Multi-Manifold Attention for Vision Transformers

DIMITRIOS KONSTANTINIDIS, ILIAS PAPASTRATIS, KOSMAS DIMITROPOULOS, AND PETROS DARAS
Information Technologies Institute, Centre for Research and Technology Hellas (CERTH), 57001 Thessaloniki, Greece
Corresponding author: Dimitrios Konstantinidis (dikonsta@iti.gr)

This work was supported by the European Commission’s Horizon Europe project “DAFNEplus: Decentralized platform for fair creative content distribution empowering creators and communities through new digital distribution models based on digital tokens” under Grant 101061548.

ABSTRACT Vision Transformers are very popular nowadays due to their state-of-the-art performance in several computer vision tasks, such as image classification and action recognition. Although their performance has been greatly enhanced through highly descriptive patch embeddings and hierarchical structures, there is still limited research on utilizing additional data representations so as to refine the self-attention map of a Transformer. To address this problem, a novel attention mechanism, called multi-manifold multi-head attention, is proposed in this work to substitute the vanilla self-attention of a Transformer. The proposed mechanism models the input space in three distinct manifolds, namely Euclidean, Symmetric Positive Definite and Grassmann, thus leveraging different statistical and geometrical properties of the input for the computation of a highly descriptive attention map. In this way, the proposed attention mechanism can guide a Vision Transformer to become more attentive towards important appearance, color and texture features of an image, leading to improved classification and segmentation results, as shown by the experimental results on well-known datasets.

INDEX TERMS Attention, manifold, vision transformers, image classification, semantic segmentation.

I. INTRODUCTION
Transformer networks have been met with great interest from the research community when they were originally proposed for natural language processing in the pioneering work of [1]. A Transformer is a network architecture that relies on self-attention, which is an attention mechanism that relates different positions of a single sequence in order to compute a new representation of the sequence. Transformers are designed to handle sequential input data, however, unlike Recurrent Neural Networks (RNNs), they process the data in parallel since they receive the entire sequence as input. In this way, Transformers can extract both short- and long-term dependencies between input and output, while simultaneously achieving increased parallelization and training speed with respect to RNNs.

Recently, Vision Transformers (ViTs) [2] have been introduced for computer vision tasks, leading to state-of-the-art performance in well-known benchmark datasets due to their ability to model short- and long-range relationships between different image areas. However, ViTs lack the local inductive biases of Convolutional Neural Networks (CNNs) and do not efficiently model local information [3].
To overcome such issues, recent works aim to increase local structure modelling by introducing convolutions to the ViTs [3], [4], [5], redesign the patch tokenization process and introduce local attention mechanisms [6], [7], [8], [9] or adopt hierarchical structures similar to CNNs [6], [10], [11].

However, most ViTs operate only in the Euclidean space of pixel intensity values, overlooking the fact that data representations in other manifolds can be beneficial to their performance. Additionally, the attention maps tend to be similar in the deeper layers of ViTs and their representation stops improving [12], thus a need for more discriminative attention maps is imperative. To this end, this work proposes multi-manifold multi-head attention (MMA) that can be introduced to any ViT to improve the discriminative power of its attention map. The main advantage of MMA over the vanilla self-attention is the use of three different manifolds, namely Euclidean, Symmetric Positive Definite (SPD) and Grassmann to model the input. Computing feature representations and distances in manifolds with different statistical and geometrical properties enables a better modelling of the input through the computation of an attention map that can better enhance the important features of the input for a given computer vision task, as shown in Fig. 1. More specifically, the contributions of this work are the following:

- A novel self-attention mechanism, called MMA, is proposed to accurately model the underlying structure of the input space through its transformation into feature representations in three distinct manifolds.
- The proposed MMA can be added to any Vision Transformer to better guide its attention towards important appearance, color and texture features of an image.
- Extensive experimentation with various ViTs and on different image classification and semantic segmentation datasets verifies the performance improvements achieved with the proposed MMA.

II. RELATED WORK

A. VISION TRANSFORMERS

Dosovitskiy et al., in [2], were the first to propose ViT as a pure Transformer backbone for image classification. In their work, the input image is segmented into a series of non-overlapping image patches that are then projected into a linear embedding sequence. This sequence is concatenated with a learnable positional encoding that holds information on the spatial location of patches in the image before being fed to the Transformer for classification.

Since then, ViTs have been heavily modified to achieve improved performance on computer vision tasks [13]. Several ViTs combine CNNs with Transformers to effectively leverage the convolutional biases in images and enhance accuracy. Based on empirical observations that CNNs are better teachers than Transformers, Touvron et al. employed a teacher-student strategy to transfer the inductive bias of the CNN teacher to the Transformer student through knowledge distillation [14]. The authors in [3] proposed ConViT, in which a parallel convolution branch was attached to the Transformer branch to impose convolutional inductive biases via a Gated Positional Self-Attention that approximates the locality of convolutional layers. Yuan et al. proposed the Convolution-enhanced Image Transformer that uses a convolutional module to extract patch embeddings from low-level features and a layer-wise attention to model long-range dependencies [5].

To improve local attention and enhance the feature extraction capabilities of ViTs, Han et al. combined Transformer networks at patch and pixel levels to produce better representations with rich local and global information [15]. Meanwhile, the CSWin Transformer [9] achieves strong modeling capabilities by performing self-attention in horizontal and vertical stripes in parallel, while A-ViT [16] adaptively adjusts the inference cost of ViTs for images of different complexity by reducing the number of processed tokens as inference proceeds. In a similar fashion, the authors in [17] combined locally-grouped self-attention and global sub-sampled attention in a Vision Transformer, achieving competitive image classification and object detection results. To further reduce computational cost, several works approximated the quadratic operations in self-attention by low-rank matrices [18], positive orthogonal random features [19], locality-sensitive hashing [20] and memory-efficient coupling attention maps [21]. On the other hand, the authors in [22] replaced the attention mechanism with a spatial pooling operation, demonstrating that the resulted MetaFormer can achieve state-of-the-art performance in several computer vision tasks.

Realizing that ViTs need sufficiently large datasets to perform well, Hassani et al. proposed three compact ViT architectures, namely ViT-Lite, Compact Vision Transformers (CVTs) and Compact Convolutional Transformers (CCTs) [4]. ViT-Lite is similar to the original ViT but with smaller patch sizing suitable for small datasets. CVTs expand on ViT-Lite by pooling the sequential information from the Transformer, eliminating the need for the extra classification token. Finally, CCTs further expand on CVTs by adding convolutional blocks to the tokenization step, thus preserving local information while encoding relationships between patches, unlike the original ViT.

Recognizing that a fixed resolution across the entire network neglects fine-grained information and brings heavy computational costs, several works proposed hierarchical structures for ViTs. Yuan et al. proposed the hierarchical structuring of the patch embeddings through the combination of neighbouring embeddings into a single one [6]. Similarly, the Token Pyramid Vision Transformer produces scale-aware semantic features by processing tokens from various scales into a single token to augment the representation [23]. Other ViTs reduce the spatial dimensions of the output progressively, similarly to CNNs, using spatial-reduction attention or pooling layer [10], [11]. The Swin Transformer employs shifting windows in each layer to create hierarchical global and boundary features through cross-window
A manifold is a topological space that locally resembles a Euclidean space without considering how different deep layers. Traditionally, ViTs process raw pixel intensities directly in the Euclidean space without considering how different data representations may affect their performance. The proposed work improves local attention through the use of feature representations in different manifolds to create more descriptive attention maps.

B. MANIFOLD BACKGROUND
A manifold is a topological space that locally resembles a Euclidean space near each point [29]. Essentially, a manifold is a mapping from one space to another, allowing similar features to appear closer to each other, while dissimilar features move further apart. Manifolds have been widely employed in computer vision tasks due to their ability to model different and complementary statistical and geometrical properties of features that can be beneficial for a given task [23], [30], [31]. Two widely employed special types of manifolds used to describe image sets and videos in the literature are the SPD and Grassmann manifolds.

Huang et al. utilized properties of the Riemannian geometry on manifolds and proposed a new similarity method based on SPD features for clustering tasks [30]. Yu et al. proposed the contour covariance that lies on the SPD manifold as a region descriptor for accurate image classification [32]. Similarly, Chu et al. proposed the modelling of image sets with covariance matrices for improved classification performance [33]. The importance and usefulness of feature modelling on the SPD manifold can be further highlighted from the design of novel deep networks and network layers, such as Variational Autoencoders [34], LSTMs [35], GRUs [36] and mapping and pooling layers [37] to handle and learn from features on the SPD manifold. The main drawback of employing covariance features in a deep learning framework is the non-linearity of the SPD manifold that introduces challenging optimization problems [35].

On the other hand, Grassmannian geometry has been widely employed in several computer vision tasks, such as image classification and action recognition. Data representations that lie on the Grassmann manifold include the Vectors of Locally Aggregated Descriptors (VLAD) that have been successfully employed for medical image classification [38] and the Linear Dynamic System (LDS) features that have been employed for fire and smoke detection in video sequences [39] and skeletal sequence modelling for sign language and action recognition [40], [41]. In a different approach, Dimou et al. introduced LDS features in a ResNet architecture, achieving state-of-the-art performance in image classification [42]. Finally, there are works on the clustering of image sets modelled in high-dimensional Grassmann manifold through the projection to a low-dimensional space using an unsupervised dimensionality reduction algorithm based on Neighborhood Preserving Embeddings [43] or a new low rank approximation model that relies on the Double Nuclear norm [44].

To further leverage the discriminative power of feature representations in different manifolds, Wang et al. fused SPD and Grassmann manifold representations for clustering purposes, achieving state-of-the-art performance [45]. However, current ViTs have largely overlooked alternative data representations, typically focusing only on the Euclidean space of pixel intensities. To address this issue, this work proposes MMA that combines feature representations in three distinctive manifolds to learn a highly descriptive attention map that can better identify the important context of input images. By leveraging the statistical properties of different manifolds, MMA can guide any Vision Transformer to better model the underlying structure of the input space, leading to improved classification and segmentation results.

III. PROPOSED METHOD
Inspired by the need for more descriptive attention maps, this work proposes MMA as a novel self-attention mechanism that leverages the statistical and geometrical properties of different manifolds to produce richer feature representations. In contrast to the vanilla self-attention mechanism, MMA projects the high-dimensional Euclidean space to the Grassmann and SPD manifolds and computes the respective distance maps on them, as shown in Fig. 1. Then, a fused attention map is produced through the merging of the Euclidean, SPD and Grassmannian distance maps that is more attentive towards important appearance, color and texture features of the image, ultimately leading to more accurate results in different computer vision tasks.

A. PRELIMINARIES
Given an input image $I \in \mathbb{R}^{H \times W \times C}$ with height $H$, width $W$ and $C$ channels, a Vision Transformer divides the image into non-overlapping rectangular patches of size $P \times P$ and then
linearly project them into the space of the hidden dimension $D$ of the Transformer or employs a 2D convolution to perform both image patch extraction and linear projection. The result is a vector of patch embeddings $X \in \mathbb{R}^{L \times D}$, where $L = \frac{HW}{p^2}$ is the sequence length.

Position embeddings are then added to the patch embeddings to provide a sense of order in the input sequence, allowing ViT to model both the content of the patches and their spatial location with respect to other image patches. The most common position embeddings are either vectors with sine and cosine frequencies [2] or learned embeddings [3], [46].

The embeddings are finally fed to the self-attention mechanism that is the most important layer of a Transformer, responsible for computing an output representation that is more attentive towards important features of the input. Vaswani et al. proposed the multi-head self-attention (MHSA) that performs different linear projections of the input at different subspaces through parallel attention layers, called heads, concatenated together [47]. MHSA is computed as follows:

$$Q_i = X W_{iq}^i, \quad K_i = X W_{ik}^i, \quad V_i = X W_{iv}^i$$
$$S_i = \text{Attention}(Q_i, K_i, V_i), \quad i = 1, 2, \ldots, h$$
$$\text{MHSA}(Q, K, V) = \text{concat}(S_1, S_2, \ldots, S_h) W_o$$

where $h$ is the number of heads, $W_o \in \mathbb{R}^{D \times D}$ is the weight projection matrix, $d = D/h$ is the feature dimension in each head, $S_i \in \mathbb{R}^{L \times L}$ is the attention map of each head and $W_{iq}^i, W_{ik}^i, W_{iv}^i \in \mathbb{R}^{D \times d}$ are the weight matrices for the query, key and value vectors of each head $i$, respectively. The attention map of each head $S_i$ is equal to:

$$S_i = \text{softmax} \left( \frac{Q_i K_i^T}{\sqrt{d}} \right) V_i$$

MHSA allows ViT to jointly attend to information from different representation subspaces at different positions, enabling it to gather more positional data because each head focuses on various regions of the input and creating a more comprehensive representation after the combination of the vectors.

### B. MULTI-MANIFOLD MULTI-HEAD ATTENTION (MMA)

MMA can improve the performance of any ViT by replacing its vanilla self-attention mechanism. To achieve this, MMA transforms the input sequence to points in the Euclidean, SPD and Grassmann manifolds and computes distances between them in these manifolds in order to produce a fused attention map of high discriminative power, as shown in Fig. 2. Next, each manifold is described in detail, along with how the individual distance maps are computed and fused to form the refined attention map.

1) EUCLIDEAN MANIFOLD

Given query $Q \in \mathbb{R}^{L \times d}$ and key $K \in \mathbb{R}^{L \times d}$ vectors, the Euclidean distance map of MMA is computed similarly to a vanilla Vision Transformer as:

$$D_E(Q, K) = \frac{QK^T}{\sqrt{dk}}$$

The distance map $D_E \in \mathbb{R}^{h \times L \times L}$, with $h$ representing the number of heads, expresses the similarity between query and key vectors, with higher values denoting greater distances between the two vectors in the Euclidean manifold.

2) SPD MANIFOLD

The SPD manifold is a specific type of Riemann manifold composed of points expressed as square matrices $M$ of size.
where in the Grassmann manifold \( \mathbb{G} \), as a result, the model sequential and time-varying signals as any linear Grassmann manifold, denoted as \( \mathbb{G} \):

\[
S_{++} = \{ M \in \mathbb{R}^{d \times d} : u^T M u > 0 \forall u \in \mathbb{R}^d - \{0_d\} \} \tag{6}
\]

For a matrix to be considered as point in a SPD manifold, it should be symmetrical and have positive eigenvalues. Covariance matrices possess such properties and thus they can be considered points in a SPD manifold. Covariance matrices have been widely employed in the literature to model appearance and texture features in computer vision tasks [32], [48]. As a result, the inclusion of covariance matrices in the computation of MMA is beneficial to the performance of a Vision Transformer due to incorporating additional information about the input, enhancing the discriminative power of the output feature representation. Several metrics can be used to measure the distance between points in a SPD manifold [49]. However, this work employs the Frobenius power of the output feature representation. Several metrics can be used to measure the distance between points in a SPD manifold. The SPD distance between the ith covariance matrix of query and the jth covariance matrix of key is then calculated as:

\[
D_{SPD}(C_Q, C_K) = \frac{||C_Q - C_K||_F}{s} \tag{9}
\]

where \( || \cdot ||_F \) denotes the Frobenius norm. Similar to the Euclidean distance map, \( D_{SPD} \in \mathbb{R}^{h \times L \times L} \) expresses the similarity between query and key vectors and quantifies the distances between the two vectors in the SPD manifold.

3) GRASSMANN MANIFOLD

The Grassmann manifold is another well-known special type of Riemann manifold that embeds all \( p \)-dimensional linear subspaces that lie in a \( d \)-dimensional Euclidean space. The Grassmann manifold, denoted as \( \mathbb{G} \), can be represented by the set of orthogonal matrices from the orthogonal group \( \mathcal{O}(p) \) as follows:

\[
\mathbb{G} = \{ X \in \mathbb{R}^{d \times p} : X^T X = I_p \}/\mathcal{O}(p) \tag{10}
\]

where \( X \) represents any point on the Grassmann manifold. Grassmann manifolds have been commonly employed to model sequential and time-varying signals as any linear dynamic system can be easily transformed to a point in the Grassmann manifold [38], [40]. As a result, the transformation of the input space to points in the Grassmann manifold can provide to a Vision Transformer additional information regarding texture and color variations in an image patch, leading to enriched feature representations with high discriminative power.

Several metrics have been defined to measure the distance between Grassmanian points. The most common technique is to embed the manifold into the space of symmetric matrices with the mapping \( \Pi : \mathbb{G}(p, d) \to \text{Sym}(d), \Pi(X) = XX^T \), which is a one-to-one, continuous and differentiable mapping [50]. Moreover, to avoid the computation of coordinate transformations as well as to improve efficiency, the kernel form of the projection distance [51] is adopted.

Given query \( Q \in \mathbb{R}^{L \times d} \) and key \( K \in \mathbb{R}^{L \times d} \), the covariance matrices of (7) and (8) are first obtained. Since the orthogonality of the covariance matrices \( C_Q \) and \( C_K \) is not guaranteed, a transformation is required to ensure that the new matrices are orthogonal and thus represent points in the Grassmann manifold. One of the most popular techniques to achieve orthogonality is through QR decomposition [1]. However, this work employs the Cayley transform [52] due to its much smaller computational cost. According to this transform, the covariance matrices \( C_Q \) and \( C_K \) are transformed to the skew-symmetric matrices \( C_{Q}^{\text{skew}} \) and \( C_{K}^{\text{skew}} \) as follows:

\[
C_{Q}^{\text{skew}} = \frac{1}{2}(C_Q - C_Q^T) \tag{11}
\]
\[
C_{K}^{\text{skew}} = \frac{1}{2}(C_K - C_K^T) \tag{12}
\]

Then, the Cayley map is employed to the skew-symmetric matrices to produce the following matrices:

\[
G_Q = (I_s + \frac{C_{Q}^{\text{skew}}}{2})(I_s - \frac{C_{Q}^{\text{skew}}}{2})^{-1} \tag{13}
\]
\[
G_K = (I_s + \frac{C_{K}^{\text{skew}}}{2})(I_s - \frac{C_{K}^{\text{skew}}}{2})^{-1} \tag{14}
\]

The matrices \( G_Q, G_K \in \mathbb{R}^{L \times s \times s} \) correspond to the representation of the query and key vectors in the Grassmann manifold and describe clusters of \( L \) orthogonal matrices that lie on the manifold. Finally, the projection distance is employed to calculate the scaled distance between the i-th orthogonal matrix of query and the j-th orthogonal matrix of key in the Grassmann manifold as follows:

\[
D_{G}^{(i)}(G_Q, G_K) = \frac{||G_Q^{(i)} - G_K^{(j)}||_F}{s} \tag{15}
\]

where \( || \cdot ||_F \) denotes the Frobenius norm. As with the other manifold distance maps, \( D_G \in \mathbb{R}^{h \times L \times L} \) expresses the similarity between query and key vectors and quantifies the distances between the two vectors in the Grassmann manifold.

4) FUSION OF MANIFOLDS

After the computation of the individual distance maps \( D_E \), \( D_{SPD} \) and \( D_G \in \mathbb{R}^{h \times L \times L} \) in each manifold, two setups,
denoted as early and late fusion, are proposed to derive the output feature representation.

\( a: \text{EARLY FUSION} \)

In this setup, the three distance maps are concatenated together and then a 3D convolution operation is employed to learn an element-wise weight matrix to perform an effective mapping of the distances in the different manifolds and generate the output feature representation. More specifically, the three distance maps are concatenated together forming the map:

\[
D_{\text{concat}} = \text{BN}(\text{concat}(D_E, D_{\text{SPD}}, D_G)) \in \mathbb{R}^{3 \times L \times L}
\]  

(16)

with BN denoting batch normalization. Afterwards, a weight matrix is computed to merge the different distance maps in an optimal way:

\[
W_D = \text{softmax}(\text{LeakyRELU}(\rho(D_{\text{concat}}))) \in \mathbb{R}^{3 \times L \times L}
\]  

(17)

The weight matrix \( W_D \) is responsible for computing appropriate weights in an element-wise manner to accurately merge the different distances. Finally, the output feature representation \( V' \) is computed as follows:

\[
V' = \text{softmax}(\text{sum}(W_D \otimes D_{\text{concat}}))V \in \mathbb{R}^{L \times d}
\]  

(18)

where the \text{sum} operation is applied to the first dimension of the product of the weight matrix and the concatenated distance maps, effectively removing the first dimension.

\( b: \text{LATE FUSION} \)

In this setup, three parallel attention mechanisms are employed, each processing the input in a different manifold, thus computing the manifold feature representations \( V'_E = \text{softmax}(D_E)V, V'_{\text{SPD}} = \text{softmax}(D_{\text{SPD}})V \) and \( V'_G = \text{softmax}(D_G)V \) for the Euclidean, SPD and Grassmann manifolds, respectively. Then, the output feature representation \( V' \in \mathbb{R}^{L \times d} \) is equal to:

\[
V' = L(\text{concat}(V'_E, V'_{\text{SPD}}, V'_G))
\]  

(19)

where \( L \) performs a linear projection from the size of the concatenated feature representation \( 3d \) to the final size of the feature representation \( d \).

IV. EXPERIMENTAL RESULTS

A. IMPLEMENTATION DETAILS

MMA was introduced in several state-of-the-art hierarchical and compact ViTs to replace their vanilla self-attention mechanism. More specifically, the ViT-Lite-6/4, CVT-6/4 and CCT-7/3 × 2 models proposed in \cite{4}, the Swin-T model proposed in \cite{7} and the UniFormer-S model proposed in \cite{28} were employed, giving rise to new models with the same name and the MMA-* prefix, when MMA is utilized. For the Swin-T model, a patch size of 2, a window size of 4, an embedding dimension of 96, an mlp ratio of 2, depths of (2, 6, 4) and number of heads equal to (3, 6, 12) for the different layers were selected so that the model can be trained on the small datasets that was tested on.

Experiments were conducted on 4 well-known image classification datasets, namely C-10 \cite{53}, C-100 \cite{53}, T-ImageNet \cite{54} and ImageNet, as well as on a popular semantic segmentation dataset, namely ADE20K \cite{55}, \cite{56}. C-10 consists of 50K training and 10K test images equally distributed among 10 classes, C-100 consists of 50K training and 10K test images equally distributed among 100 classes, T-ImageNet consists of 100K training and 10K validation images equally distributed among 200 classes, while ImageNet consists of more than 1.2M training and 50K validation images distributed among 1000 classes. In T-ImageNet and ImageNet, the validation images were used to test the performance of ViTs. Finally, the Ade20K dataset consists of around 20K training and 2K test images depicting 150 classes of objects.

All experiments were run with a fixed batch size of 128 and for 200 epochs for C-10 and C-100 and 300 epochs for T-ImageNet and ImageNet. For fair comparison, the implementation of \cite{4} was used for the experiments on C-10, C-100 and T-ImageNet and the implementation of \cite{28} was used for the experiments on ImageNet and ADE20K. As a result, the AdamW optimizer \cite{57} was used with a weight decay of 0.1, a base learning rate of 5 \(-4\) and a cosine learning rate scheduler that adjusts the learning rate during training \cite{58}. A warmup of 10 epochs was applied by increasing gradually the learning rate from 0 to the initial value of the cosine learning rate scheduler \cite{59}. Label smoothing \cite{60} with a probability \( \epsilon = 0.1 \) was applied during training, where the true label is considered to have a probability of \( 1 - \epsilon \) and the probability \( \epsilon \) is shared between the other classes. For data augmentation, Auto-Augment \cite{61} was adopted to transform the training data with adaptive learnable transformations, such as shift, rotation, and color jittering. Moreover, the Mixup strategy \cite{62} was used to generate weighted combinations of random sample pairs from the training images.

Finally, the projection size for the covariance matrices was set to 4, the 3D convolution layer for the computation of the weight matrix had a kernel of size \( 1 \times 1 \times 1 \) and the LeakyRELU function had a negative slope of 0.1. Unless otherwise stated, the proposed MMA with the early fusion of all three manifolds was adopted in all experiments due to its optimal performance, as verified in the ablation study (i.e., Section IV-E). The code was implemented in PyTorch and all tested transformers with and without the proposed MMA were trained from scratch on a PC with 2 Nvidia 3090Ti gpus for fair comparison.

B. IMAGE CLASSIFICATION ON SMALL DATASETS

To demonstrate the benefits of MMA, a comparative evaluation of various hierarchical and compact ViTs with their vanilla self-attention and the proposed MMA in C-10, C-100 and T-ImageNet is performed. In addition, the selected
TABLE 1. Comparative evaluation on C-10, C-100 and T-ImageNet. Parameters and FLOPs were computed for images of size 32 \( \times \) 32.

| Method                  | Params (M) | FLOPs (G) | C-10    | Accuracy (%) |
|-------------------------|------------|-----------|---------|--------------|
| ResNet-100 [63]         | 1.70       | 0.25      | 93.39   | 72.78        |
| ResNet-164 [63]         | 1.73       | 0.26      | 94.54   | 75.67        |
| EfficientNet-B0 [64]    | 3.70       | 0.12      | 94.66   | 76.04        |
| Linformer [18]          | 3.96       | 0.28      | 92.45   | 70.87        |
| Performer [19]          | 3.85       | 0.28      | 91.58   | 73.11        |
| Reformer [20]           | 3.39       | 0.25      | 90.58   | 73.02        |
| Couplformer-7 [21]      | 3.85       | 0.28      | 93.44   | 74.53        |
| ViT-Lite-6/4 [4]        | 3.195      | 0.219     | 90.94   | 69.2         |
| MMA-ViT-Lite-6/4        | 3.197      | 0.222     | 92.47   | 72.55        |
| CVT-6/4 [4]             | 3.195      | 0.216     | 92.58   | 72.55        |
| MMA-CVT-6/4             | 3.197      | 0.218     | 93.53   | 75.92        |
| Swin-T [7]              | 7.049      | 0.243     | 91.88   | 72.34        |
| MMA-Swin-T              | 7.05       | 0.246     | 92.94   | 73.7         |
| CCT-7/3\(\times\)2 [4] | 3.859      | 0.29      | 93.65   | 74.77        |
| MMA-CCT-7/3\(\times\)2 | 3.861      | 0.293     | 94.74   | 77.5         |

FIGURE 3. Class activations in T-ImageNet. (a) Original images and class activation maps using (b) CVT-6/4, (c) MMA-CVT-6/4, (d) Swin-T, (e) MMA-Swin-T, (f) CCT-7/3 \(\times\)2 and (g) MMA-CCT-7/3 \(\times\)2 models.

Transformers are compared against other CNN networks and ViTs and the results are presented in Table 1. The results demonstrate that the MMA-enhanced ViTs achieve superior performance in all datasets with respect to their original versions at the expense of a small increase in parameters and FLOPs. A comparison with other state-of-the-art deep networks shows that MMA-CCT-7/3 \(\times\)2 achieves the highest performance in all datasets, outperforming other ViTs and even well-established CNN networks, such as ResNet-164 [63] and EfficientNet-B0 [64].

Furthermore, Fig. 3 depicts the class activation maps for CVT-6/4, Swin-T and CCT-7/3 \(\times\)2, and their MMA-* counterparts. It can be deduced that the MMA-enhanced ViTs become more attentive towards significant parts of the object of interest, such as the legs of a spider, the arch rib of a bridge and the eyes and noise of a dog, thus leading to more accurate classification results. Additional conclusions can be drawn from Fig. 4 that presents the impact of MMA on the tested ViTs in terms of model performance and generalization ability. Irrespective of their network architecture, all ViTs are significantly improved with MMA through a decrease in the training and validation losses and an increase in the validation accuracy.

Finally, Fig. 5 depicts the distribution of 10 random classes of T-ImageNet using the t-distributed stochastic neighbor embedding (t-SNE) algorithm [65] that is suitable for visualizing high-dimensional data by applying nonlinear dimensionality reduction. To this end, the feature vectors after the head concatenation of (3) are employed as the high-dimensional input to the t-SNE algorithm, which then computes a two-dimensional output. From the distribution of the 10 T-ImageNet classes in the 2D space, it can be observed...
that MMA leads to more compact classes (i.e., points of the same class closer to each other) and less stray points for all tested ViTs. All these results verify the ability of MMA to guide any Vision Transformer to accurately model the underlying input space and produce highly descriptive output representations. This is achieved through the fusion of different and complementary manifold representations that enables a Vision Transformer to model the important context of an image through a refined attention map of high discriminative power. As a result, the proposed MMA can effectively substitute the vanilla self-attention in Vision Transformers, significantly improving their classification performance.

TABLE 2. Comparative evaluation on ImageNet with images of size 224 × 224.

| Method       | Arch. | Params (M) | FLOPs (G) | Top-1 acc. |
|--------------|-------|------------|-----------|------------|
| ResNet-50 [66]  |      | 25.6       | 3.8       | 76.7       |
| EfficientNet-B0 [64] |      | 5.3        | 0.4       | 77.1       |
| ResNet-101 [66]  | CNN  | 44.5       | 7.6       | 78.3       |
| ResNet-152 [66]  |      | 60.2       | 11.3      | 78.9       |
| ConvNeXt-T [67] |      | 28.0       | 4.5       | 82.1       |
| PVT-S [10]     |      | 24.5       | 3.8       | 79.8       |
| DeiT-S [14]    |      | 22.1       | 4.6       | 79.9       |
| Twins-PCPVT-S [17] |    | 24.1       | 3.8       | 81.2       |
| Swin-T [7]     |      | 29.0       | 4.5       | 81.3       |
| ConvViT-S [3]  |      | 27.0       | 5.4       | 81.3       |
| PoolFormer-S36 [22] |   | 30.8       | 5.0       | 81.4       |
| T2T-ViT-14 [6] |      | 21.5       | 5.2       | 81.5       |
| Crossformer-T [25] |    | 27.8       | 2.9       | 81.5       |
| Twins-SVT-S [17] | ViT | 24.0       | 2.9       | 81.7       |
| SwinV2-T [24]  |      | 28.0       | 5.9       | 81.8       |
| DeiT-T [27]    |      | 28.3       | 4.6       | 82.0       |
| CvT-S [5]      |      | 24.2       | 4.5       | 82.0       |
| ViTAE-S [26]   |      | 23.6       | 5.6       | 82.0       |
| UniFormer-S [28] |    | 21.5       | 3.6       | 82.3       |
| Crossformer-S [25] |    | 30.7       | 4.9       | 82.5       |
| MMA-UniFormer-S |      | 21.5       | 3.7       | 82.6       |

C. IMAGE CLASSIFICATION ON IMAGENET

To demonstrate the ability of the proposed MMA to improve the performance of Vision Transformers even on large and more challenging image classification datasets, such as ImageNet, we employed one of the state-of-the-art ViTs, namely UniFormer [28] and we replaced the attention mechanism of its small version (i.e., UniFormer-S) with the proposed MMA, thus obtaining the MMA-UniFormer-S version of the model. Table 2 presents the performance improvement achieved after the modification, as well as performs a comparison against other state-of-the-art small-sized Vision Transformers.

From the results, it can be seen that the MMA-UniFormer-S outperforms UniFormer-S by 0.3% on ImageNet, when the proposed multi-manifold attention is introduced, managing to achieve a state-of-the-art performance on ImageNet with an accuracy of 82.6%. The results verify the ability of the proposed MMA to improve the performance of Vision Transformers in larger and more challenging datasets. A visualization of the attention maps extracted from the original UniFormer-S and the modified MMA-UniFormer-S
models on a few images of ImageNet is presented in Fig. 6. This figure further demonstrates that the use of additional manifolds in the attention mechanism of a Vision Transformer can lead to a more refined and descriptive attention map that can better describe the content of an image. Of notable importance are the cases of mountain bike identification (Fig. 6(b-c)), in which it can be seen that the MMA-UniFormer-S manages to correctly identify the object of interest, overcoming the mis-classification errors performed by the UniFormer-S.

D. SEMANTIC SEGMENTATION ON ADE20K

To further evaluate the ability of the proposed MMA mechanism to improve the performance of Vision Transformers on tasks different from image classification, we train and evaluate the MMA-UniFormer-S on semantic segmentation using the challenging scene parsing ADE20K dataset. More specifically, we employ the MMA-UniFormer-S, pre-trained on ImageNet, as a backbone to the semantic FPN network [10] and we employ the training scheme of the UniFormer with 80K iterations for fair comparison among the models.

From Table 3, it can be seen that the introduction of MMA-UniFormer-S as a backbone to the semantic FPN network can lead to significant improvements in the performance of the model in semantic segmentation. More specifically, the use of MMA-UniFormer-S leads to an improvement of 0.9% in mIoU with respect to UniFormer-S, thus leading
TABLE 4. Ablation study using the MMA-ViT-Lite-6/4 model with the early and late fusion of manifolds.

| Manifolds | E | SPD | G | Params (M) | FLOPs (G) | Accuracy (%) |
|-----------|---|-----|---|-----------|----------|--------------|
| Early Fusion | ✓ | ✓ | ✓ | 3.195 | 0.219 | 90.94 | 69.2 |
|            | ✓ | ✓ | ✓ | 3.197 | 0.214 | 90.49 | 70.38 |
|            | ✓ | ✓ | ✓ | 3.197 | 0.214 | 88.96 | 67.48 |
|            | ✓ | ✓ | ✓ | 3.197 | 0.221 | 91.89 | 72.41 |
|            | ✓ | ✓ | ✓ | 3.197 | 0.221 | **92.85** | **72.48** |
|            | ✓ | ✓ | ✓ | 3.197 | 0.215 | 92.03 | 71.35 |
|            | ✓ | ✓ | ✓ | 3.197 | 0.222 | 92.47 | **72.55** |

Late Fusion

| Early Fusion | ✓ | ✓ | ✓ | 3.59 | 0.252 | 91.43 | 71.71 |
| Late Fusion  | ✓ | ✓ | ✓ | 3.59 | 0.253 | 91.15 | 71.21 |
| Late Fusion  | ✓ | ✓ | ✓ | 3.59 | 0.246 | 89.3 | 68.25 |
| Late Fusion  | ✓ | ✓ | ✓ | 3.983 | 0.285 | 90.95 | 71.78 |

FIGURE 8. Class activations in C-100 using MMA-ViT-Lite-6/4. (a) Original images and class activation maps using (b) Euclidean, (c) SPD, (d) Grassmann and (e) all three manifolds.

From the results of Table 4, it can be concluded that when the manifolds are utilized on their own, the Euclidean and SPD manifolds perform similarly and slightly better than the Grassmann manifold. However, when two or more manifolds are combined, the performance of MMA-ViT-Lite-6/4 is significantly improved in both C-10 and C-100 datasets. The best results are achieved when all three manifolds are employed, as it is seen from the performance of MMA-ViT-Lite-6/4 in the more challenging C-100 dataset. Noteworthy is the fact that the increase in the performance of MMA-ViT-Lite-6/4 is simultaneously accompanied by a negligible increase in the number of network parameters and a small increase in the number of floating point operations (FLOPs). Similar observations can be made when the late fusion of the manifold representations is performed. These results verify that the SPD and Grassmann manifolds contain different and supplementary information to the Euclidean manifold through the modelling of the appearance, color and texture variations in images. This information can guide a Vision Transformer to become more attentive towards a better modelling of the underlying input space, enabling it to achieve improved performance.

On the other hand, a comparison between early and late fusion of manifolds reveals that early fusion leads to superior performance in both datasets, while utilizing almost 25% fewer network parameters and around 28% fewer FLOPs. These results show that fusing the manifold distance maps inside the Transformer ensures the generation of a refined attention map with high discriminative power.

To further highlight the benefits of employing additional manifolds, a visualization of class activation maps in a few images from the C-100 dataset is illustrated in Fig. 8. It can be observed that although the vanilla self-attention allows the model to pay attention on the ears of a kangaroo or the hump of a camel, the proposed MMA enables the model to become highly attentive towards additional information, such as the entire body of the kangaroo and both the hump and the legs of a camel, thus increasing the confidence of the model in its classification results.

V. CONCLUSION

This work proposes MMA as a novel self-attention mechanism that is suitable for any Vision Transformer irrespective of its architecture. The motivation behind MMA lies in the use of three distinct manifolds, namely Euclidean, SPD and Grassmann to model the input and produce a fused attention map that can more accurately attend to the important context of the input. Experimental results with hierarchical and compact ViT variants on several image classification datasets (i.e., C-10, C-100, T-ImageNet and ImageNet) and a challenging semantic segmentation dataset (i.e., ADE20K) verify the effectiveness of MMA in producing highly descriptive output representations and improving the performance of Vision Transformers in both image classification and semantic segmentation.
B. Wang, Y. Hu, J. Gao, Y. Sun, F. Ju, and B. Yin, “Adaptive fusion of heterogeneous manifolds for subspace clustering,” IEEE Trans. Neural Netw. Learn. Syst., vol. 32, no. 8, pp. 3484–3497, Aug. 2021.

X. Chu, Z. Tian, B. Zhang, X. Wang, and C. Shen, “Conditional positional encodings for vision transformers,” 2021, arXiv:2102.10882.

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 5998–6008.

S. Bhattacharya, N. Souly, and M. Shah, “Covariance of motion and appearance features for spatio temporal recognition tasks,” 2016, arXiv:1605.05355.

R. Venelupalli and D. W. Jacobs, “Riemannian metric learning for symmetric positive definite matrices,” 2015, arXiv:1501.02393.

M. Harandi, C. Sanderson, C. Shen, and B. Lovell, “Dictionary learning and sparse coding on Grassmann manifolds: An extrinsic solution,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 3120–3127.

J. Zhang, G. Zhu, R. W. Heath Jr., and K. Huang, “Grassmannian learning: Embedding geometry awareness in shallow and deep learning,” 2018, arXiv:1808.02229.

A. Trockman and J. Z. Kolter, “Orthogonalizing convolutional layers with the Cayley transform,” in Proc. Int. Conf. Learn. Represent., 2021, pp. 1–12.

A. Krizhevsky, “Learning multiple layers of features from tiny images,” M.S. thesis, Dep. Comput. Sci., Univ. Toronto, 2009.

Y. Le and X. Yang, “Tiny imagenet visual recognition challenge,” CS 231N, pp. 3, 2015, vol. 7, no. 7.

B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, “Scene parsing through ADE20K dataset,” in Proc. IEEE Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5122–5130.

B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, “Semantic understanding of scenes through the ADE20K dataset,” Int. J. Comput. Vis., vol. 127, no. 3, pp. 302–321, Mar. 2019.

I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” in Proc. ICLR, 2019, pp. 10–19.

T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, “Bag of tricks for image classification with convolutional neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 558–567.

P. Goyal, P. Dollár, R. Girshick, P. Noordhuis, L. Wesolowski, A. Kyrola, A. Tulloch, Y. Jia, and K. He, “Accurate, large minibatch SGD: Training ImageNet in 1 hour,” 2017, arXiv:1706.02677.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2818–2826.

E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “AutoAugment: Learning augmentation strategies from data,” in Proc. CVPR, 2019, pp. 113–123.

H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “Mixup: Beyond empirical risk minimization,” in Proc. ICLR, 2018, pp. 1–13.

K. He, X. Zhang, S. Ren, and J. Sun, “Identity mappings in deep residual networks,” in Proc. ECCV. Cham, Switzerland: Springer, 2016, pp. 630–645.

M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proc. ICML, 2019, pp. 6105–6114.

L. Van der Maaten and G. Hinton, “Visualizing data using t-SNE,” J. Mach. Learn. Res., vol. 9, no. 11, pp. 1–27, 2008.

K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A convnet for the 2020s,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 11966–11976.

Dimitrios Konstantinidis received the B.S. degree in electrical and electronic engineering from AUTH, in 2009, the Advanced M.S. degree in artificial intelligence from KU Leuven, in 2012, and the Ph.D. degree from Imperial College London, in 2017, with a focus on monitoring urban changes from satellite images. He is currently with the Information Technologies Institute (ITI), Centre for Research and Technology Hellas (CERTH), as a Postdoctoral Researcher. His research interests include computer vision, machine/deep learning applications, and robotics.