An Efficient End-to-End Framework for Localized Second-Order Pooling

Anonymous authors
Paper under double-blind review

Abstract

Second-order pooling has proven effective for deep image classification by representing an image with the covariance matrix of its local feature descriptors. However, the diverse visual content in images leads to local descriptors being distributed as multiple modes in the feature space, limiting the effectiveness of a single global covariance matrix. In this work, we propose an efficient end-to-end framework for localized second-order pooling. Our approach jointly learns clusters of local feature descriptors across the entire training set, adaptively assigns the descriptors of each image to the appropriate clusters, computes a localized covariance matrix based on the descriptors assigned to each cluster, and then integrates these matrices to form the image representation.This is achieved through an attention-based local cluster mining branch that automatically identifies the clusters that each local descriptor shall be assigned to. Furthermore, to manage the significant computational overhead incurred by the use of multiple local covariance matrices, we utilize a simple but efficient sample-adaptive feature fusion scheme. This scheme adaptively generates fusion weights for each localized covariance matrix using a lightweight predictor, ensuring both computational efficiency and flexibility. Extensive experiments on multiple fine-grained and large-scale image classification datasets demonstrate that our method consistently improves performance when integrated with state-of-the-art second-order pooling methods and leading network architectures. Ablation studies further verify the efficiency of our feature fusion scheme compared to the existing common alternatives.

1 Introduction

With the widespread adoption of deep neural networks, an image is typically transformed—via a series of layer-by-layer projections—into an $h \times w \times d$ feature map, consisting of hw local feature descriptors, each with d channels. To derive a global image representation from this feature map, pooling operations are commonly applied. While sum-pooling and max-pooling capture only the first-order statistics of local descriptors, second-order pooling goes a step further by extracting covariance information among the channels (i.e., the dimensions of the local descriptors). This process produces a $d \times d$ covariance matrix, which, after appropriate processing, is reshaped into a long feature vector to represent the image. Second-order pooling has demonstrated strong performance across a range of computer vision tasks (Sun et al., 2021; Zhang et al., 2020; Rahman & Moghadam, 2024), with fine-grained classification (Li et al., 2018; Rahman et al., 2023) often cited as a representative application.

Most existing second-order pooling methods (Li et al., 2018; Rahman et al., 2020; Engin et al., 2018; Dai et al., 2022; Rahman et al., 2023; Gao et al., 2019) compute a single global covariance matrix over all local descriptors extracted from an image. This practice implicitly assumes that the distribution of these local descriptors in the feature space is unimodal. However, given the rich and diverse visual content present in real-world images, this assumption rarely holds, and a single global covariance matrix may fail to accurately capture the complex second-order statistics needed for effective representation.

To overcome this limitation, some studies have attempted localized second-order pooling strategies (Li et al., 2017b; Wang et al., 2018). They either apply offline clustering to fixed deep features or replace clustering

with multiple projection layers. The former breaks the interaction between clustering and feature learning, while the latter often yields redundant or misaligned subspaces. Although effective to some extent, these approaches do not provide a unified, end-to-end framework for learning descriptors, discovering latent structure, and computing localized second-order representations.

To improve this situation, we propose an efficient end-to-end framework for localized second-order pooling, which computes covariance matrices within the clusters that are jointly and explicitly discovered during network training. Specifically, we introduce an attention-based local cluster mining branch into the deep network, which is jointly trained with the network on the entire dataset. This branch automatically identifies the underlying clusters from training data, encouraging local descriptors with similar visual patterns to be grouped together as they are learned. To represent an image, instead of using all local descriptors from this image to compute a single global covariance matrix, the proposed network adaptively assigns the descriptors of this image to the appropriate clusters, computes a localized covariance matrix based on the descriptors assigned to each cluster. This online clustering process effectively partitions the feature space into multiple regions, each corresponding to semantically coherent groups. Within each region, second-order statistics can be captured more accurately by the corresponding covariance matrix.

Furthermore, we address a direct consequence of localized approach, that is, the significant computational overhead arising from the increased number of covariance matrices. In practice, each $d \times d$ covariance matrix is typically reshaped into a feature vector of dimension $d(d+1)/2$ by retaining only the upper triangular part due to symmetry. Simply concatenating all local covariance features (Li et al., 2017b) greatly increases dimensionality and model parameters. To address the computational overhead, we propose a sample-adaptive feature fusion scheme that dynamically aggregates local covariance features. Instead of relying on fixed or global weighting, our method learns dynamic, instance-specific fusion weights that model the contribution of each of the localized covariance matrices to the final class prediction, resulting in a more compact yet discriminative representation.

Together, these components constitute an efficient end-to-end framework for localized second-order feature pooling. Our framework not only dynamically captures complex channel-wise relationships but also effectively manages computational costs. Furthermore, experimental results demonstrate that our approach integrates well with various deep neural network architectures and enhances state-of-the-art second-order pooling methods across both fine-grained and large-scale classification benchmarks.

Our contributions are summarized as follows:

1. We propose an efficient framework for localized second-order pooling in an end-to-end manner. This framework jointly learns feature descriptors and identifies their clusters across the entire training set, facilitating more effective second-order pooling for feature descriptors.

2. To mitigate the increase on computational costs, we develop a sample-adaptive feature fusion scheme. This scheme flexibly merges features from all local covariance matrices per image, not only helping to improve performance but also better reducing computational overhead compared to other fusion alternatives.

3. We perform extensive experimental studies on both fine-grained and large-scale image classification datasets, validating the efficacy of the proposed framework.

2 Related work

Second-order pooling has long been recognized as a powerful tool for capturing pairwise correlations among feature components. Initially employed in classical vision tasks such as object classification and tracking (Tuzel et al., 2006; 2007), covariance matrices served as standalone region descriptors. With the advent of deep learning, they have been integrated into neural networks as trainable modules. Pioneering works such as Bilinear CNN (Lin et al., 2015) and DeepO_2P (Ionescu et al., 2015) laid the foundation for end-to-end second-order pooling. Subsequent efforts addressed the challenges posed by symmetric positive definite (SPD) matrices—namely their high dimensionality and non-Euclidean geometry—through techniques such

as matrix normalization (Li et al., 2018; Koniusz et al., 2018; Gao et al., 2019; Wang et al., 2021; Song et al., 2021; 2023a;b), compact and low-rank pooling (Gao et al., 2016; Kong & Fowlkes, 2017; Zheng et al., 2019; Rahman et al., 2020; Yu et al., 2020; 2021; Wang et al., 2022; Yu et al., 2022b), linear transformations (Brooks et al., 2019; Huang & Gool, 2017; Nguyen et al., 2019; Yu & Salzmann, 2017; Wang et al., 2021), and inverse covariance (Rahman et al., 2023).

Global Second-Order Pooling Approaches. Among these advances, global second-order pooling approaches compute a single covariance matrix over all local descriptors from an image, simplifying the representation pipeline while retaining expressive second-order statistics. Notable examples include matrix square-root normalization (Li et al., 2018) and compact bilinear pooling (Gao et al., 2016), which improve representation quality through power normalization and dimensionality reduction. Kernel pooling methods (Cui et al., 2017) further enrich the representation by projecting features into high-dimensional Hilbert spaces. Despite their effectiveness, these global pooling techniques inherently assume feature homogeneity and often dilute critical local variations, limiting their capacity to capture fine-grained distinctions in complex visual scenes. As a result, the performance may be suboptimal in tasks that require detailed spatial discrimination.

Local Second-Order Pooling Approaches. To address the global approach's limitations, localized pooling methods compute multiple covariance matrices for distinct feature regions or in different subspaces, offering more nuanced representations. High-order local pooling (Li et al., 2017b) encodes local descriptors as Gaussian components based on a pre-learned dictionary, where feature grouping and descriptor extraction are conducted in a decoupled manner. In contrast, GM-SOP (Wang et al., 2018) learns a set of nonlinear projection layers that implicitly map local features into multiple subspaces, where separate local covariance matrices are computed. Although these methods enhance representation fidelity, there is still room for further improvement and exploration. High-order local pooling decouples clustering from descriptor learning, relying on offline processes, while GM-SOP does not explore the potential of an explicit clustering mechanism within its end-to-end learning framework. In contrast, our proposed framework integrates an attention-driven cluster mining mechanism directly within the network, enabling explicit feature space partition for localized second-order pooling in an end-to-end training manner.

In addition, the feature fusion scheme in GM-SOP remains limited, as the softmax-normalized weighted sum restricts the final representation to a convex combination of covariance experts, often leading to over-smoothed second-order features. Therefore, the method relies on an additional diversity regularizer to prevent the experts from collapsing into similar subspaces, indicating that the fusion itself provides limited incentive for learning truly distinct structures.

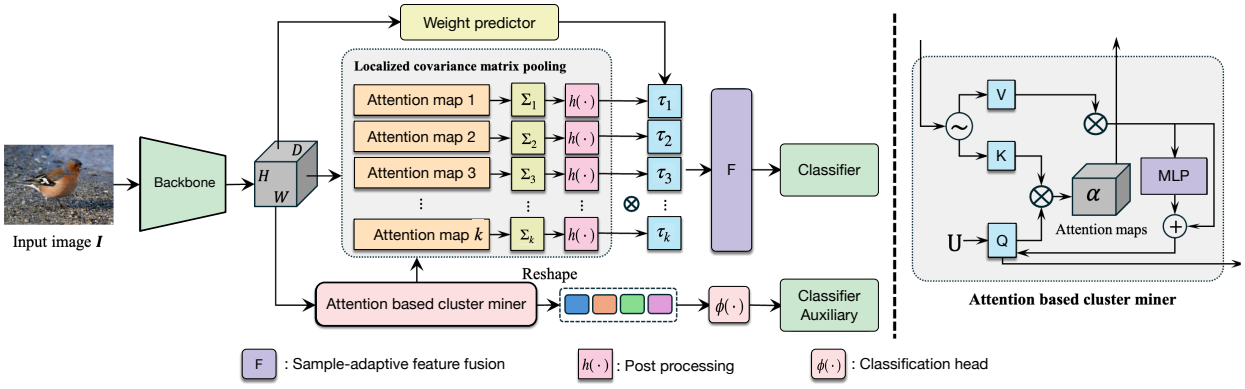


Figure 1: Overview of the proposed end-to-end localized second-order pooling framework (LSOP), including the attention-based cluster miner, localized covariance computation, sample-adaptive weight predictor, and feature fusion.

3 Proposed method

Our method consists of three main components: 1) an attention-based local cluster mining branch, which groups local descriptors into visually similar clusters; 2) a localized covariance matrix branch, which assigns the local descriptors of an input image and computes a covariance matrix for each group, followed by post-processing steps such as normalization and vectorization; and 3) a sample-adaptive feature fusion scheme, which generates fusion weights to aggregate the localized covariance features for final classification. The overall pipeline is illustrated in Figure 1. The following subsections describe each component in detail.

3.1 Attention-based Cluster Mining

The goal of our online clustering component is to perform clustering and feature learning simultaneously during training, while maintaining a consistent ordering of clusters across images so that downstream fusion weights can be reliably applied to the localized covariance features. This requires a differentiable, data-adaptive procedure that updates cluster assignments in real time and yields stable, repeatable indices. We considered several alternatives. VQ-based approaches (van den Oord et al., 2017; Esser et al., 2021) use hard assignments to a static codebook, which can restrict gradient flow and make the ordering of codes across images less stable. DeepCluster-style methods (Caron et al., 2018; Zhan et al., 2020) typically rely on separate or batch-level clustering steps (e.g., k-means) and thus decouple assignment from the main optimization, often leading to label churn and additional synchronization overhead. In contrast, we adopt a slot attention scheme (Locatello et al., 2020), which performs soft, differentiable assignments and updates the slot prototypes jointly with the network. Recent work has noted its connection to k-means (Yu et al., 2022a), while retaining the benefit that cluster updates are directly driven by the task loss. This yields an online procedure with stable slot indexing across images and efficient end-to-end optimization, which is well matched to our setting.

Slot-attention Module. The basic idea of slot attention is to embed a set of m input vectors into a set of K output vectors that refer to as the slots in an iterative way. In our case, the input vectors are the hw local descriptors from the feature map $\mathbf{F} \in \mathbb{R}^{d \times hw}$ and the output vectors are the K learnable slots $\mathbf{U} \in \mathbb{R}^{d \times K}$. At each iteration, three learnable linear transformations $\mathbf{q}$, $\mathbf{k}$, and $\mathbf{v}$ are used to map inputs and slots to a common dimension d and a cross-attention operation is conducted followed by a residual network. In particular, the slots $\mathbf{U}^{t+1}$ at the $(t+1)$th iteration can be derived from the previous ones $\mathbf{U}^t$ with the following forms:

$$
\begin{aligned}
&\hat{\boldsymbol{\alpha}} = \mathrm{softmax}_k \frac{\mathbf{q}(\mathbf{U}^t)^\top \cdot \mathbf{k}(\mathbf{F})}{\sqrt{d}}, \\
&\boldsymbol{\alpha} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_K]^\top, \boldsymbol{\alpha}_k = \frac{\hat{\boldsymbol{\alpha}_k}}{\sum_{hw} \hat{\boldsymbol{\alpha}_{ki}}}, \\
&\mathbf{U}^{t'} = \mathbf{U}^t + \mathbf{v}(\mathbf{F}) \cdot \boldsymbol{\alpha}^\top, \\
&\mathbf{U}^{t+1} = \mathbf{U}^{t'} + \mathrm{MLP}(\mathrm{LayerNorm}(\mathbf{U}^{t'})),
\end{aligned}
\tag{1}
$$

where the subscript k indicates that the softmax is applied along the slot dimension. The above process will be conducted T times to obtain the final slots $\mathbf{U}^T$. In the meantime, the obtained attention map $\boldsymbol{\alpha} \in \mathbb{R}^{K \times hw}$ from the last iteration is used to represent the assignment for each local descriptor, as it denotes their probabilities of being assigned to the slots.

Module Training as Dictionary-based Encoding. Having described the internal design of the slot attention block, we now turn to how this module is trained within our framework. In prior works, slot attention is often supervised either through slot-level objectives, such as contrastive losses (Wen et al., 2022; Weinzaepfel et al., 2022) assigned to individual slots, or via unsupervised reconstruction-based losses (Locatello et al., 2020) that encourage slots to decompose an image into distinct parts. However, in our case, supervision is provided solely at the image level through class labels. This setting calls for a training scheme

that can effectively guide the slot attention module to learn meaningful and discriminative representations without direct supervision at the slot level. To this end, we adopt a simple yet effective design: after the final iteration of slot refinement, we concatenate all slot embeddings and feed the resulting vector into a lightweight embedding layer followed by a standard classifier trained with cross-entropy loss. This allows each slot to specialize through gradient signals propagated from the global objective, encouraging implicit feature decomposition in an end-to-end fashion.

Specifically, we concatenate the slots of the final iteration, $\mathbf{U}^T \in \mathbb{R}^{d\times K}$, into a vector $\mathbf{u}' \in \mathbb{R}^{dK}$, and apply an embedding layer before feeding them into the auxiliary classifier, which is trained using the following online clustering loss:

$$L_{oc} = -\sum_{i=1}^{N} y_c \cdot \log\left(\frac{\exp\left(\phi(\mathbf{u}'_i)\cdot\omega_c\right)}{\sum_{j=1}^{C}\exp\left(\phi(\mathbf{u}'_i)\cdot\omega_j\right)}\right), \tag{2}$$

where $\phi(\cdot)$ denotes an embedding function consisting of a layer normalization, a ReLU, and a linear layer. Here, ω_c and y_c represent the classification weight and label for class c, respectively. C and N denote the total number of classes and the number of samples in a mini-batch, respectively.

This design is inspired by dictionary-based encoding methods such as VLAD (Jégou et al., 2010) and its variants, which were popular in the Bag-of-Words (BoW) era. These methods assign local descriptors to a set of predefined codewords and aggregate the residuals between descriptors and their assigned centers. The resulting residual vectors are concatenated to form a global image representation for classification. Our slot attention mechanism follows a similar principle. The initial learnable slots $\mathbf{U}^0$ act as cluster centers, and a soft assignment is performed, where each local descriptor contributes to multiple slots based on attention weights. These assignments are then aggregated to update the slots, and this update implicitly resembles a residual connection between the aggregated features and the current slot state. Unlike conventional dictionary-based encodings, our slots are iteratively refined, but the final representation is still formed by concatenating all slot embeddings for classification.

3.2 Localized Covariance Matrices

Before computing localized covariance matrices, it is important to clarify that the attention-based cluster mining module only provides soft assignments that assign local descriptors into semantically coherent groups; it does not perform any statistical aggregation or pooling. Instead, the second-order pooling of local descriptors is conducted in each cluster individually, by computing a localized covariance matrix within each cluster.

Recall that we have obtained the attention map $\boldsymbol{\alpha}$ denoting the soft assignments of each local descriptor with respect to the learned clusters. Instead of hard assignment, we use soft assignment here by considering two issues. First, it avoids the situation that the local descriptors from a given image are only assigned to a subset of the K clusters. This situation usually happens when the visual content of the image does not cover the semantic patterns represented by some cluster centers. In other words, if the number of clusters is limited or if certain centers correspond to rare or specific visual structures, and those structures are absent in the image, then none of the local descriptors from that image will be assigned to those centers. In this case, the local covariance matrices corresponding to the empty clusters cannot be calculated. Second, it fully utilizes all the local descriptors from an image to more reliably estimate each local covariance matrix. This helps to mitigate the small sample issue when only using the descriptors hard-assigned to a specific cluster to estimate the covariance matrix therein. The attention map for each cluster is used as weights to compute a weighted covariance matrix $\boldsymbol{\Sigma}_k$ $(k = 1, \cdots, K)$ for the k-th cluster, with its (p, q)-th entry as

$$\boldsymbol{\Sigma}_k(p,q) = \frac{1}{1-\|\boldsymbol{\alpha}_k\|_2^2}\sum_{i=1}^{hw}\alpha_{ki}(x_{ip}-\hat{\mu}_{kp})(x_{iq}-\hat{\mu}_{kq}), \tag{3}$$

where $\boldsymbol{\alpha}_k = (\alpha_{k1}, \alpha_{k2}, \cdots, \alpha_{k(hw)})^\top$ and x_{ip} denotes the p-th component of the local descriptor $\mathbf{x}_i$ while $\hat{\mu}_{kp}$ is the weighted mean defined as $\hat{\mu}_{kp} = \sum_{i=1}^{hw}\alpha_{ki}x_{ip}$ over all local descriptors from an image. Repeating this for each local cluster produces the K local covariance matrices.

3.3 Feature Fusion of Local Covariance Matrices

With the presence of multiple local covariance matrices, a fusion scheme that can lead to more accurate classification while avoiding a substantial increase on computational cost is needed to fuse the features extracted from the matrices.

A straightforward baseline is to directly **concatenate** the K features into a single vector, as done in (Li et al., 2017b). While this approach retains complete information from all branches, it causes a substantial increase in dimensionality[1] and ignores the relative importance or redundancy among features. To address this, one can introduce a **cross-attention** module, allowing each branch to interact with others dynamically. For example, we could treat the feature vector from a global covariance matrix as the *key* and the feature vector from each local matrix as the *query* and *value*. By cross-attention, the K local feature vectors can be combined by a weighted combination. This enables context-aware fusion but comes at the cost of increased parameters and compute, which may be prohibitive in certain settings. A more compact alternative is to predict a set of K normalized weights (via softmax) that reflect the relative importance of each branch, and use them to perform a **weighted sum** of features. This approach is lightweight and interpretable, but it treats branches independently and lacks the ability to model richer interactions. Building on this, a **gated selection mechanism** can be employed to enforce sparsity by selecting a subset of $K - n$ branches, before applying softmax-weighted fusion, as done in GM-SOP (Wang et al., 2018). Although this introduces stronger selection capability, it depends on the design of the gating function and the choice of the sparsity level, i.e., how many branches to retain.

To achieve more efficient fusion, we propose a **sample-adaptive fusion** scheme that avoids discrete selection altogether. Instead of gating, we predict K continuous scaling factors that softly modulate the contribution of each branch. This approach eliminates the need to set hard thresholds or sparsity levels, while still allowing the model to downweight less relevant features and amplify more informative ones, leading to a more flexible and stable fusion.

Sample-adaptive Feature Fusion. The basic idea of this fusion scheme is to measure the importance of each bunch of local descriptors that are assigned to the same cluster and use it to rescale the contribution of the features in the classification loss. To achieve that, a weight predictor consisting of a global average pooling followed by a dimension reduction layer is used to measure the scalers $\boldsymbol{\tau} \in \mathbb{R}^K$ for all the matrices as follows:

$$\boldsymbol{\tau} = \text{sigmoid}(g(\mathbf{F}) \cdot \mathbf{w}), \tag{4}$$

where $\mathbf{F}$ is a $h \times w \times d$ feature map, $g(\cdot)$ is global average pooling, $\mathbf{w} \in \mathbb{R}^{d \times K}$ is the dimension reduction layer, and sigmoid$(\cdot)$ controls the range of the output. After that, $\boldsymbol{\tau}$ is used to re-weigh the covariance matrices, with a carefully designed regularization term imposed on $\boldsymbol{\tau}$. Our loss function is defined as follows:

$$\mathcal{L}_{sa} = -\sum_{i=1}^{N} y_c \cdot \log\left(\frac{\exp\left(\mathbf{f}_i \cdot \omega_c\right)}{\sum_{j=1}^{C} \exp\left(\mathbf{f}_i \cdot \omega_j\right)}\right) + \beta \sum_{i=1}^{N}\sum_{k=1}^{K} \log\left(\frac{1}{\tau_k}\right), \text{where } \mathbf{f}_i = \sum_k \frac{h(\boldsymbol{\Sigma}_k)}{\tau_k} \tag{5}$$

$h(\cdot)$ denotes the operations of matrix normalization and reshaping applied to $\boldsymbol{\Sigma}_k$, producing a $(d(d+1)/2)$-dimensional feature vector. β is the coefficient to control the penalty on the small τ.

After assembling all three components together, the proposed framework could be trained with image-wise annotations in an end-to-end manner by integrating the two losses in Eq. (2) and Eq. (5) as

$$\mathcal{L}_{total} = \mathcal{L}_{sa} + \lambda \mathcal{L}_{oc}. \tag{6}$$

4 Experiment

In this section, we first introduce the datasets, evaluation metrics, and implementation details used in our experimental study. Next, we compare three mainstream covariance matrix-based feature pooling methods

[1]The dimension of a covariance matrix feature is usually $d(d+1)/2$ after the common steps of normalization and reshaping.

Table 1: Accuracy comparison (top-1%) between three representative covariance matrix based feature pooling methods (iSQRT (Li et al., 2018), iSICE (Rahman et al., 2023), and DeepKSPD (Engin et al., 2018)) and the ones obtained by applying the proposed localized approach. "+ LSOP" denotes the performance is obtained by applying the proposed localized second-order pooling method. The second column represents the types of second-order pooling, and "cov", "pco", and "ker" denote *conventional covariance matrix*, *partial correlation*, and *kernel based non-linear* second-order pooling respectively.

Backbone	Type	Methods	Birds	Cars	Airplane	Dogs	MIT	Ave Gain
ResNet-50	cov	iSQRT	84.1 ± 0.4	92.4 ± 0.3	86.5 ± 0.5	84.5 ± 0.3	79.0 ± 1.1	↑1.4
		+ LSOP (ours)	**86.0 ± 0.1**	**93.3 ± 0.1**	**87.2 ± 0.2**	**86.5 ± 0.1**	**80.7 ± 0.5**	
	pco	iSICE	85.9 ± 0.4	93.2 ± 0.2	89.4 ± 0.2	84.6 ± 0.2	80.0 ± 0.3	↑1.7
		+ LSOP (ours)	**87.1 ± 0.0**	**94.5± 0.0**	**89.7 ± 0.2**	**86.9± 0.3**	**83.3 ± 0.3**	
	ker	DeepKSPD	**86.1 ± 0.2**	92.0 ± 0.2	83.2 ± 0.1	87.5 ± 0.3	82.9 ± 0.3	↑1.0
		+ LSOP (ours)	85.9 ± 0.1	**92.2 ± 0.2**	**85.8 ± 0.1**	**88.7 ± 0.2**	**84.0 ± 0.5**	
ResNet-101	cov	iSQRT	84.7 ± 0.6	92.9 ± 0.1	86.7 ± 0.3	85.1 ± 0.5	79.0 ± 0.2	↑1.8
		+ LSOP (ours)	**87.0 ± 0.1**	**94.0 ± 0.0**	**88.8 ± 0.0**	**86.9 ± 0.1**	**80.7 ± 0.4**	
	pco	iSICE	86.7 ± 0.3	93.7 ± 0.1	89.8 ± 0.2	85.3 ± 0.1	80.5 ± 0.4	↑1.6
		+ LSOP (ours)	**87.3 ± 0.1**	**94.5 ± 0.2**	**90.1 ± 0.1**	**88.4 ± 0.2**	**83.7 ± 0.1**	
	ker	DeepKSPD	**87.1 ± 0.0**	92.2 ± 0.4	85.8 ± 0.2	88.3 ± 0.1	82.6 ± 0.5	↑1.0
		+ LSOP (ours)	86.9 ± 0.2	**92.4 ± 0.1**	**87.7 ± 0.3**	**88.9 ± 0.2**	**85.2 ± 0.5**	
Swin-T	cov	iSQRT	86.2 ± 0.0	92.0 ± 0.1	83.2 ± 0.0	88.8 ± 0.0	81.2 ± 1.0	↑1.2
		+ LSOP (ours)	**86.9 ± 0.0**	**92.6 ± 0.3**	**84.9 ± 0.2**	**88.9 ± 0.1**	**84.0 ± 0.6**	
	pco	iSICE	86.9 ± 0.1	92.7 ± 0.1	86.7 ± 0.3	86.9 ± 0.1	84.3 ± 0.8	↑0.8
		+ LSOP (ours)	**87.8 ± 0.1**	**93.4 ± 0.1**	**87.5 ± 0.1**	**88.0 ± 0.1**	**84.9 ± 0.6**	
	ker	DeepKSPD	86.8 ± 0.1	92.2 ± 0.1	83.9 ± 0.0	89.2 ± 0.1	85.9 ± 0.7	↑0.9
		+ LSOP (ours)	**86.9 ± 0.1**	**92.5 ± 0.2**	**85.8 ± 0.2**	**89.9 ± 0.1**	**87.2 ± 0.4**	
DINO-v2	cov	iSQRT	86.7 ± 0.0	92.7 ± 0.0	74.0 ± 0.0	88.0 ± 0.0	91.5 ± 0.0	↑0.7
		+ LSOP (ours)	**87.4 ± 0.0**	**93.8 ± 0.0**	**75.2 ± 0.0**	**88.2 ± 0.0**	**91.6 ± 0.0**	
	pco	iSICE	88.4 ± 0.0	93.7 ± 0.0	**79.6 ± 0.0**	88.8 ± 0.0	91.2 ± 0.0	↑0.2
		+ LSOP (ours)	**88.9 ± 0.0**	**94.2 ± 0.0**	78.4 ± 0.0	**89.1 ± 0.0**	**91.9 ± 0.0**	
	ker	DeepKSPD	83.6 ± 0.0	92.3 ± 0.0	71.8 ± 0.0	86.7 ± 0.0	91.7 ± 0.0	↑0.4
		+ LSOP (ours)	**83.7 ± 0.0**	**92.3 ± 0.0**	**72.9 ± 0.0**	**87.1 ± 0.0**	**92.1 ± 0.0**	

with their counterparts enhanced by the proposed approach. After that, the performance on large-scale dataset is reported. We then present a comprehensive comparison of different feature fusion schemes. Finally, we conduct an ablation study to further investigate the properties of the proposed method.

4.1 Datasets, Metric, and Implementation

Dataset. We evaluate our method on four fine-grained visual categorization (FGVC) datasets, i.e., Birds (Wah et al., 2011), Cars (Krause et al., 2013), Airplane (Maji et al., 2013), and Dogs (Dataset, 2011), one scene dataset, i.e., MIT (Quattoni & Torralba, 2009), and one large-scale visual classification dataset i.e., ImageNet-1K (Deng et al., 2009). For our experiments, we use the protocol of Bilinear CNN (Lin et al., 2015) which is largely followed in the literature for the first five datasets and follow the protocol in their original papers for the last one. Further details of the datasets are provided in the supplementary materials.

Metric. We use average classification accuracy as in literature to evaluate and compare our method, e.g., (Lin et al., 2015; Li et al., 2018; Rahman et al., 2020) for the first five datasets, but top-1 and top-5 error rates for ImageNet-1K instead.

Implementation detail. We implement our method using PyTorch, which serves as the platform for both our codebase and the pretrained backbone models. For all datasets except ImageNet-1K, we adopt backbones initialized with weights pretrained on the ImageNet-1K benchmark. For experiments on ImageNet-1K itself, models are trained from scratch. For all covariance matrix-based feature pooling methods, we follow the

settings in prior works (Li et al., 2017a; 2018; Rahman et al., 2020), where the number of feature channels is reduced to 256 using a 1×1 convolution for computational efficiency. To ensure consistency with existing methods, input images are resized to 448×448, and random horizontal flipping is the only data augmentation applied. Models are trained for 100 epochs with an initial learning rate of 0.00012, which is reduced by a factor of 10 at the 45th and 90th epochs on the first five datasets. We set the hyperparameters $K = 4$, $\beta = 0.1$, and $\lambda = 1$ throughout these experiments. For the ImageNet-1K, input images are resized to 224×224 in accordance with its standard training protocol. The models are trained for 65 epochs from scratch using SGD with an initial learning rate of 0.1, which is decayed by a factor of 10 at the 30th and 45th epochs. Further experimental details can be found in the supplementary materials.

4.2 Comparison with Representative Global Methods

Table 2: Impact of feature fusion schemes on computational cost, number of model parameters, inference time per image, and top-1 classification accuracy (%). The symbols "*cro*", "*wsp*", "*gs*", "*con*", and "*saf*" denote cross-attention, weighted sum prediction, gated selection (the fusion scheme used in (Wang et al., 2018)), concatenation, and sample-adaptive feature fusion scheme, respectively.

Method	Fusion	Birds	Cars	Airplane	Dogs	MIT	MACs (G)	#Para (M)	Latency (ms)	Avg. Top-1 Acc. (%)
iSQRT (baseline)		84.7	92.9	86.7	85.1	79.0	8.07	49.6	10.4	85.7
+ LSOP	*cro*	85.8	93.1	87.9	85.2	79.7	9.04	86.2	12.7	86.3 (↑0.6)
	wsp	86.0	93.4	87.5	85.7	79.7	8.75	52.6	12.6	86.4 (↑0.7)
	gs	85.6	93.3	87.7	86.5	79.5	9.80	56.7	14.5	86.5 (↑0.8)
	con	85.7	93.4	88.2	**87.1**	**82.5**	8.77	72.3	12.6	87.4 (↑1.7)
	saf (ours)	**87.0**	**94.0**	**88.8**	86.9	80.7	8.75	52.6	12.6	87.5 (↑**1.8**)

To verify the effectiveness and applicability of the proposed localized method, we select three representative covariance matrix based feature pooling methods, iSQRT (Li et al., 2018), iSICE (Rahman et al., 2023), and DeepKSPD (Engin et al., 2018) as our baselines to make a comparison. iSQRT introduces an iterative way to fast normalize the covariance matrix and this approach has been one of the most widely used approaches in recent covariance matrix pooling related work. As one of the most recent second-order pooling methods, iSICE proposes a sparse partial correlation algorithm to address the confounding issue and it achieves state-of-the-art performance on many datasets. DeepKSPD utilizes a non-linear kernel to describe the relationship between channels in the training process, that could be regarded as a parametric-based covariance pooling. In this study, we apply our method to all three baseline methods to give a direct and clear comparison to better demonstrate its effect on performance and applicability. In addition, four widely used backbones, i.e., ResNet-50, ResNet-101 (He et al., 2016), Swin-T (Liu et al., 2021), and DINOv2 (Oquab et al., 2024), are utilized to help further verify the applicability of our method on different deep architectures. In particular, the first three backbones are involved in parameter updates during training, while DINOv2 remains entirely frozen throughout the training process to simulate transfer learning from a foundation model. To make a fair comparison, all the baseline methods are re-implemented with the training settings from iSQRT. We also simply follow this setting for our method instead of further parameter tuning to purely demonstrate the impacts coming from the proposed method. For datasets without predefined validation or test splits, we randomly divide the original test set into 20% for validation and 80% for testing. The final results are reported on the test set using the model checkpoint that achieves the best performance on the validation set during training.

4.3 Feature Fusion Alternatives

In this subsection, our target is to verify the efficacy of the sample-adaptive feature fusion (referred to as *saf*) scheme. Specifically, we make the study from the following two perspectives. On one hand, we compare *saf* with other four commonly used feature fusion schemes as mentioned in the subsection "Feature Fusion of Local Covariance Matrices", i.e., concatenation (*con*), cross-attention (*cro*), weighted sum prediction (*wsp*), and gated selection (*gs*) by evaluating their classification accuracy. On the other hand, we also report the computational cost in GMACs (Sovrasov, 2018-2024) and inference time with a 224×224 input image and

the number of model parameters, along with the accuracy to provide an overall performance on the trade-off between accuracy and efficiency. For this comparison, iSQRT is used as baseline.

From Table 2, we can see that fusing localized covariance features with all the five mentioned schemes could improve the performance on the five datasets comparing with the baseline. That indicates the proposed attention-based cluster mining effectively benefits the localized second-order pooling no matter what fusion scheme is used. In light of the computational complexity analysis, *saf* shows more advances on keeping good balances between the accuracy and efficiency. As shown, it takes the least extra computational cost (only 8.3% GMac and 5.6% parameters increase) but obtains the most average performance gain (1.8% increase) among the five schemes. This verifies the capability of our framework on both accuracy and efficiency.

The performances are reported in Table 1. In particular, we partition the table into four parts according to the backbone type and provide a individual comparison between the baselines and the ones obtained by applying our method (indicated by "+ LSOP") within each part. An average performance gain on all five datasets is also given in the last column to better demonstrate the impact of our method. Generally, our method obtains better or competitive results on all datasets with all backbone models. In most cases, we could improve the performance by more than 1% for two CNN backbones. In particular, we achieve a performance gain of about 3.3% on MIT with iSICE when using ResNet-50. As to the ViT variants, our method obtains the overall best performance on all datasets. In a nut shell, the performance gain verifies that our method is widely applicable to various backbones and existing covariance matrix based feature pooling methods. This further demonstrates the benefits of conducting localized second-order pooling within the local clusters.

Table 3: Error rate (top-1/top-5%) performance on large-scale classification dataset. † denotes our reimplementation.

Method	ImageNet-1K	
	Top-1 ↓	Top-5 ↓
iSQRT (Li et al., 2018)†	20.97	5.73
GM-SOP (Wang et al., 2018)†	20.42	5.31
iSQRT + LSOP (ours)	**20.12**	**5.04**

4.4 On Large-scale Dataset

We further evaluate our method on the ImageNet-1K benchmark. Specifically, we compare it with iSQRT, a representative covariance matrix-based pooling method, and GM-SOP, which shares a similar motivation with our approach. As shown in Table 3, applying LSOP leads to lower top-1 and top-5 error rates compared to the iSQRT baseline. This demonstrates that our method is effective not only on fine-grained datasets but also on large-scale classification tasks. Moreover, our approach outperforms GM-SOP, highlighting the importance of explicit cluster mining in localized second-order pooling frameworks.

Table 4: Accuracy comparison (top-1%) with SOTA FGVC related methods. All the results are quote from the original papers. iSQRT (Li et al., 2018) is selected are the base second-order pooling approach.

Method	Backbone	Birds	Cars	Airplane
TransFG (He et al., 2022)	ViT-B_16	91.7	94.8	-
IELT (Xu et al., 2023)	ViT-B_16	91.8	-	-
FRe-Net (Zhao et al., 2023)	DenseNet-161	89.9	95.1	94.2
FET (Chen et al., 2024)	SwinT-B	92.9	95.9	95.3
SwinFG (Ma et al., 2024)	SwinT-B	**93.3**	96.5	95.2
I2-HOFT (Sikdar et al., 2025)	ResNet50+GNN	91.6	**96.9**	**96.4**
iSQRT+LSOP (ours)	SwintT-B	91.4	95.3	95.1

4.5 Comparison with SOTA FGVC Related Methods

Although fine-grained visual classification (FGVC) is often cited as a representative application motivating second-order pooling, our framework is not specifically tailored for this task. Nevertheless, to comprehensively assess its potential, we conduct additional experiments comparing our method with the state-of-the-art methods designed explicitly for FGVC. To ensure a fair and consistent comparison, we re-implement our LSOP framework strictly following the training settings adopted by existing FGVC state-of-the-art methods, which differ from the settings used in our earlier experiments. Specifically, we evaluate our approach under the same configurations used by TransFG (He et al., 2022), IELT (Xu et al., 2023), FRe-Net (Zhao et al., 2023), FET (Chen et al., 2024), SwinFG (Ma et al., 2024), and I2-HOFT (Sikdar et al., 2025).

As summarized in Table 4, despite not being specifically tailored for FGVC, the proposed LSOP method attains competitive performance on multiple benchmark FGVC datasets when trained using stronger pre-trained backbones and the same training protocols as competing methods. In contrast, many existing FGVC approaches are explicitly designed with task-specific mechanisms—such as region-proposal modules, GNN-based relational reasoning, or multi-stage feature extractors—to capture fine-grained cues. These specialized components, while effective for FGVC, often increase architectural complexity and reduce generality. The strong results achieved by our simple and general-purpose framework—without relying on FGVC-specific designs—further validate both the effectiveness and the applicability of LSOP for visual recognition tasks.

4.6 Ablation Study

In this section, we conduct ablation studies to investigate several main properties of our method. In particular, we first demonstrate the impact of two key hyperparameter used in our framework, which are β in eq. (5) and the number of cluster K. Moreover, we provide more studies including the accuracy and computational overhead against the number of clusters, detail analysis on the sample-adaptive feature fusion scheme, more visual examples with attention maps in the supplementary material.

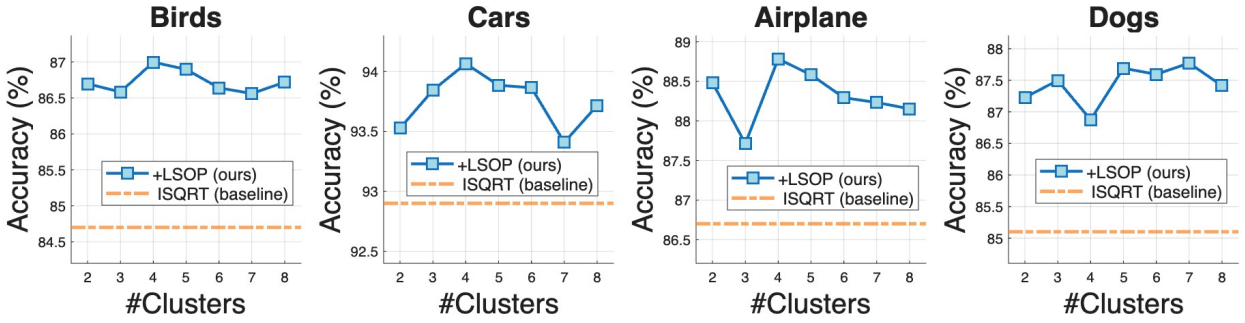


Figure 2: Impact of the number of clusters on classification performance across four FGVC datasets.

4.6.1 On the Number of Clusters K

As seen in Figure 2, while the results do not exhibit a consistent or monotonic trend, we observe that extreme settings—either too few or too many clusters—tend to yield slightly inferior performance. In contrast, moderate cluster numbers generally lead to better results. Nevertheless, all settings consistently outperform the baseline by a clear margin, indicating that our method is robust to the choice of cluster granularity.

4.6.2 On the Selection of β

A similar observation can be made based on the results shown in Figure 3. Although no clear trend emerges as the regularization coefficient β varies, our method consistently outperforms the baseline by a substantial margin across a wide range of β values. This further confirms the robustness of our approach with respect to the choice of the regularization strength.

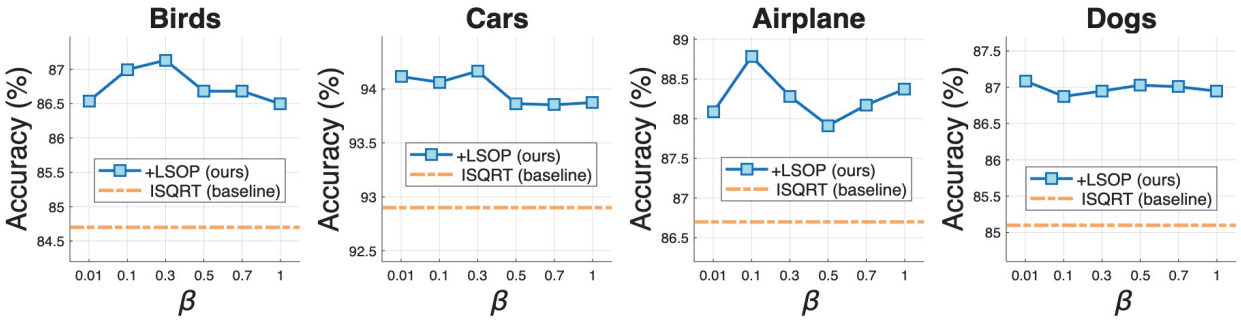


Figure 3: Impact of regularization coefficient β on classification performance across four FGVC datasets.

5 Conclusion

In this paper, we present an efficient end-to-end framework for localized second-order pooling, where covariance is computed within explicitly mined clusters. By integrating a dedicated cluster-mining branch with a sample-adaptive feature fusion strategy, our method overcomes key limitations of existing local second-order approaches. Experiments on multiple fine-grained and large-scale benchmarks show that LSOP achieves strong accuracy–efficiency trade-offs while outperforming prior methods. As future work, we aim to extend the proposed localized pooling mechanism to intermediate network stages.

References

Daniel A. Brooks, Olivier Schwander, Frédéric Barbaresco, Jean-Yves Schneider, and Matthieu Cord. Riemannian batch normalization for SPD neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 15463–15474, 2019. URL `https://proceedings.neurips.cc/paper/2019/hash/6e69ebbfad976d4637bb4b39de261bf7-Abstract.html`.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV (14)*, volume 11218 of *Lecture Notes in Computer Science*, pp. 139–156. Springer, 2018.

Huazhen Chen, Haimiao Zhang, Chang Liu, Jianpeng An, Zhongke Gao, and Jun Qiu. FET-FGVC: feature-enhanced transformer for fine-grained visual classification. *Pattern Recognit.*, 149:110265, 2024.

Yin Cui, Feng Zhou, Jiang Wang, Xiao Liu, Yuanqing Lin, and Serge J. Belongie. Kernel pooling for convolutional neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 3049–3058. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.325. URL `https://doi.org/10.1109/CVPR.2017.325`.

Jindou Dai, Yuwei Wu, Zhi Gao, and Yunde Jia. Infinite-dimensional feature aggregation via a factorized bilinear model. *Pattern Recognit.*, 124:108397, 2022.

E Dataset. Novel datasets for fine-grained image categorization. In *First Workshop on Fine Grained Visual Categorization, CVPR. Citeseer. Citeseer*. Citeseer, 2011.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255. IEEE Computer Society, 2009.

Melih Engin, Lei Wang, Luping Zhou, and Xinwang Liu. Deepkspd: Learning kernel-matrix-based SPD representation for fine-grained image recognition. In *ECCV (2)*, volume 11206 of *Lecture Notes in Computer Science*, pp. 629–645. Springer, 2018.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pp. 12873–12883. Computer Vision Foundation / IEEE, 2021.

Yang Gao, Oscar Beijbom, Ning Zhang, and Trevor Darrell. Compact bilinear pooling. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 317–326. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.41. URL https://doi.org/10.1109/CVPR.2016.41.

Zilin Gao, Jiangtao Xie, Qilong Wang, and Peihua Li. Global second-order pooling convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 3024–3033. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00314. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Gao_Global_Second-Order_Pooling_Convolutional_Networks_CVPR_2019_paper.html.

Ju He, Jieneng Chen, Shuai Liu, Adam Kortylewski, Cheng Yang, Yutong Bai, and Changhu Wang. Transfg: A transformer architecture for fine-grained recognition. In *AAAI*, pp. 852–860. AAAI Press, 2022.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pp. 770–778. IEEE Computer Society, 2016.

Zhiwu Huang and Luc Van Gool. A riemannian network for SPD matrix learning. In Satinder Singh and Shaul Markovitch (eds.), *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pp. 2036–2042. AAAI Press, 2017. doi: 10.1609/AAAI.V31I1.10866. URL https://doi.org/10.1609/aaai.v31i1.10866.

Catalin Ionescu, Orestis Vantzos, and Cristian Sminchisescu. Matrix backpropagation for deep networks with structured layers. In *ICCV*, pp. 2965–2973. IEEE Computer Society, 2015.

Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pp. 3304–3311. IEEE Computer Society, 2010.

Shu Kong and Charless C. Fowlkes. Low-rank bilinear pooling for fine-grained classification. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 7025–7034. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.743. URL https://doi.org/10.1109/CVPR.2017.743.

Piotr Koniusz, Hongguang Zhang, and Fatih Porikli. A deeper look at power normalizations. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 5774–5783. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00605. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Koniusz_A_Deeper_Look_CVPR_2018_paper.html.

Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *ICCV Workshops*, pp. 554–561. IEEE Computer Society, 2013.

Peihua Li, Jiangtao Xie, Qilong Wang, and Wangmeng Zuo. Is second-order information helpful for large-scale visual recognition? In *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 2089–2097. IEEE Computer Society, 2017a. doi: 10.1109/ICCV.2017.228. URL https://doi.org/10.1109/ICCV.2017.228.

Peihua Li, Hui Zeng, Qilong Wang, Simon C. K. Shiu, and Lei Zhang. High-order local pooling and encoding gaussians over a dictionary of gaussians. *IEEE Trans. Image Process.*, 26(7):3372–3384, 2017b.

Peihua Li, Jiangtao Xie, Qilong Wang, and Zilin Gao. Towards faster training of global covariance pooling networks by iterative matrix square root normalization. In *CVPR*, pp. 947–955. Computer Vision Foundation / IEEE Computer Society, 2018.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. Bilinear CNN models for fine-grained visual recognition. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pp. 1449–1457. IEEE Computer Society, 2015. doi: 10.1109/ICCV.2015.170. URL https://doi.org/10.1109/ICCV.2015.170.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 9992–10002. IEEE, 2021.

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. In *NeurIPS*, 2020.

Zhipeng Ma, Xiaoyu Wu, Anzhuo Chu, Lei Huang, and Zhiqiang Wei. Swinfg: A fine-grained recognition scheme based on swin transformer. *Expert Syst. Appl.*, 244:123021, 2024.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *CoRR*, abs/1306.5151, 2013. URL http://arxiv.org/abs/1306.5151.

Xuan Son Nguyen, Luc Brun, Olivier Lezoray, and Sébastien Bougleux. A neural network based on SPD manifold learning for skeleton-based hand gesture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 12036–12045. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.01231. URL http://openaccess.thecvf.com/content_CVPR_2019/html/Nguyen_A_Neural_Network_Based_on_SPD_Manifold_Learning_for_Skeleton-Based_CVPR_2019_paper.html.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024.

Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 413–420. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206537. URL https://doi.org/10.1109/CVPR.2009.5206537.

Saimunur Rahman and Peyman Moghadam. A deeper look into second-order feature aggregation for lidar place recognition. *arXiv preprint arXiv:2409.15919*, 2024.

Saimunur Rahman, Lei Wang, Changming Sun, and Luping Zhou. Redro: Efficiently learning large-sized SPD visual representation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XV*, volume 12360 of *Lecture Notes in Computer Science*, pp. 1–17. Springer, 2020. doi: 10.1007/978-3-030-58555-6_1. URL https://doi.org/10.1007/978-3-030-58555-6_1.

Saimunur Rahman, Piotr Koniusz, Lei Wang, Luping Zhou, Peyman Moghadam, and Changming Sun. Learning partial correlation based deep visual representation for image classification. In *CVPR*, pp. 6231–6240. IEEE, 2023.

Arindam Sikdar, Yonghuai Liu, Siddhardha Kedarisetty, Yitian Zhao, Amr Ahmed, and Ardhendu Behera. Interweaving insights: High-order feature interaction for fine-grained visual recognition. *Int. J. Comput. Vis.*, 133(4):1755–1779, 2025.

Yue Song, Nicu Sebe, and Wei Wang. Why approximate matrix square root outperforms accurate SVD in global covariance pooling? In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 1095–1103. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00115. URL https://doi.org/10.1109/ICCV48922.2021.00115.

Yue Song, Nicu Sebe, and Wei Wang. On the eigenvalues of global covariance pooling for fine-grained visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(3):3554–3566, 2023a. doi: 10.1109/TPAMI.2022.3178802. URL https://doi.org/10.1109/TPAMI.2022.3178802.

Yue Song, Nicu Sebe, and Wei Wang. Fast differentiable matrix square root and inverse square root. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(6):7367–7380, 2023b. doi: 10.1109/TPAMI.2022.3216339. URL https://doi.org/10.1109/TPAMI.2022.3216339.

Vladislav Sovrasov. ptflops: a flops counting tool for neural networks in pytorch framework, 2018-2024. URL https://github.com/sovrasov/flops-counter.pytorch.

Qiule Sun, Zhimin Zhang, and Peihua Li. Second-order encoding networks for semantic segmentation. *Neurocomputing*, 445:50–60, 2021.

Oncel Tuzel, Fatih Porikli, and Peter Meer. Region covariance: A fast descriptor for detection and classification. In Ales Leonardis, Horst Bischof, and Axel Pinz (eds.), *Computer Vision - ECCV 2006, 9th European Conference on Computer Vision, Graz, Austria, May 7-13, 2006, Proceedings, Part II*, volume 3952 of *Lecture Notes in Computer Science*, pp. 589–600. Springer, 2006. doi: 10.1007/11744047_45. URL https://doi.org/10.1007/11744047_45.

Oncel Tuzel, Fatih Porikli, and Peter Meer. Human detection via classification on riemannian manifolds. In *2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA*. IEEE Computer Society, 2007. doi: 10.1109/CVPR.2007.383197. URL https://doi.org/10.1109/CVPR.2007.383197.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *NIPS*, pp. 6306–6315, 2017.

Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

Qilong Wang, Zilin Gao, Jiangtao Xie, Wangmeng Zuo, and Peihua Li. Global gated mixture of second-order pooling for improving deep convolutional neural networks. In *NeurIPS*, pp. 1284–1293, 2018.

Qilong Wang, Jiangtao Xie, Wangmeng Zuo, Lei Zhang, and Peihua Li. Deep cnns meet global covariance pooling: Better representation and generalization. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(8):2582–2597, 2021. doi: 10.1109/TPAMI.2020.2974833. URL https://doi.org/10.1109/TPAMI.2020.2974833.

Qilong Wang, Mingze Gao, Zhaolin Zhang, Jiangtao Xie, Peihua Li, and Qinghua Hu. Dropcov: A simple yet effective method for improving deep architectures. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/d9888cc7baa04c2e44e8115588133515-Abstract-Conference.html.

Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. Learning super-features for image retrieval. In *ICLR*. OpenReview.net, 2022.

Xin Wen, Bingchen Zhao, Anlin Zheng, Xiangyu Zhang, and Xiaojuan Qi. Self-supervised visual representation learning with semantic grouping. In *NeurIPS*, 2022.

Qin Xu, Jiahui Wang, Bo Jiang, and Bin Luo. Fine-grained visual classification via internal ensemble learning transformer. *IEEE Trans. Multim.*, 25:9015–9028, 2023.

Kaicheng Yu and Mathieu Salzmann. Second-order convolutional neural networks. *CoRR*, abs/1703.06817, 2017. URL http://arxiv.org/abs/1703.06817.

Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell D. Collins, Yukun Zhu, Hartwig Adam, Alan L. Yuille, and Liang-Chieh Chen. k-means mask transformer. In *ECCV (29)*, volume 13689 of *Lecture Notes in Computer Science*, pp. 288–307. Springer, 2022a.

Tan Yu, Yunfeng Cai, and Ping Li. Toward faster and simpler matrix normalization via rank-1 update. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XIX*, volume 12364 of *Lecture Notes in Computer Science*, pp. 203–219. Springer, 2020. doi: 10.1007/978-3-030-58529-7_13. URL https://doi.org/10.1007/978-3-030-58529-7_13.

Tan Yu, Xiaoyun Li, and Ping Li. Fast and compact bilinear pooling by shifted random maclaurin. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 3243–3251. AAAI Press, 2021. doi: 10.1609/AAAI.V35I4.16435. URL https://doi.org/10.1609/aaai.v35i4.16435.

Tan Yu, Yunfeng Cai, and Ping Li. Efficient compact bilinear pooling via kronecker product. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pp. 3170–3178. AAAI Press, 2022b. doi: 10.1609/AAAI.V36I3.20225. URL https://doi.org/10.1609/aaai.v36i3.20225.

Xiaohang Zhan, Jiahao Xie, Ziwei Liu, Yew-Soon Ong, and Chen Change Loy. Online deep clustering for unsupervised representation learning. In *CVPR*, pp. 6687–6696. Computer Vision Foundation / IEEE, 2020.

Shan Zhang, Dawei Luo, Lei Wang, and Piotr Koniusz. Few-shot object detection by second-order pooling. In *ACCV (4)*, volume 12625 of *Lecture Notes in Computer Science*, pp. 369–387. Springer, 2020.

Peng Zhao, Yi Li, Baowei Tang, Huiting Liu, and Sheng Yao. Feature relocation network for fine-grained image classification. *Neural Networks*, 161:306–317, 2023.

Heliang Zheng, Jianlong Fu, Zheng-Jun Zha, and Jiebo Luo. Learning deep bilinear transformation for fine-grained image representation. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 4279–4288, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/959ef477884b6ac2241b19ee4fb776ae-Abstract.html.

A Datasets

In this section, the description of the six datasets, Birds (Wah et al., 2011), Cars (Krause et al., 2013), Airplane (Maji et al., 2013), Dogs (Dataset, 2011), MIT (Quattoni & Torralba, 2009), and ImageNet-1K (Deng et al., 2009) used in the experiment are provided. Some example images randomly sampled from these datasets are shown in Figure 4.

Birds. As one of the most popular datasets for fine-grained visual classification tasks, it contains 11,788 images of 200 classes in total. The main difficulty lies in the subtle differences between bird species which are hard to distinguish by even human experts. This dataset provides the bounding box annotations for ground truth objects. However, we did not use this information at all.

Cars. This dataset has a total of 16,185 images of 196 car classes. The classes are categories concerning car production year, car model, and car manufacturer. Compared with the Birds dataset, the car images in this dataset have a more complex background.

Airplane. Similar to the above two datasets, the airplane dataset is also a widely used fine-grained visual classification dataset. It has 10,000 images in total coming from 100 aircraft classes. These images are uniformly assigned to the 100 classes.

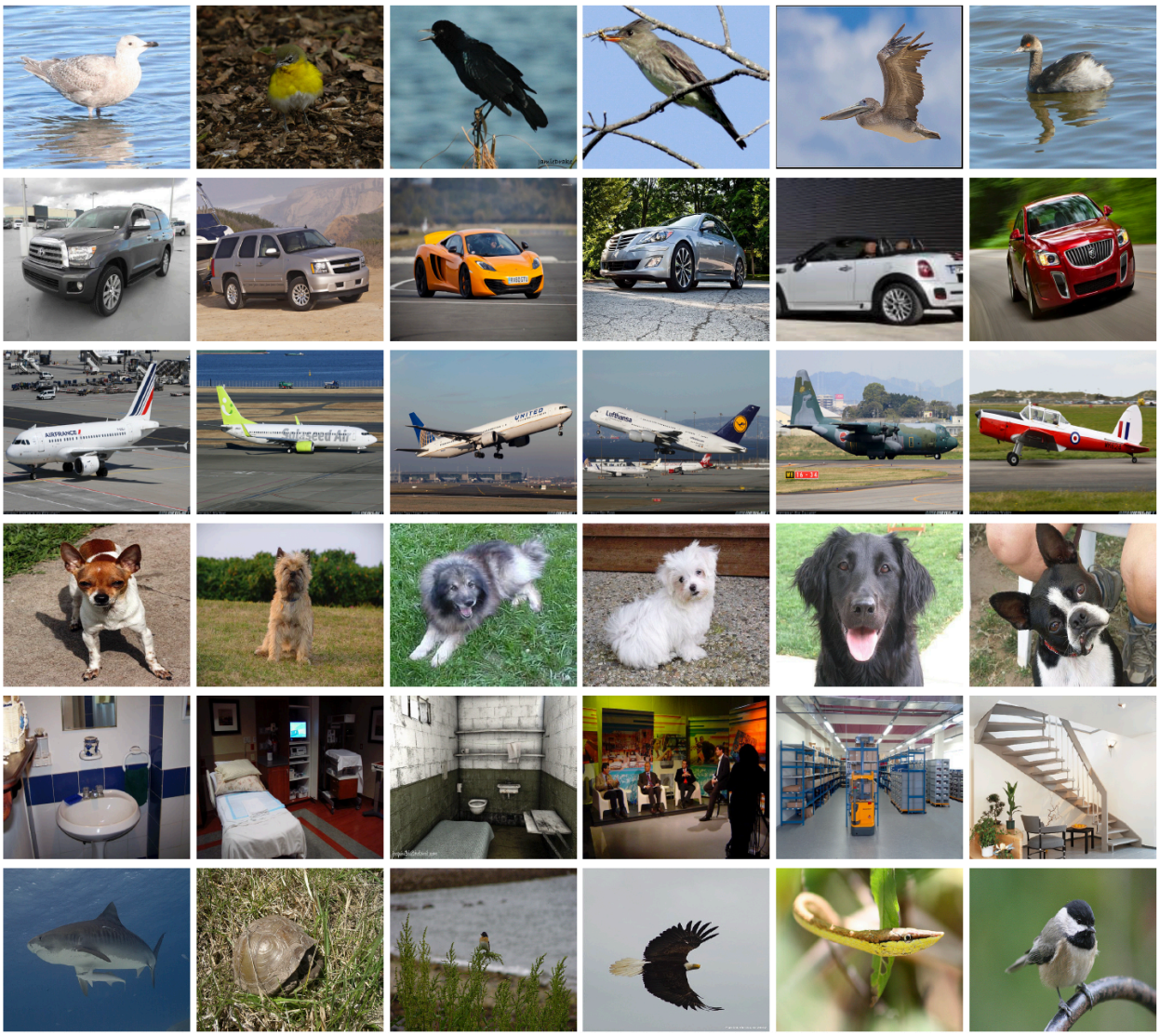


Figure 4: Images sampled from the datasets used in our experiments. Rows 1, 2, 3, 4, 5, and 6 have images from Birds, Cars, Airplane, Dogs, MIT, and ImageNet-1K, respectively.

Dogs. The Stanford Dogs dataset contains 20,580 images of 120 breeds of dogs from around the world. This dataset has been built using images and annotation from ImageNet for the task of fine-grained image categorization.

MIT. MIT is an indoor scene recognition dataset that has a total of 15,620 images from 67 classes. The main difficulty is that while some indoor scenes (*e.g.*, corridors) can be well characterized by global spatial properties, others (*e.g.*, bookstores) are better characterized by the objects they contain.

ImageNet-1K. ImageNet-1K is a landmark dataset with over 1.2 million images across 1,000 categories, used extensively for benchmarking image classification algorithms in deep learning. It has been instrumental in the evolution of convolutional neural networks by providing a diverse and challenging set of classes for models to learn from.

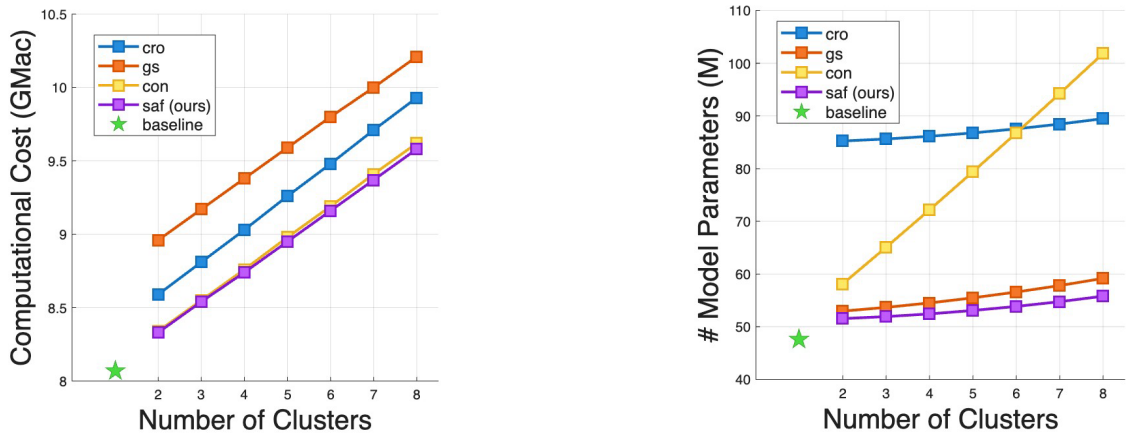


Figure 5: Computational overhead vs. the number of clusters when conducting iSQRT (Li et al., 2018) with ResNet101. The "baseline" denotes the original iSQRT.

B Computational Overhead w.r.t. the Number of Clusters

We demonstrate the impact of the number of clusters on the computational overhead by applying our method with a set of different cluster numbers (i.e., 2, 3, 4, 5, 6, 7, and 8). As seen in Figure 5, as the number of clusters increases, both the computational complexity evaluated by GMac (on the left side) and the number of parameters (on the right side) increases accordingly. This is expected since the proposed method needs to conduct the matrix normalization process multiple times corresponding to the number of local covariance matrices. However, compared to the three other feature fusion schemes which are concatenation (denoted as “con”), cross-attention (denoted as “cro”), and gated selection (denoted as “gs”), the proposed sample-adaptive feature fusion dramatically saves the extra computational cost on both complexity and the number of parameters perspectives[2].

C Illustration of Local Descriptor Clusters on Example Images

We provide some visual examples of the attention maps generated by our attention-based cluster mining block. As shown in Figure 6, 7, and 8, the first column displays the raw images and the following columns provide the attention maps calculated with the soft assignment weights in eq. (1) in the main text. In particular, the position in red denotes the value there is high and the position in blue denotes the opposite. As seen, the attention maps corresponding to the same cluster (presented in the same column) consistently focus on the same semantic pattern across different images within a dataset. For example, the first cluster consistently pays more attention to the front of the cars and the last cluster usually focuses more on the background no matter what car types are in the images in the Cars dataset (first line in Figure 6). The same result could be observed from the images of the other datasets. This result demonstrates that our attention-based cluster mining branch effectively mines the underlying semantic information and generates reliable clusters for the dataset.

D Further Demonstration on Impact of Number of Cluster

E Visualization on the Clustering

F t-SNE on Various Number of K

[2]Note the weighted sum prediction (denoted as “wsp” in the main text) is not displayed since it shares the same computational complexity with our fusion scheme.

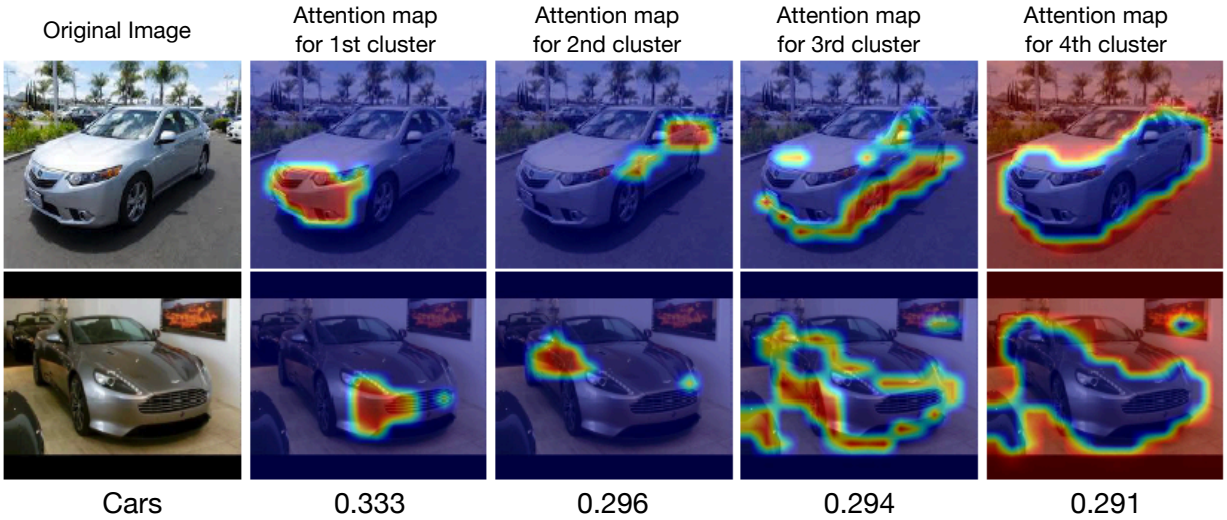


Figure 6: Visualization of the end-to-end learned clusters (K=4) on images randomly sampled from the Cars dataset. The first column shows the original images, while the following four columns present the attention maps corresponding to the first, second, third, and fourth clusters, respectively.

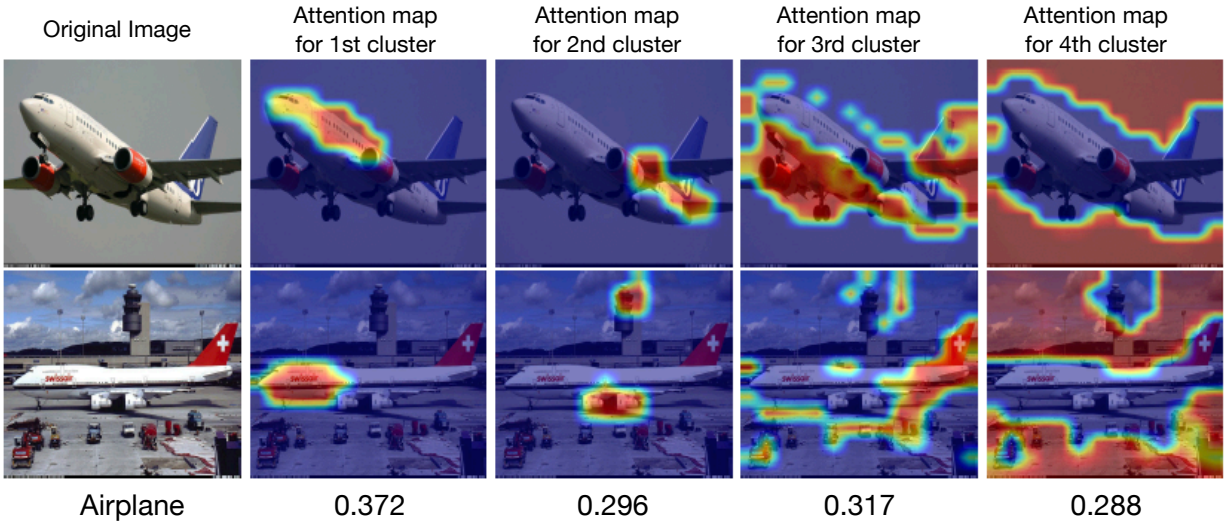


Figure 7: Visualization of the end-to-end learned clusters (K=4) on images randomly sampled from the Airplane dataset. The first column shows the original images, while the following four columns present the attention maps corresponding to the first, second, third, and fourth clusters, respectively.

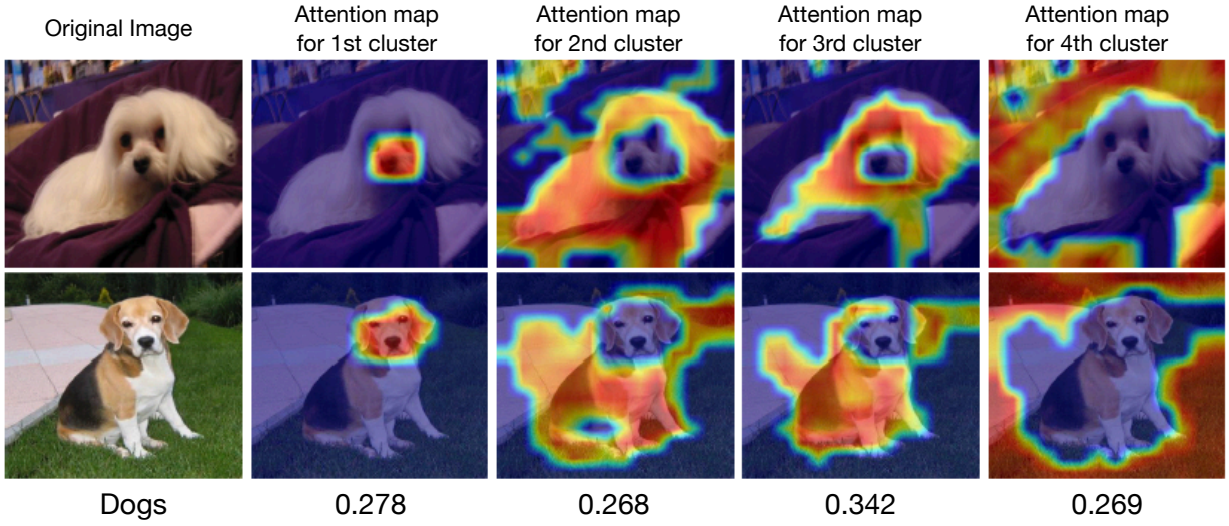


Figure 8: Visualization of the end-to-end learned clusters (K=4) on images randomly sampled from the Dogs dataset. The first column shows the original images, while the following four columns present the attention maps corresponding to the first, second, third, and fourth clusters, respectively.

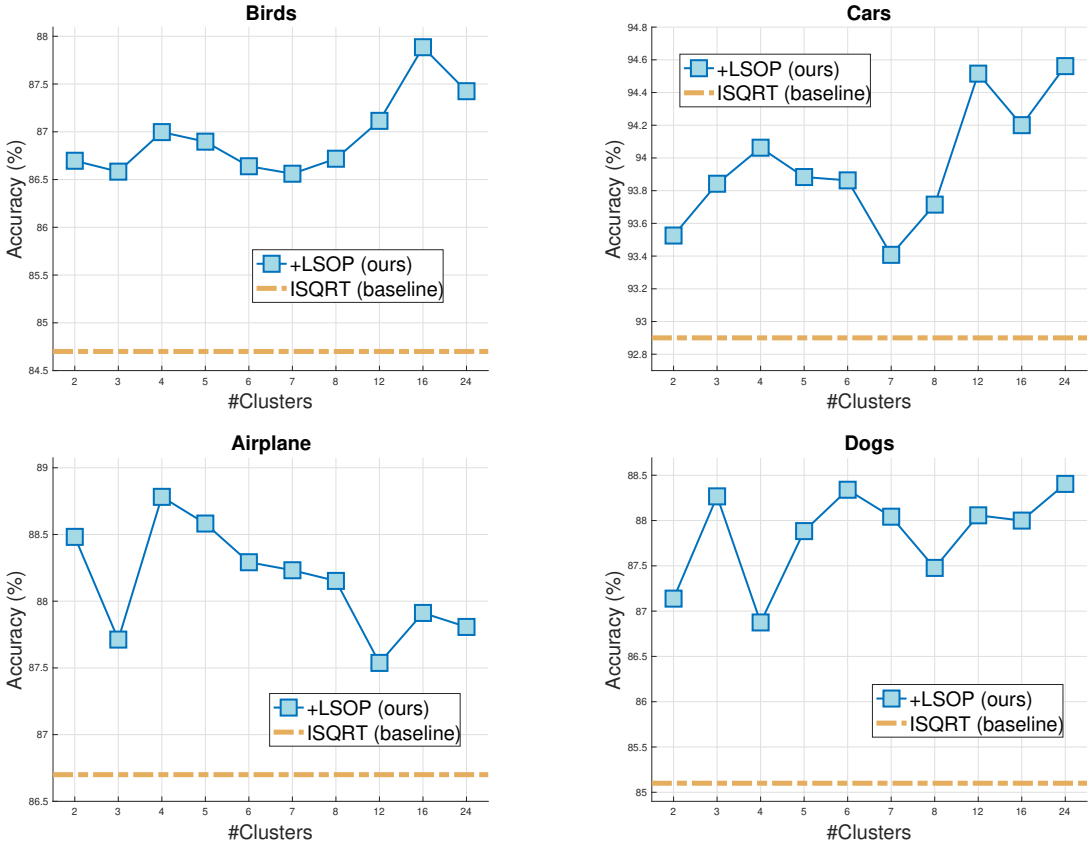
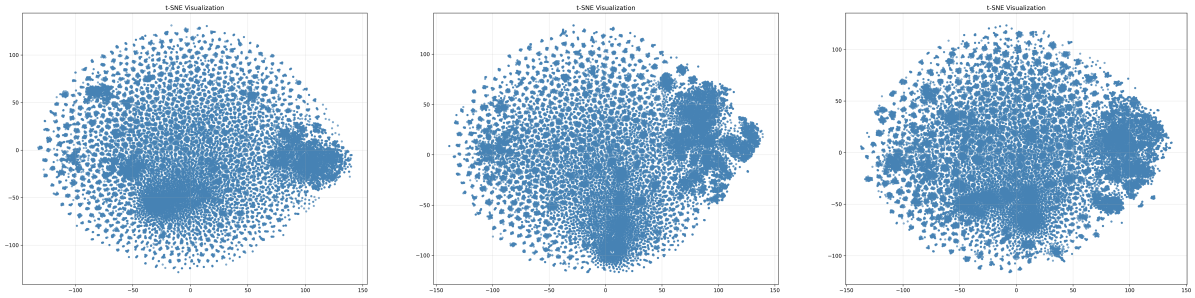
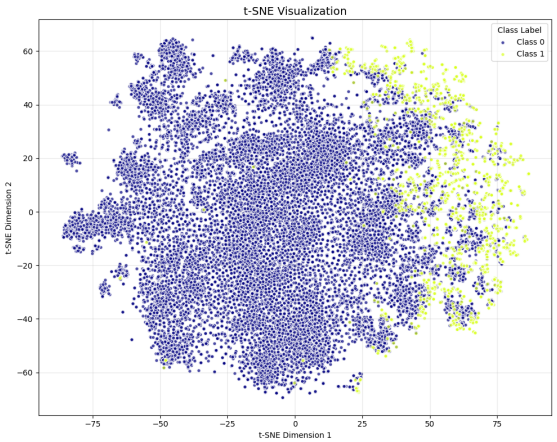


Figure 9: Impact of the number of clusters on classification performance.

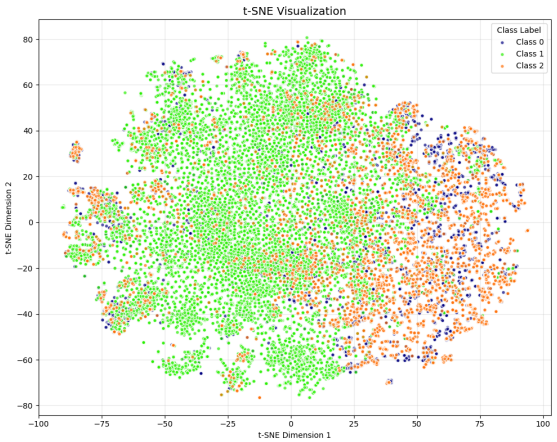


Birds Cars Airplane

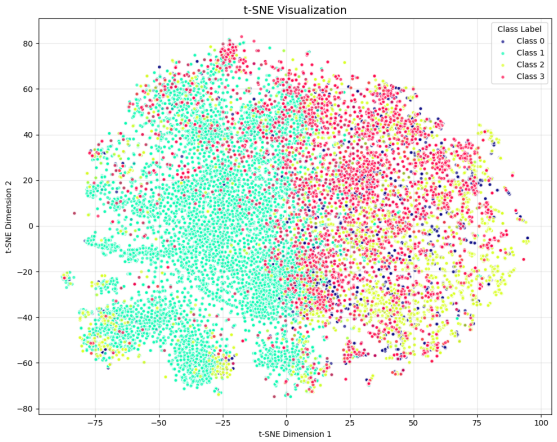
Figure 10: t-SNE for dataset Birds, Cars, and Airplane.



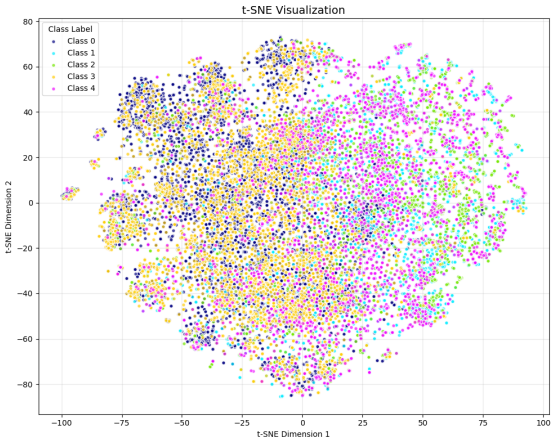
K = 2 (Accuracy: 86.7)



K = 3 (Accuracy: 86.6)



K = 4 (Accuracy: 87.0)



K = 5 (Accuracy: 86.9)

Figure 11: t-SNE for local descriptors with various K selection on Birds.