order statistics models that have powerful representation ability recently widely used in numerous vision tasks such as image classification [5], visual questioning answering [21,22], video action recognition [23], *etc*.. Covariance-based models [24,25] and bilinear models [5,21] are two successful examples that capture interactions of all pairs of inputs. Covariance-based models represent a set of features (*e.g.*, local features extracted from an image) as a covariance descriptor. Recently, Wang et al. [26] studied the reason on effectiveness of covariance descriptors from the perspective of optimization and concluded that it makes the optimization landscape more smooth and the gradients more predictive. Covariance descriptors properly regularized are symmetric positive definite (SPD) matrices that form a Riemannian manifold. Gao et al. [27] aggregated local features into an SPD matrix to obtain a powerful descriptor. In this case, the Euclidean metric is not applicable and a Riemannian metric respecting manifold structure is needed. The Log-Euclidean metric is used in DeepO$_2$P [28] that applies a tangent space mapping on covariance matrix via the matrix logarithm operator.

The bilinear model is a function of two variables, which is independent linear for both variables. Lin et al. [5] first introduced a bilinear pooling into deep networks, and the output of the model is a weighted outer product of convolutional features. Existing bilinear models mainly focus on three directions: redundancy reduction [21,29,30], normalization techniques [31], and richer statistics modeling [8,12]. Separately speaking, Li et al. [30] utilized matrix decomposition such that high-dimensional parameter matrices in bilinear models are factorized into low-rank matrices. Liu et al. [32] utilized an factorized bilinear model to aggregate audio and face features for speaker naming. Gao et al. [29] proposed compact bilinear pooling (CBP) to reduce the feature dimension two orders of magnitude compared to the original bilinear descriptor without accuracy decrease. Li et al. [33] used CBP to model the interaction between skeleton and RGB information for action recognition. To capture more informative features, Cai et al. [12] presented a framework that integrates higher-order statistics of hierarchical convolutional layers. Generally, combining higher-order statistics is accompanied by high-dimensional representations and a heavy computational burden. Different from the methods that explicitly integrate higher-order statistics of inputs [8,12], we study the problem of how to rich the information underlying the original features in an infinite-dimensional RKHS. We construct informative infinite-dimensional features that containing infinite order statistics of inputs and aggregate them via a factorized bilinear model. Meanwhile, the factorized scheme enables our method to involve less computational and storage costs.

3. Infinite-dimensional feature aggregation

Feature aggregation refers to aggregate a set of features into a global descriptor. High-quality descriptors should be both discriminative (*i.e.*, large inter-class similarities) and compact (*i.e.*, small intra-class similarities). In this section, we focus on how to aggregate infinite-dimensional features that contain richer information than finite-dimensional features.

3.1. Formulation

In order to aggregate input features into a discriminative and compact global descriptor, we define the infinite-dimensional feature aggregation as

$$\mathbf{y} = D_{\mathcal{W}}(\{\phi(\mathbf{x}^i)\}_{i\in\Omega}), \tag{1}$$

where $\mathbf{y}$ is the resulting global descriptor. $\{\mathbf{x}^i \in \mathbb{R}^c\}_{i\in\Omega}$ are a set of finite-dimensional features where Ω denotes the feature set, and i denotes the index in Ω. $\phi(\cdot)$ is an infinite-dimensional mapping function that maps $\mathbf{x}^i$ into an infinite-dimensional RKHS. $D_{\mathcal{W}}(\cdot)$ is an aggregation function that is used to capture the discriminative information of infinite-dimensional features where $\mathcal{W}$ denotes a learnable parameter.

The behavior of infinite-dimensional mapping in Eq. (1) is motivated by recent advances in infinite-dimensional features [6,8,13,14]. It claims that infinite-dimensional features contain richer information than original finite-dimensional features, making the aggregated descriptor more discriminative.

3.2. Non-approximate infinite-dimensional feature aggregation

We propose to directly aggregate infinite-dimensional features without relying on approximation strategy. To explore discriminative information, we first introduce a bilinear model to capture the pairwise relationships of the infinite-dimensional features. Formally, a global descriptor is computed by

$$\begin{aligned} y_s &= \sum_{i\in\Omega} D_{\mathcal{W}}(\phi(\mathbf{x}^i)) = \phi(\mathbf{x}^i)^\top \mathcal{W}_s \phi(\mathbf{x}^i), \\ \mathbf{y} &= [y_1, y_2, \cdots, y_n], \end{aligned} \tag{2}$$

where $\mathbf{y} \in \mathbb{R}^n$, and $\mathcal{W}_s$ is the sth slice of the parameter $\mathcal{W} \in \mathbb{R}^{n\times\infty\times\infty}$. Note that since $\phi(\mathbf{x}^i)$ is an infinite-dimensional feature in the RKHS, $\mathcal{W}$ is an infinite-dimensional three-order tensor. Nevertheless, a natural issue is that we neither store nor represent such an infinite-dimensional tensor, and thus it is unfeasible to calculate Eq. (2) explicitly. To tackle this issue, we factorize $\mathcal{W}_s$ as

$$\mathcal{W}_s = \widehat{\mathcal{U}}_s \widehat{\mathcal{V}}_s^\top, \tag{3}$$

where $\widehat{\mathcal{U}}_s \in \mathbb{R}^{\infty\times R}$ and $\widehat{\mathcal{V}}_s \in \mathbb{R}^{\infty\times R}$ are two rank-R matrices, but they are still infinite-dimensional. Recall the infinite-dimensional mapping function $\phi(\cdot)$ in Eq. (1), the $\widehat{\mathcal{U}}_s$ and $\widehat{\mathcal{V}}_s$ can be obtained by projecting two finite-dimensional matrices via the feature mapping $\phi(\cdot)$,

$$\begin{cases} \widehat{\mathcal{U}}_s = \phi(\mathcal{U}_s) \\ \widehat{\mathcal{V}}_s = \phi(\mathcal{V}_s) \end{cases}, \tag{4}$$

where $\mathcal{U}_s \in \mathbb{R}^{c\times R}, \mathcal{V}_s \in \mathbb{R}^{c\times R}$. All $\{\mathcal{U}_s\}_{s=1}^n$ and $\{\mathcal{V}_s\}_{s=1}^n$ compose two total parameters $\mathcal{U} = [\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_n] \in \mathbb{R}^{n\times c\times R}$ and $\mathcal{V} = [\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_n] \in \mathbb{R}^{n\times c\times R}$, respectively. The learnable parameters $\mathcal{U}, \mathcal{V}$ are finite-dimensional three-order tensors and thus can be stored and represented explicitly. By plugging Eqs. (3) and (4) into Eq. (2), we have

$$\begin{aligned} y_s &= \sum_{i\in\Omega} \phi(\mathbf{x}^i)^\top \mathcal{W}_s \phi(\mathbf{x}^i) \\ &= \sum_{i\in\Omega} \phi(\mathbf{x}^i)^\top \widehat{\mathcal{U}}_s \widehat{\mathcal{V}}_s^\top \phi(\mathbf{x}^i) \\ &= \sum_{i\in\Omega} \phi(\mathbf{x}^i)^\top \phi(\mathcal{U}_s) \phi(\mathcal{V}_s)^\top \phi(\mathbf{x}^i). \end{aligned} \tag{5}$$

Equation (5) cannot be calculated directly due to the existence of infinite-dimensional mapping function $\phi(\cdot)$. One possible solution is to approximate the infinite-dimensional mapping $\phi(\cdot)$ with a finite-dimensional mapping $\phi_{app}(\cdot)$ [6,8]. However, it will introduce the approximation error that could harm the performance of the global descriptor $\mathbf{y}$.

Kernel trick provides us a better solution to calculate the inner product between a pair of infinite-dimensional vectors. It claims

that the inner product between infinite-dimensional $\phi(\mathbf{p})$ and $\phi(\mathbf{q})$ in $\mathcal{H}$ can be calculated by a kernel function

$$k(\mathbf{p},\mathbf{q}) = \langle \phi(\mathbf{p}), \phi(\mathbf{q}) \rangle_{\mathcal{H}}. \tag{6}$$

To calculate Eq. (5) directly, we reformulate it as

$$\begin{aligned} y_s &= \sum_{i\in\Omega} \mathbf{1}^\top \Big(\big(\phi(\mathcal{U}_s)^\top \phi(\mathbf{x}^i)\big) \circ \big(\phi(\mathcal{V}_s)^\top \phi(\mathbf{x}^i)\big) \Big) \\ &= \sum_{i\in\Omega} \sum_{r}^{R} \Big(\big\langle \phi(\mathcal{U}_{s,r}), \phi(\mathbf{x}^i) \big\rangle \big\langle \phi(\mathcal{V}_{s,r}), \phi(\mathbf{x}^i) \big\rangle \Big), \end{aligned} \tag{7}$$

where $\mathbf{1}$ denotes a column vector of ones, $\mathcal{U}_{s,r} \in \mathbb{R}^c$ and $\mathcal{V}_{s,r} \in \mathbb{R}^c$ are the rth column of $\mathcal{U}_s$ and $\mathcal{V}_s$, respectively. $\langle \cdot, \cdot \rangle$ denotes the inner product, and "$\circ$" denotes the Hadamard product. It indicates that the condition for using kernel trick, the existence of the inner product, has been met. Reformulating Eq. (7), we obtain the storable, representable, and computable global descriptor

$$y_s = \sum_{i\in\Omega} \sum_{r}^{R} \big(k(\mathcal{U}_{s,r}, \mathbf{x}^i) k(\mathcal{V}_{s,r}, \mathbf{x}^i) \big). \tag{8}$$

Our factorized aggregation scheme solves the issue where learnable parameters $\mathcal{W}_\mathbf{s}$ cannot be stored or represented, and Eq. (2) cannot be calculated explicitly. More importantly, it enables the infinite-dimensional features to be directly aggregated instead of relying on kernel approximations, thereby avoiding the approximation error.

3.3. Sigmoid kernel

The kernel function $k(\cdot,\cdot)$ in Eq. (8) heavily influences the performance of the final descriptor. It should not only be able to map the original finite-dimensional features to an infinite-dimensional $\mathcal{H}$ implicitly but also have the capability to rich the information of the original features. In our method, we consider the Sigmoid kernel

$$k(\mathbf{p},\mathbf{q}) = \tanh(\alpha \mathbf{p}^\top \mathbf{q} + \beta) \tag{9}$$

where $\mathbf{p}, \mathbf{q}$ are two finite-dimensional vectors (*e.g.*, original features $\mathbf{x}^i$ and $\mathbf{x}^j$). α is a scaling parameter and β is a shifting parameter.

Theorem 1. *The Sigmoid kernel is a valid conditionally positive definite kernel when $\alpha > 0$ and β is small enough.*

For detailed proof, please refer to [34]. From this theorem, as long as we select appropriate hyper-parameters α and β, the RKHS induced by the Sigmoid kernel can be guaranteed to be an infinite-dimensional space.

The final aggregation formulation is Eq. (8) equipped with the Sigmoid kernel Eq. (9). Fig. 1 shows our non-approximate infinite-dimensional feature aggregation method. To further analyze why our method can generate an informative and discriminative descriptor, we have the following proposition.

Proposition 1. *Given features $\{\mathbf{x}^i\}_{i\in\Omega}$, the aggregated descriptor $\mathbf{y}$ using Eq. (8) equipped with Eq. (9) contains infinite-order statistics of $\{\mathbf{x}^i\}_{i\in\Omega}$.*

Proof. Let first derive the infinite-dimensional mapping $\phi_{sig}(\cdot)$ induced by the Sigmoid kernel. As the Sigmoid kernel is a valid conditionally positive definite kernel when β is small enough, we set $\beta = 0$ here. Given two vector $\mathbf{p}$ and $\mathbf{q}$, the output of the Sigmoid kernel is

$$\begin{aligned} k(\mathbf{p},\mathbf{q}) &= \tanh(\alpha \mathbf{p}^\top \mathbf{q}) \\ &\overset{\text{Taylor}}{=} \sum_{m=1}^{+\infty} \frac{2^{2m}(2^{2m}-1)B_{2m}(\alpha \mathbf{p}^\top \mathbf{q})^{2m-1}}{(2m)!} \\ &= \sum_{m=1}^{+\infty} \frac{2^{2m}(2^{2m}-1)B_{2m}(\alpha)^{2m-1}}{(2m)!} (\mathbf{p}^\top)^{2m-1} (\mathbf{q})^{2m-1} \\ &= \sum_{m=1}^{+\infty} \sqrt{\frac{2^{2m}(2^{2m}-1)B_{2m}(\alpha)^{2m-1}}{(2m)!}} (\mathbf{p}^{2m-1})^\top \\ &\quad \times \sqrt{\frac{2^{2m}(2^{2m}-1)B_{2m}(\alpha)^{2m-1}}{(2m)!}} \mathbf{q}^{2m-1} \\ &= \phi_{sig}(\mathbf{p})^\top \phi_{sig}(\mathbf{q}), \end{aligned} \tag{10}$$

Thus, we have

$$\begin{aligned} \phi_{sig}(\mathbf{p}) &= [\eta_1 \mathbf{p}, \eta_2 \mathbf{p}^3, \cdots, \eta_m \mathbf{p}^{2m-1}, \cdots], \\ \eta_m &= \sqrt{\frac{2^{2m}(2^{2m}-1)B_{2m}(\alpha)^{2m-1}}{(2m)!}}, \end{aligned} \tag{11}$$

where B_m is mth Bernoulli number and α is a constant parameter. We can find that $\phi_{sig}(\mathbf{p})$ contains infinite-order statistics $\mathbf{p}^{2m-1}$ of $\mathbf{p}$ with the increase of m. As Eq. (8) is derived from Eq. (5), by plugging Eq. (11) into Eq. (5), we have

$$y_s = \sum_{i\in\Omega} \phi_{sig}(\mathbf{x}^i)^\top \phi_{sig}(\mathcal{U}_s) \phi_{sig}(\mathcal{V}_s)^\top \phi_{sig}(\mathbf{x}^i), \tag{12}$$

where $\phi_{sig}(\cdot)$ is in the form of Eq. (11). $\mathbf{y}$ contains infinite-order statistics of $\{\mathbf{x}^i\}_{i\in\Omega}$ because it is aggregated from $\{\phi_{sig}(\mathbf{x}^i)\}_{i\in\Omega}$ that contains infinite-order statistics $(\mathbf{x}^i)^{2m-1}$ of $\mathbf{x}^i$. □

3.4. Network architecture for image classification

We instantiate a network architecture for image classification, as shown in Fig. 2. The proposed non-approximate infinite-dimensional feature aggregation can be inserted into a deep network for end-to-end training. We use the convolutional features as original features and send them to the proposed method to obtain the global descriptor $\mathbf{y}$. A batch normalization (BN) layer is used to accelerate convergence. A fully-connected (FC) layer followed by a softmax is used for classification.

End-to-end training Since our feature aggregation module is differentiable and the network architecture is a directed acyclic graph, all parameters including $\mathcal{U}$ and $\mathcal{V}$ in Eq. (8) can be learned by the back-propagation algorithm. Let ℓ be cross entropy loss and $d\ell/d\mathbf{y}$ be the gradient of the loss function ℓ w.r.t. global descriptor $\mathbf{y}$, we have

$$\begin{aligned} \frac{d\ell}{d\mathcal{U}_{s,r}} &= \frac{d\ell}{d\mathbf{y}_s}\frac{d\mathbf{y}_s}{d\mathcal{U}_{s,r}} = \frac{d\ell}{d\mathbf{y}_s} \sum_{i\in\Omega} \alpha \tanh(\mathcal{V}_{s,r}^\top \mathbf{x}^i)\big(1 - \tanh^2(\alpha \mathcal{U}_{s,r}^\top \mathbf{x}^i)\big)\mathbf{x}^i, \\ \frac{d\ell}{d\mathcal{V}_{s,r}} &= \frac{d\ell}{d\mathbf{y}_s}\frac{d\mathbf{y}_s}{d\mathcal{V}_{s,r}} = \frac{d\ell}{d\mathbf{y}_s} \sum_{i\in\Omega} \alpha \tanh(\mathcal{U}_{s,r}^\top \mathbf{x}^i)\big(1 - \tanh^2(\alpha \mathcal{V}_{s,r}^\top \mathbf{x}^i)\big)\mathbf{x}^i. \end{aligned} \tag{13}$$

The gradient of other layers, such as BN and convolutional layers, can be computed by chain rule straightforwardly. We use the stochastic gradient descent optimizer to update the parameters.

3.5. Error comparisons

We compare our method with two infinite-dimensional feature aggregation methods KP [8] and RAID-G [6], and then provide theoretical error analyses of the infinite-dimensional features. Both KP and RAID-G adopted the approximation strategy to replace infinite-dimensional features $\phi(\mathbf{x})$ with finite-dimensional

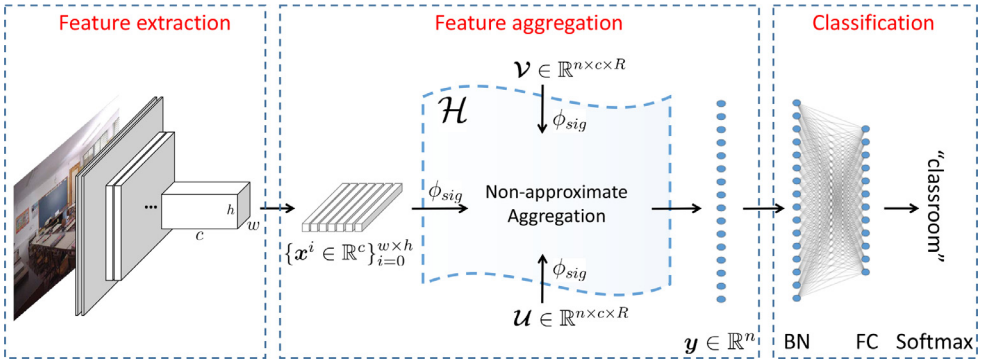


Fig. 2. Network architecture for image classification. It consists of three modules: feature extraction, infinite-dimensional feature aggregation, and classification. $\mathcal{U}$ and $\mathcal{V}$ are learnable parameters.

Table 2
A unified mapping perspective of the previous methods and our method. Given an input $\mathbf{x} \in \mathbb{R}^c$, the difference is how to construct mapping function. KP [8] defines a Tayler kernel to approximate the RBF kernel. RAID-G [6] defines two mapping functions to approximate the Hellinger's and the χ^2 kernels (L is a constant). The mapping function in our method is induced by the Sigmoid kernel, and η is defined in Eq. (11).

Methods	Mapped dim.	Mapping functions
KP	$1+c+\sum_{i=2}^{p} d_i$	$\phi_{app}: \mathbf{x} \mapsto \left[\lambda_0, \lambda_1(\mathbf{x})^\top, \lambda_2\left(TS(\mathbf{x}^{(2)})\right)^\top, \cdots, \lambda_p\left(TS(\mathbf{x}^{(p)})\right)^\top\right]^\top$
RAID-G-Hel	c	$\phi_{app}: \mathbf{x} \mapsto \sqrt{\mathbf{x}}$
RAID-G-Chi	$3c$	$\phi_{app}: \mathbf{x}_i \mapsto \sqrt{\mathbf{x}_i}\left[\sqrt{L}, \sqrt{2Lsech(L\pi)}\cos(Llog(\mathbf{x}_i)), \sqrt{2Lsech(L\pi)}sin(Llog(\mathbf{x}_i))\right]^\top$
Ours	$infinite$	$\phi_{sig}: \mathbf{x} \mapsto [\eta_1\mathbf{x}, \eta_2\mathbf{x}^3, \cdots, \eta_m\mathbf{x}^{2m-1}, \cdots]$

representation $\phi_{app}(\mathbf{x})$, such that for two vectors $\mathbf{x}, \mathbf{x}'$, there is $\hat{k}(\mathbf{x}, \mathbf{x}') = \langle \phi_{app}(\mathbf{x}), \phi_{app}(\mathbf{x}') \rangle \approx \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}')$.

KP approximates the RBF kernel up to a given order by using compact feature maps, namely Taylor series kernel. Given input features $\mathbf{x}$ and $\mathbf{x}'$, it is represented as $k(\mathbf{x}, \mathbf{x}') = \sum_{r=0}^{p} \lambda_r^2 (\mathbf{x}^\top \mathbf{x}')^r$, where p is given order for approximation. The induced mapping function is shown in Table 2, where $TS(\mathbf{x}^{(i)})$ is the approximation of $\otimes_i \mathbf{x}$ using Tensor Sketching [16] to a d_i-dimensional vector. Integrating very high order statistics will bring a heavy computational burden. The relative approximation error of KP, according to Chebyshev's inequality, is bounded as

$$P\left[\left|\hat{k}(\mathbf{x}, \mathbf{x}') - k(\mathbf{x}, \mathbf{x}')\right|\right] \geq \epsilon k(\mathbf{x}, \mathbf{x}') \leq \frac{1}{d_{min}\epsilon^2}\Delta(p), \tag{14}$$

where $d_{\min} = \min(d_2, \cdots, d_p)$ and

$$\Delta p = \begin{cases} 2(p-1) & \text{if } S = \pm 1 \\ \frac{2S^2(S^{2p}-1)}{S^2-1} & \text{otherwise.} \end{cases}, \tag{15}$$

$S = \frac{1}{\cos\theta}$ equals to the reciprocal of the cosine similarity between features $\mathbf{x}$ and $\mathbf{x}'$. The approximation error is not only related to the dimensionality $d_{\min}$ and order p, but heavily depends on the angle between two features. In other words, if $\mathbf{x}$ is orthogonal to $\mathbf{x}'$, S will approach infinity, leading to an infinite upper error bound.

RAID-G introduces approximated mapping functions corresponding to two additive and homogeneous kernels, the Hellinger's kernel and χ^2 kernel. The Hellinger's kernel is of the form $k(\mathbf{x}, \mathbf{x}') = \sum_{b=1} \sqrt{x_b x_b'}$, where x_b is the bth element of $\mathbf{x}$. The χ^2 kernel is of the form $k(\mathbf{x}, \mathbf{x}') = \sum_{b=1} 2(x_b x_b')/(x_b + x_b')$. Although its motivation is to take advantage of infinite-dimensional features to obtain infinite-dimensional descriptors, it adopts an approximation strategy to approximate infinite-dimensional features with infinite-dimensional representations. The normalized approximation error of RAID-G is

$$\epsilon(x_b, x_b') = \frac{\hat{k}(x_b, x_b')}{\sqrt{x_b x_b'}} - \frac{k(x_b, x_b')}{\sqrt{x_b x_b'}} = N\left(log\frac{x_b'}{x_b}\right), \tag{16}$$

where N is a constant and b is the index. Notice that the normalized approximation error only depends on the ratio x_b'/x_b. It may be large when either x_b' is much larger than x_b, or vice-versa.

Different from the above approximation-based methods, there is no approximation in our method because we directly aggregate infinite-dimensional features ($\phi_{sig}(\mathbf{x})$ in our method), rather than approximated finite-dimensional representations $\phi_{app}(\mathbf{x})$. This is achieved via a factorized bilinear model as derived from Eqs. (2) to (8). In particular, we used " = " instead of "≈" from Eqs. (7) to (8) because $\left\langle \phi(\mathcal{U}_{s,r}), \phi(\mathbf{x}^i) \right\rangle$ is strictly equal to $k(\mathcal{U}_{s,r}, \mathbf{x}^i)$. It indicates no approximation error in the aggregation process.

3.6. Computational complexity

The final infinite-dimensional aggregation is realized via Eq. (8) equipped with the Sigmoid kernel Eq. (9), which involves two matrix multiplications, one matrix element-wise multiplication, $\tanh(\cdot)$, and sum-pooling. Given features $\{\mathbf{x}^i \in \mathbb{R}^c\}_{i\in\Omega}$ where $|\Omega| = l$, and aggregation parameters $\mathcal{U} = [\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_n] \in \mathbb{R}^{n\times c\times R}$, and $\mathcal{V} = [\mathcal{V}_1, \mathcal{V}_2, \cdots, \mathcal{V}_n] \in \mathbb{R}^{n\times c\times R}$, the computational complexity of our aggregation formula is $\mathcal{O}(2nlcR + 2nlR + 2nlR) = \mathcal{O}(2nlcR + 4nlR)$. As will be analyzed in Section 4.4 that we adopt $R = 1$ in our implementation, and thus the complexity is reduced to $O(2nlc + 4nl)$.

4. Experiments

To evaluate the performance of the proposed method, we conduct experiments on four image classification tasks: the fine-grained, indoor scene, texture, and material classifications.

4.1. Implementation

Following the state-of-the-art methods [8,35], we utilize the VGG-16 model pretrained on the ImageNet dataset as the backbone if not special specified. Layers after the $conv5-3$ are removed, and our method is applied to aggregate features from the $conv5-3$ layer (see Fig. 2). In detail, we train our networks on all

Table 3
Comparisons with state-of-the-art methods in terms of average accuracy (%).

Methods	Final dim.	Aircraft	Cars	MIT-Indoor	DTD	MINC
VGG-16 [36]	4,096	74.1	79.8	64.5	60.1	73.0
B-CNN [5]	2.6×10^5	84.1	91.3	77.6	67.5	74.5
CBP [29]	8,192	84.1	91.2	76.8	67.7	73.3
LRBP [37]	100	87.3	90.9	73.6	65.8	69.0
HiO [12]	8,704	88.3	91.3	-	-	-
DeepKSPD [9]	1.3×10^5	91.5	93.2	81.0	-	-
MPN-COV [25]	32,896	-	-	76.5	68.0	76.2
iSQRT-COV [25]	2,080	-	-	78.9	70.6	78.6
iSQRT-COV [25]	8,256	-	-	79.2	71.0	78.8
iSQRT-COV [25]	32,896	-	-	79.6	71.2	78.9
SMSO [35]	2,048	-	-	79.5	69.3	78.0
FBC [21]	8,192	-	-	79.9	71.5	80.2
Ours	1,024	**91.6**	**93.5**	**81.6**	**72.6**	**80.9**

datasets using stochastic gradient descent with the batch size of 16, momentum of 0.9, and weight decay of 5×10^{-4}. The learning rate is initialized as 0.001 for layers of the VGG-16 backbone and 0.01 for new layers. We then divide the learning rate by 10 every 10 epochs for the MINC dataset and 40 epochs for the others.

4.2. Comparisons with state-of-the-art methods

We compare our method with state-of-the-art feature aggregation methods on the Aircraft [38], Cars [39], MIT-Indoor [40], DTD [41] and MINC [42] datasets, and compare our method with several state-of-the-art feature aggregation methods. Performance comparisons are shown in Table 3, where "Final dim." denotes the dimensionality of the aggregated global descriptor.

On fine-grained classification datasets, the Aircraft and the Cars, methods based on second-order statistic model, including B-CNN [5], CBP [29] and LRBP [37] obtain significant improvements compared to VGG-16, demonstrating the second-order information does help to improve the discriminative power of the global descriptors. Compared with these methods, HiO [12] maps the original features into a higher feature space and has a better performance. DeepKSPD [9] is a competitive method as it captures nonlinear relationships of features and utilizes matrix square-rooting normalization. However, the methods mentioned above are limited to feature aggregation in a finite-dimensional space. In contrast, our method aggregates features in an infinite-dimensional RKHS, making it possible to capture more information. We consistently achieve the best results, 91.6% on the Aircraft dataset and 93.5% on the Cars dataset and do not need the time-consuming matrix square-rooting normalization. It is mainly because our method aims at the infinite-dimensional feature aggregation, and infinite-dimensional features carry richer and more discriminative information than finite-dimensional features. Besides, our method achieves the best performance using a relatively low-dimensional descriptor, reducing a large number of required parameters in the classifier. These results demonstrate that our method can extract discriminative information into a compact descriptor.

We also conduct experiments on the indoor scene, texture, and material classification tasks, evaluated on the MIT-Indoor, DTD, and MINC datasets. From Table 3, our method consistently achieves better results than other methods. MPN-COV [25] applies an global matrix power normalization to improve the representation and generalization abilities of deep CNNs, and its variant, iSQRT-COV [25], uses Newton-Schulz iteration to avoid EIG or SVD in MPN-COV for a fast training. SMSO [35] is an advanced method that is motivated by a statistical analysis of the distribution of the network's intermediate responses. The recently proposed method, FBC, achieves the state-of-the-art performance, which revisited bilinear models from a coding perspective and proposed a factorized bilinear coding via sparse coding. Our method surpasses it, 1.7% on the MIT-Indoor dataset, 1.1% on the DTD dataset, and 0.7% on the MINC. All those methods consider the second-order statistics of finite-dimensional features. In contrast, our method pays attention to infinite-dimensional features. The second-order statistics of infinite-dimensional features demonstrate more discriminative power than the finite-dimensional counterparts, consistently achieving the highest average precision on the three image classification tasks.

Table 4
Comparisons with infinite-dimensional descriptors. On KTH-T2b, we use 4 data splits suggested by Wang et al. [6] and report mean accuracy and the standard deviation values. On Aircraft and Cars, we use the public data split and report the best performance.

Methods	Final dim.	KTH-T2b	Aircraft	Cars
Harandi et al. [13]	–	80.1 ± 4.6	–	–
Log-HS [14]	–	81.9 ± 3.3	–	–
RAID-G-Hel [6]	1.3×10^5	89.0 ± 5.4	81.0	82.1
RAID-G-Chi [6]	1.2×10^6	89.3 ± 4.5	81.7	85.7
KP [8]	12,801	85.4 ± 3.1	86.9	92.4
Ours	1,024	$\mathbf{90.1 \pm 2.0}$	**91.6**	**93.5**

4.3. Comparisons with infinite-dimensional descriptors

We compare our descriptor with several infinite-dimensional descriptors: Harandi et al. [13], Log-HS [14], RAID-G [6], and KP [8] on the KTH-T2b, Aircraft and Cars datasets as shown in Table 4. On the KTH-T2b dataset, following the setting in [6], we randomly choose three training samples for per subset in each class and the rest are used for testing. We report the mean and the standard deviation value for all the 4 splits of the KTH-T2b dataset. On the Aircraft and the Cars datasets, we use the public data split and report the best performance. For RAID-G, we use single-scale input size and the same backbone, VGG-16, for fair comparison. Considering that the feature of RAID-G-Chi is up to 1.2×10^6 dimensions and both Aircraft and Cars have 10,000 or more images, it is hard to use SVM adopted in RAID-G for classification in our experimental environment (64G RAM). Thus, we reduced the dimension of RAID-G-Chi to 1.3×10^5, which is about the same as that of RAID-G-hell.

Harandi et al. and Log-HS design a metric between infinite-dimensional representations to avoid calculating the infinite-dimensional mapping explicitly. The side effects are that they need to compute the Gram matrix based on all data samples in the dataset, and thus it is hard for them to scale to large datasets. Besides, it will be computationally prohibitive to utilize CNN features [6]. Our method does not rely on the Gram matrix so that it

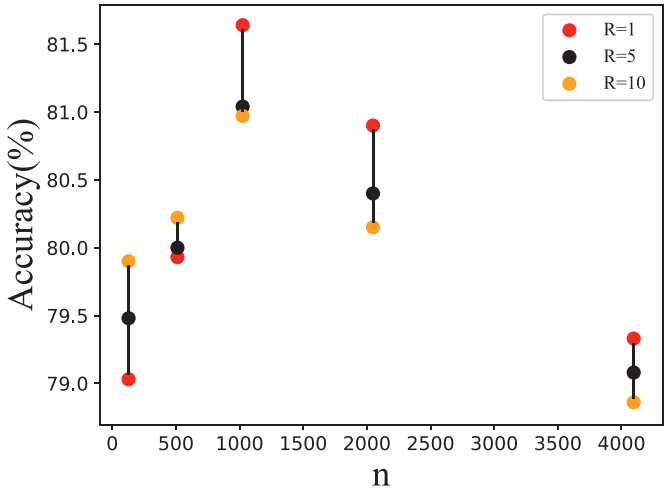


Fig. 3. Effects of rank R and final dimension n on the MIT-Indoor dataset (best viewed in color). When $n < 1000$, yellow dots are always at the top and red dots are always at the bottom, which means a larger R performs better. When $n > 1000$, the opposite is true. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

can scale well to large datasets. RAID-G and our method surpass the above two methods by a large margin. RAID-G-Hel and RAID-G-Chi approximate the Hellinger's and χ^2 kernels via explicit feature maps, respectively. As RAID-G is not an end-to-end method and the local features can not be learned, there is still much room for RAID-G to improve its performance on the Aircraft and Cars datasets. Compared with RAID-G, our method is trained in an end-to-end fashion and achieves better performance in term of average accuracy and standard deviation. We speculate that it is not only due to end-to-end training with a better data fitting ability but mainly benefits from no approximation error to aggregate infinite-dimensional features. KP defines a Taylor series kernel to approximate infinite-dimensional mapping induced by the RBF kernel. Due to the approximation error of KP, its experimental results on these two datasets are worse than ours on these datasets. More accurate approximation comes at the cost of expensive storage and computation, which is usually unaffordable. In contrast, as analyzed in Section 3.3 and Eq. (12), our global descriptor $\mathbf{y}$ does contain infinite order information of the input features and thus is more informative. Our method gains obvious improvements, 4.7% on the Aircraft dataset and 0.9% on the Cars dataset. Experiment results prove the superiority of our method that can avoid the approximation error caused by the approximation strategy and achieve better performance.

4.4. Analyses of hyper-parameters

We here evaluate several important hyper-parameters, including the rank R in Eq. (8), the dimension n in Eq. (2) and the α in the Sigmoid kernel.

Effects of R and n The rank R of $\mathcal{U}_{\mathbf{s}} \in \mathbb{R}^{c\times R}$ and $\mathcal{V}_{\mathbf{s}} \in \mathbb{R}^{c\times R}$, and the dimension n of the final descriptor $\mathbf{y} \in \mathbb{R}^n$ play important roles in our method. We evaluated the effects of n from 128 to 4096 and R from 1 to 10 on the MIT-Indoor dataset. Experimental results are shown in Fig. 3. When the rank R is fixed, as the dimension n increases, the accuracy increases first and then decreases. We found that when n is small, a larger R performs better. In Fig. 3, yellow dots (*i.e.*, $R = 10$) are always at the top, and red dots (*i.e.*, $R = 1$) are always at the bottom when $n < 1000$. The reason may be that a small n causes underfitting, and a large R enhances the fitting ability. When n is a large value, a smaller R performs better. This is reflected in Fig. 3 as red dots (*i.e.*, $R = 1$) are always at the top, and yellow dots (*i.e.*, $R = 10$) are always at the bottom when $n > 1000$. The highest accuracy, 81.6%, is achieved when $R = 1, n = 1024$. We speculate that a high-dimensional descriptor, *e.g.*, $n = 4096$, adversely leads to overfitting, and a small R can alleviate this problem. The experimental results indicate that our method can achieve a good performance with fewer parameters and generate a compact descriptor. We suggest to adopt R as 1 and n as 1024 in all experiments.

Table 5
Effect of α in the Sigmoid kernel.

α	0.01	0.1	0.5	1.0
MINC	**80.9**	**80.9**	80.3	80.4
MIT-Indoor	81.0	**81.6**	81.0	81.0

Table 6
Effect of β in the Sigmoid kernel.

β	0.0	0.001	0.01	0.1	1.0
MINC	**80.9**	80.2	79.5	79.3	79.1
MIT-Indoor	**81.6**	80.0	79.9	79.9	78.8

Effect of α We evaluate the effect of α in the Sigmoid kernel. To this end, we vary α in the range from 0.01 to 1.0. The results on the MINC and MIT-Indoor datasets are presented in Table 5. When $\alpha = 0.1$, the highest accuracy on the MINC and MIT-Indoor datasets are obtained, 80.9% and 81.6% respectively. Experimental results demonstrate that our method is not sensitive to α in the Sigmoid kernel.

Effect of β We evaluate the effect of β in the Sigmoid kernel. We fix α to 0.1 and set β to the range from 0.0 to 1.0. The results on the MINC and MIT-Indoor datasets are presented in Table 6. With the increase of β, the accuracy decreases on the both datasets. The results are consistent with Theorem 1: the Sigmoid kernel is a valid conditionally positive definite kernel on the premise that β is small enough. We set $\beta = 0$ in all experiments.

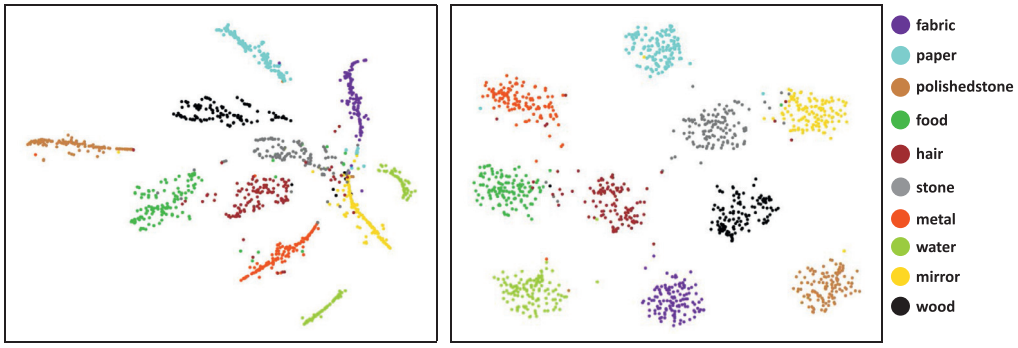
4.5. Comparisons with different kernels

The kernel function in Eq. (8) plays an important role in our method. It determines the induced infinite-dimensional feature space and the quality of the infinite-dimensional features. We evaluate several commonly used kernels, shown in Table 7, on the MINC and MIT-Indoor dataset.

The polynomial kernel is well studied for problems where training data is normalized. It induces a finite-dimensional mapping when the polynomial degree d is small. The RBF kernel is one of the most commonly used kernels, and γ is a constant parameter. The Sigmoid kernel has been studied in Section 3.3. Among these kernels, the RBF kernel and the Sigmoid kernel could induce an infinite-dimensional feature mapping. As indicated in Table 7, the two kernels consistently outperform the polynomial kernel, which further confirms the advantages of infinite-dimensional features over finite-dimensional features. Our method with the Sigmoid kernel achieves the best performance among these kernels.

Table 7
Comparisons with different kernels on the MINC and MIT-Indoor datasets.

Kernels	Formula	MINC	MIT-Indoor
polynomial	$k(\mathbf{p},\mathbf{q}) = (\alpha\mathbf{p}^\top\mathbf{q} + \beta)^d$	79.4	79.3
RBF	$k(\mathbf{p},\mathbf{q}) = \exp(-\gamma\|\|\mathbf{p}-\mathbf{q}\|\|^2)$	79.7	80.5
Sigmoid (Ours)	$k(\mathbf{p},\mathbf{q}) = \tanh(\alpha\mathbf{p}^\top\mathbf{q} + \beta)$	**80.9**	**81.6**



(a) B-CNN on the MINC (b) Ours on the MINC

Fig. 4. Distributions of B-CNN descriptors and our descriptors on the MINC dataset using the t-SNE. Different colors represent different classes (best viewed in color).

Table 8
Influence of the infinite-dimensional mapping induced by the Sigmoid kernel in terms of Average Accuracy (%). "w/o Sigmoid" denotes the kernel function in Eq. (8) is replaced with the inner product.

Methods	Aircraft	Cars	MIT-Indoor
w/o Sigmoid	89.8	92.3	80.6
Ours (Sigmoid)	**91.6**	**93.5**	**81.6**

It outperforms the RBF kernel by 1.2% and 1.1% on the MINC and MIT-Indoor datasets respectively, which indicates the effectiveness of our method with the Sigmoid kernel.

4.6. Ablation study

We further conduct ablation experiments to evaluate the effectiveness of infinite-dimensional mapping induced by the Sigmoid kernel on the Aircraft, Cars, and MIT-Indoor datasets. The results are presented in Table 8. "w/o Sigmoid" means feature aggregation is carried out in a finite-dimensional space without the Sigmoid kernel. After applying the infinite-dimensional feature mapping induced by the Sigmoid kernel, the accuracies on these three datasets all achieve considerable improvements. There are 1.8%, 1.2% and 1.0% improvements on the Aircraft, Cars, and MIT-Indoor datasets, respectively. This demonstrates that aggregating infinite-dimensional features induced by the Sigmoid kernel can effectively improve the discriminative power of the aggregated global descriptor.

4.7. Comparisons of training speed

We compare the training speed of B-CNN, iSQRT-COV, CBP, and infinite-dimensional feature aggregation methods, KP, and our method. We tested how many images can be processed per second on the MINC and MIT - Indoor datasets in the same experimental environment, and the results are presented in Table 9. The speed of our method is similar to that of B-CNN, and is obviously better than that of other methods, which demonstrates that our method has advantages in computational complexity.

Table 9
Comparisons of training speed (images per second).

Methods	MINC	MIT-Indoor
B-CNN [5]	104.5	31.9
CBP [29]	57.7	24.5
iSQRT-COV [25]	94.1	31.0
KP [8]	65.3	26.5
Ours	103.2	31.8

4.8. Visualization

To further demonstrate the superiority of our method over finite-dimensional feature aggregation methods, we visualize distributions of B-CNN [5] descriptors and our descriptors on the MINC dataset using t-SNE in Fig. 4. We observe that the distribution of B-CNN descriptors in Fig. 4(a) is in the shape of an elongated strip and samples of different classes tend to be close to each other, which is not conducive to the determination of the classification hyperplanes. For example, the gray class ("stone") and the gold class ("mirror") have overlapping distributions, while the two classes have obvious differences in appearance. The pale green class ("water") is clustered into two clusters, which is difficult for the classifier. On the contrary, our descriptors in Fig. 4(b) have clear boundaries between different classes, and each class is clustered tightly. There are large inter-class similarities and small intra-class similarities. It clearly illustrates that our method generates discriminative descriptors.

5. Conclusion

In this paper, we have presented a non-approximate infinite-dimensional feature aggregation method to generate a discriminative and compact descriptor. Our method directly aggregates infinite-dimensional features, avoiding approximation error compared to existing infinite-dimensional methods. By utilizing the factorized bilinear model, our method can not only capture second-order statistics of infinite-dimensional features but also tackle the issue that infinite-dimensional features are infeasible to store, represent and compute explicitly. Experiments demonstrate that our method achieves state-of-the-art performance on four image classification tasks. However, there still exists weakness: the Sigmoid kernel used in our method is a conditionally positive definite kernel, which causes limitations on the selection of hyper-parameters. In the future, we will establish a unified infinite-dimensional feature aggregation framework and investigate other kernel functions so as to provide flexible choices of kernel functions and infinite-dimensional features for different visual tasks.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by the Natural Science Foundation of China (NSFC) under Grants No. 62172041 and No. 62176021.

References

[1] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach.Intell. T-PAMI 34 (9) (2011) 1704–1716.
[2] J. Zhang, Y. Cao, Q. Wu, Vector of locally and adaptively aggregated descriptors for image feature representation, Pattern Recognit. (PR) (2021) 107952.
[3] H. Li, P. Xiong, H. Fan, J. Sun, DFANet: deep feature aggregation for real-time semantic segmentation, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9522–9531.
[4] J. Sun, Y. Li, H. Chen, B. Zhang, J. Zhu, MEMF: multi-level-attention embedding and multi-layer-feature fusion model for person re-identification, Pattern Recognit. (PR) (2021) 107937.
[5] T.-Y. Lin, A. RoyChowdhury, S. Maji, Bilinear CNN models for fine-grained visual recognition, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1449–1457.
[6] Q. Wang, P. Li, W. Zuo, L. Zhang, RAID-G: robust estimation of approximate infinite dimensional gaussian with application to material recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 4433–4441.
[7] Z. Gao, J. Xie, Q. Wang, P. Li, Global second-order pooling convolutional networks, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 3024–3033.
[8] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, S. Belongie, Kernel pooling for convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2921–2930.
[9] M. Engin, L. Wang, L. Zhou, X. Liu, DeepKSPD: learning kernel-matrix-based SPD representation for fine-grained image recognition, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 612–627.
[10] J. Cavazza, P. Morerio, V. Murino, Scalable and compact 3D action recognition with approximated RBF kernel machines, Pattern Recognit. (PR) 93 (2019) 25–35.
[11] Z. Yu, J. Yu, C. Xiang, J. Fan, D. Tao, Beyond bilinear: generalized multimodal factorized high-order pooling for visual question answering, IEEE Trans. Neural Netw. Learn.Syst. (T-NNLS) (2018) 5947–5959.
[12] S. Cai, W. Zuo, L. Zhang, Higher-order integration of hierarchical convolutional activations for fine-grained visual categorization, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 511–520.
[13] M. Harandi, M. Salzmann, F. Porikli, Bregman divergences for infinite dimensional covariance matrices, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1003–1010.
[14] M.H. Quang, M. San Biagio, V. Murino, Log-hilbert-schmidt metric between positive definite operators on hilbert spaces, in: Advances in Neural Information Processing Systems (NeurIPS), 2014, pp. 388–396.
[15] H. Quang Minh, M. San Biagio, L. Bazzani, V. Murino, Approximate log-hilbert-schmidt distances between covariance operators for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5195–5203.
[16] N. Pham, R. Pagh, Fast and scalable polynomial kernels via explicit feature maps, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 239–247.
[17] Y. Mukuta, T. Harada, Kernel approximation via empirical orthogonal decomposition for unsupervised feature learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 5222–5230.
[18] Y. Pan, T. Yao, Y. Li, T. Mei, X-linear attention networks for image captioning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10971–10980.
[19] Z. Zhang, M. Wang, Y. Huang, A. Nehorai, Aligning infinite-dimensional covariance matrices in reproducing kernel hilbert spaces for domain adaptation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3437–3445.
[20] B. Chen, J. Li, G. Wei, B. Ma, A novel localized and second order feature coding network for image recognition, Pattern Recognit. (PR) 76 (2018) 339–348.
[21] Z. Gao, Y. Wu, X. Zhang, J. Dai, Y. Jia, M. Harandi, Revisiting bilinear pooling: a coding perspective, in: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), 2020, pp. 3954–3961.
[22] Z. Yu, J. Yu, J. Fan, D. Tao, Multi-modal factorized bilinear pooling with co-attention learning for visual question answering, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 1821–1830.
[23] L. Chi, Z. Yuan, Y. Mu, C. Wang, Non-local neural networks with grouped bilinear attentional transforms, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11804–11813.
[24] K.-X. Chen, J.-Y. Ren, X.-J. Wu, J. Kittler, Covariance descriptors on a gaussian manifold and their application to image set classification, Pattern Recognit. (PR) (2020) 107463.
[25] Q. Wang, J. Xie, W. Zuo, L. Zhang, P. Li, Deep CNNs meet global covariance pooling: better representation and generalization, IEEE Trans. Pattern Anal. Mach.Intell. T-PAMI (2020). 1–1
[26] Q. Wang, L. Zhang, B. Wu, D. Ren, P. Li, W. Zuo, Q. Hu, What deep CNNs benefit from global covariance pooling: an optimization perspective, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10771–10780.
[27] Z. Gao, Y. Wu, X. Bu, T. Yu, J. Yuan, Y. Jia, Learning a robust representation via a deep network on symmetric positive definite manifolds, Pattern Recognit. (PR) 92 (2019) 1–12.
[28] C. Ionescu, O. Vantzos, C. Sminchisescu, Matrix backpropagation for deep networks with structured layers, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015, pp. 2965–2973.
[29] Y. Gao, O. Beijbom, N. Zhang, T. Darrell, Compact bilinear pooling, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 317–326.
[30] Y. Li, N. Wang, J. Liu, X. Hou, Factorized bilinear models for image recognition, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017a, pp. 2079–2087.
[31] P. Li, J. Xie, Q. Wang, W. Zuo, Is second-order information helpful for large-scale visual recognition? in: Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017b, pp. 2070–2078.
[32] X. Liu, J. Geng, H. Ling, Y.-m. Cheung, Attention guided deep audio-face fusion for efficient speaker naming, Pattern Recognit. (PR) 88 (2019) 557–568.
[33] J. Li, X. Xie, Q. Pan, Y. Cao, Z. Zhao, G. Shi, SGM-Net: skeleton-guided multimodal network for action recognition, Pattern Recognit. (PR) 104 (2020) 107356.
[34] H.-T. Lin, C.-J. Lin, A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods, Neural Comput. 3 (2003) 1–32.
[35] K. Yu, M. Salzmann, Statistically-motivated second-order pooling, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 600–616.
[36] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: International Conference on Learning Representations (ICLR), 2015.
[37] S. Kong, C. Fowlkes, Low-rank bilinear pooling for fine-grained classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 365–374.
[38] S. Maji, E. Rahtu, J. Kannala, M. Blaschko, A. Vedaldi, Fine-grained visual classification of aircraft, arXiv preprint arXiv:1306.5151(2013).
[39] J. Krause, M. Stark, J. Deng, L. Fei-Fei, 3D object representations for fine-grained categorization, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2013, pp. 554–561.
[40] A. Quattoni, A. Torralba, Recognizing indoor scenes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009, pp. 413–420.
[41] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3606–3613.
[42] S. Bell, P. Upchurch, N. Snavely, K. Bala, Material recognition in the wild with the materials in context database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 3479–3487.

Jindou Dai received the B.S. degree in computer science from Beijing Institute of Technology (BIT), Beijing, China, in 2019. He is now a M.S. candidate in the Beijing Laboratory of Intelligent Information Technology, BIT, Beijing, China. His research interests include pattern recognition, machine learning, and computer vision on Riemannian manifolds.

Yuwei Wu received the Ph.D. degree in computer science from Beijing Institute of Technology (BIT), Beijing, China, in 2014. He is now an Assistant Professor at School of Computer Science, BIT. From August 2014 to August 2016, he was a post-doctoral research fellow at School of Electrical Electronic Engineering (EEE), Nanyang Technological University (NTU), Singapore. He has strong research interests in computer vision and machine learning. He received Distinguished Dissertation Award Nominee from China Association for Artificial Intelligence (CAAI).

Zhi Gao received the B.S. degree in computer science from Beijing Institute of Technology (BIT), Beijing, China, in 2017. Now, he is a Ph.D. candidate in the Beijing Laboratory of Intelligent Information Technology, BIT, Beijing, China. His research interests include pattern recognition, machine learning, and computer vision on Riemannian manifolds.

Yunde Jia is Professor of Computer Science at BIT, and serves as the Director of the Beijing Laboratory of Intelligent Information Technology. He received the B.S., M.S., and Ph.D. degrees in Mechatronics from the Beijing Institute of Technology (BIT) in 1983, 1986, and 2000, respectively. He has previously served as the Executive Dean of the School of Computer Science at BIT from 2005 to 2008. He was a Visiting Scientist at Carnegie Mellon University from 1995 to 1997, and a Visiting Fellow at the Australian National University in 2011. His current research interests include computer vision, media computing, and intelligent systems.