A Robust Person Shape Representation via Grassmann Channel Pooling

Tetsu Matsukawa[0000−0002−8841−6304] and Einoshin Suzuki[0000−0001−7743−6177]

Faculty of Information Science and Electrical Engineering, Kyushu University, JAPAN
{matsukawa, suzuki}@inf.kyushu-u.ac.jp

Abstract. Robustly estimating a person's orientation in various clothing and image styles is essential for implementing vision systems in real-world applications. In this task, the spatial arrangement of local parts can be a key factor for a precise estimation. Therefore, we focus on channel pooling, which summarizes less relevant channel activations of a feature map produced by ConvNets. However, the limited discriminative ability of the representation produced by naive channel pooling methods leads to imprecise estimations. To address this problem, we propose Grassmann Channel Pooling (GCP), which summarizes each feature map as a linear subspace of its spatial bases. Specifically, GCP extracts the spatial bases from a feature map, where each basis represents globally similar positions across channels. A linear subspace spanned by these vectors is invariant to permutations of feature channels and scalings of the feature map and is thus expected to be robust. Meanwhile, GCP extracts discriminative co-occurrence information from various spatial positions using the projection metric of Grassmann manifold. Experimental results on the PersonX and TUD datasets indicate that GCP has superior discriminative power compared to existing pooling methods, as well as its robustness.

Keywords: Person Orientation Estimation · Grassmann and Bilinear Pooling · Robustness · Image Style

1 Introduction

Estimating a person's orientation from roughly aligned images [19, 39] is essential in various vision applications. For example, it can aid in developing a driver assistance system [26]. Recently, computer vision researchers have been focusing on the domain generalization setting, where test samples are from different domains of training datasets [8, 24]. A system which conducts person orientation estimation is often implemented in an open-set environment, where the testing environment differs from the domain of training datasets. By generalizing the system to different domains, we can usually enhance the system's robustness in rare locations and various weather conditions [28], as well as against noise caused by camera malfunctions and other factors.

Convolutional Neural Networks (ConvNets) [33, 47] are arguably the de facto standard for image recognition. By repeatedly applying convolutional filters and non-linear activation functions to each position in an image, ConvNets obtain a feature map, where

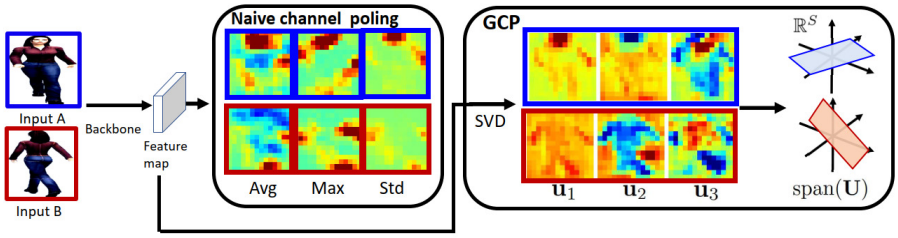


Fig. 1. Concept overview. Naive channel pooling methods often result in the loss of significant discriminative information from a feature map, as they primarily concentrate on local information. The spatial bases of a feature map encompass various aspects of spatial information, each derived from analyzing the global positions within the map. The linear subspace spanned by the spatial bases is invariant to the order of the bases and the singular values, making it robust.

each feature channel corresponds to a filter activation. Spatial pooling is typically applied to the feature map to achieve invariance to image transformations, learn more compact representations, and enhance robustness to noise and clutter [5].

Nevertheless, for several visual recognition tasks, such as estimating the orientation of pedestrians from roughly aligned images [19,39], the spatial layout of local parts can be a key factor for precise estimation, but a spatial pooling operation reduces this factor. Also, preserving channels in the pooling process could compromise the model's robustness when handling data from various domains, potentially causing overfitting. Since the spatial layout of local parts can be a key factor, we expect that the shape information obtained by compressing feature channels helps to perform precise recognition.

In this paper, we examine two variations in the feature map related to different clothing textures and image styles. First, variations in textures and styles can result in channel permutation, a phenomenon in which specific spatial patterns within a feature channel are activated in different channels. Second, alterations in image style can result in variations in activation magnitude. Past researches in domain generalization [12,24,25,36, 41] commonly employ Instance Normalization (IN) [50] to eliminate instance-specific characteristics within a feature map. While IN effectively addresses the latter variation, it is unable to manage the former, as it operates independently on each channel.

To obtain robust representations against the channel permutation, we focus on channel pooling [48] which summarizes a feature map along the channel dimension. Existing works [9,20,34,35,55] use a statistical value, e.g., the average, the standard deviation, and the maximum value. However, these standard values result in a significant loss of information within the feature map, particularly regarding global spatial patterns within specific channels. Additionally, since the feature map reflects local filter activations, the summarized representation tends to emphasize local information, such as the presence of a frontal head to differentiate between front and back poses (Fig. 1). When failing to detect such a local clue, the estimation of orientation fails.

To obtain a global shape representation beyond the local information offered by existing channel pooling methods, extracting global spatial patterns from the channels is essential. Meanwhile, the representation should be consistent with the feature channel permutation and the activation magnitude. We argue that applying Singular Value

Decomposition (SVD) to the feature map can fulfill these requirements. Each of the bases obtained through SVD represents common spatial patterns across multiple feature channels, representing a position which shares a globally similar property with different positions, e.g., extracting the shapes of limbs rather than focusing solely on the head (Fig. 1). Meanwhile, SVD offers consistent spatial bases, regardless of the order of the feature map channels. Simultaneously, the activation magnitude is decoupled in the singular values.

Although SVD extracts multiple spatial bases from a feature map, the order of similar bases, such as descending order based on the singular values, can differ across various image styles, even for the same individual and pose (Fig. 4 (b)). This variation arises because the singular values of a feature map encapsulate factors related to style. Since the order depends on image styles, simply concatenating the spatial bases as a feature representation diminishes robustness. Meanwhile, these bases span a unique linear subspace.

Based on the discussions above, we propose a novel channel pooling method called Grassmann Channel Pooling (GCP), which summarizes each feature map as a linear subspace of the spatial bases. Because the subspace is a point on Grassmann manifold [1,17,21,44,49,54], we represent it as the vector representation of its tangent space to be handled with neural networks. Specifically, we use projection metric [11, 21], which extracts co-occurrence information from all pairwise positional combinations, which is discriminative among different poses [37]. Since the spatial bases are invariant to permutations and magnitudes of feature channels, GCP encompasses robustness against these variations.

Similarly to GCP, Grassmann Spatial Pooling (GSP) [54] also summarizes a feature map as a linear subspace. Wei et al. showed that GSP corresponds to Bilinear Pooling [29, 30, 32] with homogeneous singular values, which is robust to illumination and appearance changes [54]. Nevertheless, GSP lacks robustness against style changes because it shares eigenvectors with the Gram matrix, which represents various image styles [14, 15]. This paper demonstrates that GCP is a dual form of GSP and exhibits greater robustness to style changes, as its eigenvectors and eigenspectral differ from those of the Gram matrix.

2 Related Works

We first explain exsisting features for our target problem. We then explain general domain generalization works, followed by closely related poolings and subspace representations.

Feature representation for person orientation estimation. Kim et al. proposed a two-stream ConvNet [26], which extracts appearance and Co-OCurence (COOC) features. The appearance feature uses the feature map directly to retain channel and position information. Many works on orientation estimation [38, 53] use such simple features. COOC, a spatial max pooling representation of co-occurrence activations of channels, is remarkably effective in distinguishing subtle differences with similar visual characteristics [32, 42]. However, it does reduce the spatial resolution of a feature map.

High-Resolution Net (HRNet) [45, 52], which maintains high-resolution spatial information throughout the entire process of feature extraction, showed impressive performance on person orientation estimation [56]. Originally, HRNet was proposed for person pose estimation, which detects key points of parts, e.g., elbow and wrist [45]. This task is more complex than person orientation estimation because it needs to detect many key points. HRNet is a foundation for advanced architectures [6, 57] for person pose estimation. HRNet is also helpful for semantic segmentation [52] and object detection [52], where high-resolution features are effective. In this paper, we demonstrate that GCP enhances the performance of HRNet in person orientation estimation, though the theory behind GCP is not limited to specific model architectures or tasks.

Domain generalization. Most works aim at learning robust feature representations which maintain performance regardless of the domain [7, 60]. A seminal work theoretically proved that well-generalized representations should remain consistent across different environments [3]. Based on this principle, many works focus on the learning process [60], such as by regularizing the representation by using multiple domain datasets [8]. We focus on adding domain generality from network architecture design. Existing works [12, 24, 25, 36, 41] commonly use Instance Normalization (IN) [50] in this approach. We show that GCP complemently works with IN, enabling higher robustness.

Channel pooling. Compared with spatial pooling, research on channel pooling is scarce. This pooling adds robustness against local transformations by averaging different filter responses [48] and is used to learn rotation-invariant filters [35]. Also, it is used to regularize network training [20] and achieve a compact feature map [9, 34]. A work [55] on a self-attention module, which is a mechanism to transform a feature map based on a global property of a feature map, uses channel pooling to determine the channel weights. These works use simple poolings, such as average and max, unable to extract global spatial patterns within them.

Bilinear pooling and subspace representation. Bilinear Pooling (BP) [31, 32], which produces summarized representations of pairwise correlation of feature channel activations within a feature map, well describes textural properties of images [14]. Wei et al. showed a correspondence of Grassmann Pooling (GP), which represents a feature map as a subspace, to a special case of BP [54]. Nevertheless, existing BP and GP are spatial pooling, which is unsuitable for capturing spatial properties of images. This paper proposes to apply GP to the opposite axis of a feature map, enabling the extraction of interactions among spatial positions along with channels, which is known to obtain effective attention [13, 37]. Also, we show the superiority of GP compared with BP in robustness against style variations.

Previous studies often modeled a set of images as a linear subspace and treated as a point on Grassmann manifold [17, 21, 44] - a smooth manifold of a linear subspace [1]. A domain adaptation regression work aligned the feature representations based on the subspace of feature channels on two domains [10]. Several other works use basis vectors of a feature map [16, 40]. Subspace distillation is proposed for continual learning by aligning channel bases (not spatial bases) obtained by SVD on the feature maps [40]. Also, matrix decomposition has been focused on as an alternative to self-attention because it analyzes the global context of a feature map [16]. These works did not aim

to obtain robust shape representations but to align sample distribution [10], transfer knowledge of one network to another [40], and transform a feature map for a general purpose [16]. Thus, they do not use the spatial bases of a feature map.

3 Preliminalies

3.1 Target Problem

The representation should remain consistent despite various environmental changes when implementing a vision system in an open world. Thus, we tackle a generalizable person orientation estimation problem, which is defined as follows: Given a labeled training dataset $\mathcal{D}^{\mathrm{tr}}$, a model $\mathcal{M}$ trains parameters to predict the orientation θ, which is a continuous angle greater than 0° and less than 360°. A test dataset $\mathcal{D}^{\mathrm{te}}$, which simulates various environment changes, evaluates the model $\mathcal{M}$'s performance.

We consider two environmental changes: (1) $\mathcal{D}^{\mathrm{te}}$ contains different clothing textures from $\mathcal{D}^{\mathrm{tr}}$. (2) $\mathcal{D}^{\mathrm{te}}$ contains images captured in different settings from $\mathcal{D}^{\mathrm{tr}}$, such as the camera setting, the noise, and the weather, which leads to image style variations. In order to simulate the first environmental change, we use a test dataset $\mathcal{D}^{\mathrm{te}}$, which contains different individuals from $\mathcal{D}^{\mathrm{tr}}$. Regarding the second environmental change, we modify the image style of $\mathcal{D}^{\mathrm{te}}$ using a style transfer model to generate S style-transformed datasets $\mathcal{D}_1^{\mathrm{te}}, ..., \mathcal{D}_S^{\mathrm{te}}$ and evaluate their average performance.

3.2 Backbone Model

Let $\mathsf{I} \in \mathbb{R}^{3\times H_0\times W_0}$ be an input image where 3, H_o, and W_o, are the number of color channels, the height, and the width, respectively. A ConvNet model is formulated as follows: $\mathcal{M} = h\left(g\left(f\left(\mathsf{I}\right)\right)\right)$, where $f(\cdot)$, $g(\cdot)$ and $h(\cdot)$, respectively represent a feature map extractor, a pooling operator, and a regression head. The feature map extractor $f(\cdot)$ outputs a feature map $\mathsf{X} \in \mathbb{R}^{C\times H\times W}$, where C, H, and W represent the number of feature channels, the height, and the width, respectively. The pooling operator $g(\cdot)$ summarizes X into a feature representation $\mathbf{z}$, from which the regression head $h(\cdot)$ regresses the orientation. The parameters of $h(\cdot)$ and $f(\cdot)$ are updated end-to-end, while $g(\cdot)$ have no trainable parameters.

We often view the feature map as a matrix as $\mathbf{X} \in \mathbb{R}^{S\times C} = \mathrm{reshape}(\mathsf{X}, [C,\ S])^{\top}$, where $\mathrm{reshape}(\cdot, [\,])$ is an operation to reshape an input tensor $\cdot$ to the size specified by $[\,]$ and $S = H \times W$ is the total number of spatial positions in X. On this matrix, the c-th column represents a feature map $\mathbf{x}_c \in \mathbb{R}^S$ of channel c, and the s-th row represents a feature channels $\hat{\mathbf{x}}_s \in \mathbb{R}^C$ of spatial position s, i.e., $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_C]$ and $\mathbf{X}^{\top} = [\hat{\mathbf{x}}_1, ..., \hat{\mathbf{x}}_S]$.

3.3 Related Methods

Instance Normalization (IN) [50]. IN adds the model's robustness against style variations, which enhances domain generality [12, 24, 25, 36, 41]. Let x_{cij} be an element of

X at the c-th channel and the i, j-th spatial positions. IN normalizes each of the feature channels to zero mean and unit standard deviation as follows:

$$x'_{cij} = \frac{x_{cij} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}, \quad \mu_c = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} x_{cij}, \quad \sigma_c^2 = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}(x_{cij} - \mu_c)^2, \tag{1}$$

where x'_{cij} is an element of the output layer's feature map, and ϵ is a small constant value to prevent zero division, which is set to $1e^{-5}$ in the default setting of PyTorch.

The environmental variations can lead to two variations in the feature map: channel permutation, where certain spatial patterns within a feature channel activate in different channels, and variations in activation magnitude. IN mitigates the effects of the latter variations. Meanwhile, IN cannot address the former type as it operates individually on each channel.

Channel pooling. To achieve robust representations against channel permutations, we focus on channel pooling [48]. In explaining existing methods, we use the column vectors of the reshaped matrix $\mathbf{X}$ in line with GCP. Examples of channel pooling include the average, the maximum, and the standard deviation of channels, whose outputs $\{\mathbf{z}_{\mathrm{avg}}, \mathbf{z}_{\mathrm{max}}, \mathbf{z}_{\mathrm{std}}\} \in \mathbb{R}^S$ are defined as follows:

$$\mathbf{z}_{\mathrm{avg}} = \frac{1}{C}\sum_{c=1}^{C}\mathbf{x}_c, \quad \mathbf{z}_{\mathrm{max}} = \max_{c\in\{1,..C\}} \mathbf{x}_c, \quad \mathbf{z}_{\mathrm{std}} = \sqrt{\frac{1}{C-1}\sum_{c=1}^{C}(\mathbf{x}_c - \mathbf{z}_{\mathrm{avg}})^2}, \tag{2}$$

where $\max(\cdot)$ is an elementwise maximum operation. In these methods, the representations are invariant to channel permutations. Nevertheless, summarizing feature channel activations into a single dimension significantly loses discriminative information regarding the targets, as explained in Sec. 1.

4 Grassmann Channel Pooling

To overcome the limited discriminative ability of the existing channel pooling methods, we propose Grassmann Channel Pooling (GCP), which summarizes each feature map as a linear subspace of spatial bases. In this section, we detail GCP and its properties.

4.1 Method

GCP summarizes each feature map as a linear subspace of spatial bases. More specifically, to extract global spatial patterns of channels included in the feature map, GCP obtains the spatial bases of an input feature map by applying Singular Value Decomposition (SVD). Meanwhile, it discards the singular values which contain illumination-related factors [54] in an image style and uses only the spatial bases.

Formally, we first apply the reshape operation to a feature map X as $\mathbf{X} = \mathrm{reshape}(\mathsf{X}, [C, S])^\top$. Then SVD factorizes the matrix as $\mathbf{X} = \mathbf{U}^*\mathbf{S}^*\mathbf{V}^{*\top}$, where $\mathbf{U}^* \in \mathbb{R}^{S\times R^*}$ is left singular vectors, $\mathbf{S}^* \in \mathbb{R}^{R^*\times R^*}$ is a matrix that includes the singular values in its diagonal elements, R^* is its rank, and $\mathbf{V}^* \in \mathbb{R}^{C\times R^*}$ is the right singular vector.

Algorithm 1 Grassmann Channel Pooling (GCP)

Input: Feature tensor X, rank R

$\mathbf{X} = \mathrm{reshape}(\mathsf{X}, [C, S])^\top$

$[\mathbf{U}^*, \mathbf{S}^*, \mathbf{V}^{*\top}] = \mathrm{SVD}(\mathbf{X})$

$\mathbf{U} = \mathbf{U}^*[:, 1:R]$

$\Phi(\mathbf{U}) = \mathbf{U}\mathbf{U}^\top$

Output: A representation $\mathbf{y} = \mathrm{vec}(\Phi(\mathbf{U}))$

To remove noisy and unessential basis vectors, we select the submatrix corresponding to leading $R \leq R^*$ singular vectors, and then SVD approximates the reshaped feature map as $\mathbf{X} \approx \mathbf{U}\mathbf{S}\mathbf{V}^\top$, where $\mathbf{U} \in \mathbb{R}^{S\times R}$, $\mathbf{S} \in \mathbb{R}^{R\times R}$, and $\mathbf{V} \in \mathbb{R}^{C\times R}$.

We refer $\mathbf{U}$ as the spatial bases, and they are orthonormalized vectors, i.e., $\mathbf{U}^\top\mathbf{U} = \mathbf{I}$. Meanwhile, we refer $\mathbf{V}$ as the channel bases, and they satisfy $\mathbf{V}^*\mathbf{V}^\top = \mathbf{I}$. Because the SVD decomposes a feature map by considering all channels and positions, each of the spatial bases represents globally similar points among feature channels (Fig. 1).

The order of the spatial bases, e.g., the decreasing order based on their singular values, can vary among different image styles, even within the same person and pose. Let us consider the case $R = 3$, i.e., $\mathbf{U} = [\mathbf{u}_1\mathbf{u}_2\mathbf{u}_3]$. For example, a style variation can change $\mathbf{U}$ to $\mathbf{U}' \approx [\mathbf{u}_1\mathbf{u}_3\mathbf{u}_2]$ (Fig. 4 (b)). If we simply vectorize $\mathbf{U}$ as a representation, i.e., $\mathbf{z} = [\mathbf{u}_1^\top\mathbf{u}_2^\top\mathbf{u}_3^\top]^\top$, the variation decreases the prediction's robustness unless enough training samples to the representation $\mathbf{z}' \approx [\mathbf{u}_1^\top\mathbf{u}_3^\top\mathbf{u}_2^\top]^\top$ exist.

Meanwhile, the linear subspace spanned by $\mathbf{U}$ is unique, regardless of the order. Therefore, we represent each sample as the subspace. A R-dimensional linear subspaces in $\mathbb{R}^S$ is a point on the Grassmann manifold $Gr(R, S)$, which is a compact Riemannian manifold of dimension $R(S-R)$ [1]. Unlike Euclidean space, implementing regression operations on $Gr(R, S)$ is not straightforward. Fortunately, the projection metric [11] can embed the data on $Gr(R, S)$ onto Euclidean space by using the projection transformation $\Phi(\mathbf{U}) = \mathbf{U}\mathbf{U}^\top$. The inner product defined by $< \mathbf{U}_1, \mathbf{U}_2 >_\Phi = \mathrm{Tr}\left[\Phi(\mathbf{U}_1)^\top\Phi(\mathbf{U}_2)\right]$ induces the following projection metric:

$$d_R(\mathbf{U}_1, \mathbf{U}_2) = 2^{-\frac{1}{2}}\|\Phi(\mathbf{U}_1) - \Phi(\mathbf{U}_2)\|_\mathrm{F}, \tag{3}$$

where $\|\cdot\|_\mathrm{F}$ represents the Frobenius norm.

The projection metric treats each data point as though it exists in Euclidean space following the projection transformation. Therefore, we employ the vectorized linear subspace after this transformation for the subsequent layer, which is the MLP layer for regression. Because the projection $\Phi(\mathbf{U})$ is a $S\times S$ symmetric matrix, we perform a half vectorization by eliminating the repeated elements, resulting in a vector $\boldsymbol{z} \in \mathbb{R}^{\frac{1}{2}S(S+1)}$ for the subsequent process. The half-vectorized operation $\mathrm{vec}(\cdot)$ is written as follows:

$$\mathbf{z} = \mathrm{vec}\left(\Phi(\mathbf{U})\right) = [\mathrm{diag}(\Phi(\mathbf{U}))^\top \ \ \sqrt{2}\mathrm{offdiag}(\Phi(\mathbf{U}))^\top]^\top, \tag{4}$$

where $\mathrm{diag}(\cdot)$ and $\mathrm{offdiag}(\cdot)$ respectively represent the operations to obtain diagonal and upper triangle elements.

GCP summarizes the co-occurrence of basis vector values at different points in an $S \times S$ matrix $\Phi(\mathbf{U})$, with each row and column representing $S = H \times W$ positions.

Thus, $\Phi(\mathbf{U})$ includes the product values of all pairwise positional combinations [37]. Co-occurrence statistics effectively detect subtle pattern differences [42], suggesting this statistic among positions has a strong ability to distinguish different orientations. Meanwhile, representations from absolute position pairs are sensitive to spatial misalignment in the same orientation. However, we can mitigate this sensitivity by reducing the spatial resolution of an input feature map through local pooling as long as the misalignment is not severe.

4.2 Computational cost

Algorithm 1 summarizes GCP. On the step 2, SVD of a $S \times C$ matrix $\mathbf{X}$ requires $O(\min(SC^2, S^2C))$ time. In PyTorch, torch.linalg.svd function performs this operation on CPU. On the step 4, the inner product $\mathbf{U}\mathbf{U}^\top$ requires $O(S^2R)$ time on GPU. When $C < S$, $O(SC^2)$ time on CPU, or $O(S^2R)$ time on GPU becomes the most expensive cost. Otherwise, it becomes $O(S^2C)$ time on CPU, or $O(S^2R)$ time on GPU. The cost would be acceptable if S and C are relatively low.

4.3 Robustness

We discuss the robustness of GCP against the two feature map variations: channel permutation and scalings of activations. We expect that GCP is robust because the following invariant properties hold.

Channel permutation invariance. Let us denote the permutation operation of channels as $\pi(\cdot)$, i.e., exchanging the column of an input matrix with an arbitrary order. We define $g(\cdot)$ has channel permutation invariance when the following relation holds:

$$g(\mathbf{X}) = g\left(\pi\left(\mathbf{X}\right)\right). \tag{5}$$

GCP has this invariance because we can express $\pi(\mathbf{X})$ as $\mathbf{X}\mathbf{P}$, where $\mathbf{P}$ is a $C \times C$ permutation matrix, and thus $\pi(\mathbf{X}) = \mathbf{U}\mathbf{S}\mathbf{V}^\top\mathbf{P} = \mathbf{U}\mathbf{S}\pi(\mathbf{V}^\top)$. Namely, the spatial basis vectors and their order, e.g., the decreasing order based on their singular values, remain unchanged by the order of input channels.

Channel magnitude invariance. Let us scale a feature map by a non-zero scalar $c \in \mathbb{R}^1$. We define $g(\cdot)$ has channel magnitude invariance when the following relation holds:

$$g(\mathbf{X}) = g\left(c \cdot \mathbf{X}\right). \tag{6}$$

GCP has this invariance because $c \cdot \mathbf{X} = \mathbf{U}(c \cdot \mathbf{S})\mathbf{V}^\top$. Namely, the scaling of the elements of $\mathbf{X}$ only alters the singular values which GCP neglects. Nevertheless, GCP is not invariant against different scaling values per channel and an additive change in channel elements. Applying IN [50] to the input feature map can add robustness in these situations. Meanwhile, GCP has the channel permutation invariance that IN does not have. Therefore, GCP and IN expect to work complementarily.

Of course, real variations are more complex than the above explanations, e.g., the variations may occur partially in an image region. Nevertheless, these invariances explain how the pooled feature produces a robust representation.

Table 1. Summary of BP variants.

spatial pooling ($C \times C$ matrix)			**channel pooling** ($S \times S$ matrix)		
BSP	MPN-BSP	GSP	BCP	MPN-BCP	GCP
$\mathbf{V}^{*\top}\left(\frac{1}{S}\mathbf{S}^{*^2}\right)\mathbf{V}^*$	$\mathbf{V}^{*\top}\phi\left(\frac{1}{S}\mathbf{S}^{*^2}\right)\mathbf{V}^*$	$\mathbf{V}^\top\mathbf{V}$	$\mathbf{U}^*\left(\frac{1}{C}\mathbf{S}^{*^2}\right)\mathbf{U}^{*\top}$	$\mathbf{U}^*\phi\left(\frac{1}{C}\mathbf{S}^{*^2}\right)\mathbf{U}^{*\top}$	$\mathbf{U}\mathbf{U}^\top$

4.4 Connection with Bilinear Pooling

Bilinear Pooling (BP) [31, 32] produces summarized representations of pairwise correlations of feature channel activations within a feature map. We show the connection of GCP with existing BP variants [29, 30, 54], and the cases when BP is used as channel pooling. They are expressed as follows:

Bilinear Spatial Pooling (BSP) [31,32]. Original BP, which is used as spatial pooling, is expressed as follows:

$$\mathbf{G} = \frac{1}{S}\sum_{s=1}^{S}\hat{\mathbf{x}}_s\hat{\mathbf{x}}_s^\top = \frac{1}{S}\mathbf{X}^\top\mathbf{X} = \mathbf{V}^{*^\top}\left(\frac{1}{S}\mathbf{S}^{*^2}\right)\boldsymbol{V}^*. \tag{7}$$

This matrix corresponds to the Gram matrix, which represents the textures and styles of images [14, 15]. In the style transfer work [15], transferring the style of an image is considered equivalent to altering the matrix $\mathbf{G}$ of that image. Thus, a change in style can significantly impact $\mathbf{G}$. Note that the $\mathbf{V}^{*^\top}$ corresponds to the eigenvectors, and $\frac{1}{S}\mathbf{S}^2$ corresponds to the eigenspectral of $\mathbf{G}$.

MPN-BSP [29, 30]. Matix Power Normalization (MPN) transforms the eigenspectral of $\mathbf{G}$ as $\mathbf{G}_{\mathrm{MPN}} = \mathbf{V}^{*\top}\phi\left(\frac{1}{S}\mathbf{S}^{*^2}\right)\mathbf{V}^*$, where $\phi(\cdot)$ is a normalization function, typically the matrix square-root [29,30]. MPN provides nontrivial improvements of BSP by remedying the burstiness problem - repeatedly appearing similar visual patterns [23, 27].

Grassmann Spatial Pooling (GSP) [54]. In the case when the normalization is given by $\phi(\cdot) = \mathbf{I}$ and selecting leading singular vectors, MPN-BSP reduces to GSP, i.e., $\mathbf{G}_{\mathrm{GSP}} = \mathbf{V}^\top\mathbf{V}$. GSP eliminates $\mathbf{S}^*$ in $\mathbf{G}$, and thus we expect that a style change less influences GSP compared to BSP and MPN-BSP.

Bilinear Channel Pooling (BCP). We consider the case of utilizing BP [31, 32] for channel pooling, as follows:

$$\mathbf{\Sigma} = \frac{1}{C}\sum_{c=1}^{C}\mathbf{x}_c\mathbf{x}_c^\top = \frac{1}{C}\mathbf{X}\mathbf{X}^\top = \mathbf{U}^*\left(\frac{1}{C}\mathbf{S}^{*^2}\right)\mathbf{U}^{*^\top}. \tag{8}$$

Because $\mathbf{\Sigma}$ shares the singular values $\mathbf{S}^*$ with $\mathbf{G}$, a style change can affect BCP. Meanwhile, $\mathbf{V}^*$ of $\mathbf{G}$ is absent in its eigendecomposition, and thus, we expect that a style change less influences BCP compared to BSP.

MPN-BCP. By applying MPN to BCP, we obtain $\mathbf{\Sigma}_{\mathrm{MPN}} = \mathbf{U}^*\phi\left(\frac{1}{C}\mathbf{S}^{*^2}\right)\mathbf{U}^{*\top}$. Setting $\phi(\cdot) = \mathbf{I}$ and selecting leading singular vectors, MPN-BCP reduces to GCP.

Table 1 summarizes BP variants. Spatial pooling methods (BSP/MPN-BSP/GSP) compress spatial dimensions of $\mathbf{X}$ and produce $C \times C$ feature channel co-occurrence matrix. All these methods and Gram matrix (=BSP) can be eigendecomposed with the channel bases $\mathbf{V}^*$ or $\mathbf{V}$ without the spatial bases $\mathbf{U}^*$ or $\mathbf{U}$, respectively. Meanwhile,

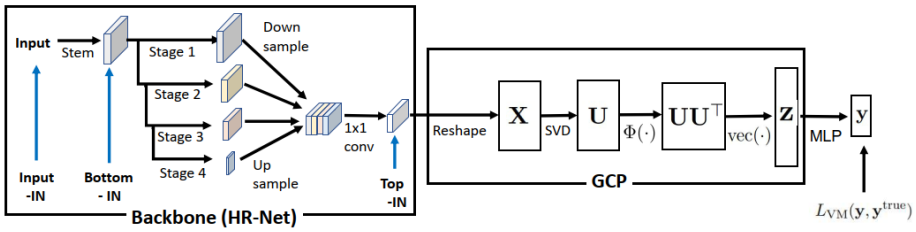


Fig. 2. Network architecture.

channel pooling methods (BCP/MPN-BCP/GCP) compress channel dimensions of $\mathbf{X}$ and produce $S \times S$ co-occurrence matrix of feature positions. All these methods can be eigendecomposed with the spatial bases $\mathbf{U}^*$ or $\mathbf{U}$ without the channel bases $\mathbf{V}^*$ or $\mathbf{V}$, respectively. We expect that GCP is most robust in these methods against a style change because only the eigendecomposition of GCP does not include both $\mathbf{V}^*$ and $\mathbf{S}^*$ in the Gram matrix $\mathbf{G}$. We will confirm these discussions in Sec. 5.3.

5 Experiments

5.1 Implementation Details

Architecture. Fig. 2 shows our architecture. Due to its effectiveness on the person orientation estimation task [56], we adopt HRNet [52] as the backbone model. We use a pre-trained model on the ImageNet dataset[1]. We resize input images to $(224, 224)$ pixels. The HRNet has four stages, each generating feature maps with dimensions of $(H, W, C) = (56, 56, 18), (28, 28, 36), (14, 14, 72)$, and $(7, 7, 144)$. We use the HRNet-v2 [52] approach, which combines four feature maps by resizing their spatial dimensions. We downsample the high image resolutions of stages 1 and 2 using local average pooling to achieve a $(H, W) = (14, 14)$ resolution. We upsample the stage 4 feature map to match the size. We concatenate the four feature maps along the channel dimension and then mix channels by a 1×1 convolution to produce an output feature map of size $(H, W, C) = (14, 14, 80)$.

To obtain a strong baseline, which has high robustness to style changes, we insert the IN layer [50] after the input (Input-IN), before stages 1,2,3,4 (Bottom-IN), and the output layer (Top-IN) of the backbone model.

We use a Multi-Layer Perceptron (MLP) and biternion representation [4] to learn the non-linear relationship between feature $\mathbf{z}$ and orientation θ. Inspired by quaternions used in computer graphics, the biternion representation expresses an angle θ as a two-dimensional (sine and cosine) vector to address the challenges of angular space discontinuities, e.g., difference between $0°$ and $359°$ should be equally treated as the difference between $0°$ and $1°$. We employ a one-hidden-layer MLP: $\mathbf{y}' \in \mathbb{R}^2 = \mathrm{FC}_2\left(\mathrm{ReLU}\left(\mathrm{FC}_1\left(\mathbf{z}\right)\right)\right)$, where $\mathrm{ReLU}(\cdot)$ is the Rectified Linear Unit, and FC_1 and FC_2 are fully connected layers with output dimensions of 256 and 2, respectively. We

[1] Implemented in timm `https://huggingface.co/timm`.

then obtain the biternion representation $\mathbf{y}$ by the L2 normalization as $\mathbf{y} = \mathbf{y}'/\|\mathbf{y}'\| = (\cos\theta, \sin\theta)^\top$. The orientation can be recovered by $\theta = \tan^{-1}(\frac{\sin\theta}{\cos\theta})$.

Loss function and training details. From several loss functions for handling angle space [4, 19, 26, 38, 56], we use the von Mises (VM) loss function [4] L_{VM}, which addresses the discontinuity of the angle by probability density of VM distribution - a normal distribution on the unit circle, due to its simplicity and effectiveness.

The VM distribution is expressed as $p_{\mathrm{VM}}(\theta|\mu,\kappa) = \frac{\exp^{\kappa\cos(\theta-\mu)}}{2\pi I_0(\kappa)}$, where μ is the mean angle of the distribution, κ is the inverse variance of the approximated Gaussian, and $I_0(\kappa)$ is the Bessel function of order 0. Note that $\cos(\theta-\mu) = \cos\theta\cos\mu - \sin\theta\sin\mu$, corresponds to an inner product of the biternion representations.

We set the ground truth angle θ^{GT} for each sample as the mean of the distribution and let $\mathbf{y}^{\mathrm{GT}}$ be its biternion representation. By inverting and appropriately scaling it, the VM loss function is defined as follows:

$$L_{\mathrm{VM}} = 1 - 2\pi I_0(\kappa)\exp(-\kappa)\cdot p_{\mathrm{VM}}(\theta|\theta^{\mathrm{GT}},\kappa) = 1 - \exp^{\kappa\left(\mathbf{y}^\top\mathbf{y}^{\mathrm{GT}}-1\right)}, \qquad (9)$$

where the presence of $\exp(\cdot)$ reduces the effect of error around the ground truth value, penalizing small mistakes less severely. Following the work [26], we set the hyperparameter $\kappa = 1$.

For all compared models, we train models 100 epochs with the Adadelta [58] optimizer, with batchsize 128 and learning rate $lr = 1.0$. Note that backpropagation including SVD is untrivial and requires much computation on CPU. Following the previous study [43], we simply use PyTorch auto-grad. We use a machine equipped with Core-i9 10980XE and a NVIDIA RTX4090 GPU which has a 24GB GPU RAM.

5.2 Datasets and Protocols

Person X dataset [46] includes 1260 individuals with orientation labels per 10 degrees (36 orientations) of 6 cameras, resulting in a total of 36 (angles) $\times$ 1266 (persons) $\times$ 6 (cameras) $= 273,456$ images. The dataset includes training/testing splits of $410/856$ individuals, respectively. Among the training data, we randomly select 150 individuals ($32,400$ images) for training and 150 individuals ($32,400$ images) as validation data. We randomly select one camera for each person in the testing dataset ($30,816$ images) for evaluation.

TUD dataset [2] consists of $5,228$ images of pedestrians. The training/validation/testing sets consist of $4732/290/309$ images, respectively. Following the previous works [18, 26], we use continuous labels created by the authors of the Ref. [18] and a coloring technique [59] to colorize monochrome images.

These datasets contain no duplicate individuals in the training and test datasets, guaranteeing that more robust methods against different clothing textures obtain better estimations. In addition, to make the changes in appearance more drastic between the training and test datasets, we perform style transfer on the test datasets. We use five styles $\mathcal{S}$ with the style transfer model STyle TRansformer (STTR) [51][2].

[2] `https://github.com/researchmm/STTR`. The five styles are the example styles of the code: dun_in_zeeland_1910, hosi, mondrian, la_muse, and stock.

Evaluation metrics. We evaluate the predictive accuracy by Accuracy 10° (Acc_{10°) and Mean Absolute Error (MAE) [22]. The former is the percentage of the predicted orientation error within 10°, and the latter is the mean prediction error.

We also evaluate the robustness by the average similarity (SIM) of the biternion representations between original and style transferred images. Let $\mathbf{y}_i$ and $\mathbf{y}_{i,s}$ be the output biternion vector of i-th test sample of original and a style $s \in \mathcal{S}$, respectively. Then SIM is defined as follows:

$$\mathrm{SIM} = 100 \times \frac{1}{|\mathcal{S}| N_{\mathrm{test}}} \sum_{i=1}^{N_{\mathrm{test}}} \sum_{s \in \mathcal{S}} \mathbf{y}_i^\top \mathbf{y}_{i,s}, \tag{10}$$

where N_{test} and $|\mathcal{S}|$ are total number of test images and styles, respectively. The inner product $\mathbf{y}_i^\top \mathbf{y}_{i,s}$ correponds $\cos(\theta_i - \theta_{i,s})$, which takes maximum and minimum values 1 and -1 when the angle difference is 0° and 180°, respectively. Therefore, SIM evaluates how similar the estimated angle θ between original and style changed images in the range $\mathrm{SIM} \in [-100\ 100]$. If the estimated angle remains unchanged by style changes, we regard the representation robust against this variation. Thus, a higher SIM indicates higher robustness against style changes.

5.3 Comparision

We compare our GCP with the following competitors:

- Full: All (CS-dim.) feature elements by reshaping the feature map.
- COOC [42]: The feature used in a person orientation estimation [26].
- BSP [32]/MPN-BSP [29]/GSP [54]: Bilinear spatial pooling methods (Sec. 4.4).
- BCP/MPN-BCP: Bilinear channel pooling methods (Sec. 4.4).
- Avg/Max/Std: The Average/Max/Std pooling and their concatenation.

The results of Table 2 indicate the following facts[3]:

(1) Spatial pooling (Avg) performs inferior to without pooling (Full) on both the ORG and Styled datasets, achieving 15.6° and 4.4° higher MAE on the PersonX Styled and the TUD Styled datasets, respectively. Also, channel pooling (Avg) outperforms spatial pooling (Avg), achieving 14.7° and 1.0° lower MAE on these datasets, respectively. These results confirm that spatial information is more important than channel information in these datasets.

(2) Channel pooling (Avg) outperforms Full on the PersonX Styled dataset, achieving 1.5° lower MAE without IN; however, it underperforms Full, achieving 0.9° higher MAE with IN. Meanwhile, it underperforms Full on the TUD Styled dataset, achieving 0.5° and 3.5° higher MAE, without and with IN, respectively. However, channel pooling (Max) outperforms Full on the TUD Styled dataset, e.g., 3.0° lower MAE. These

[3] We trained GCP and GSP with $R \in \{1, 3, 5, 8, 10,\ 20, 30, 40, 50, 60, 70\}$ and selected the model based on the lowest MAE on each validation dataset. We excluded Top-IN for Avg of spatial pooling because Top-IN makes $\mathbf{z}_{\mathrm{avg}} = \mathbf{0}$ for any input as IN normalizes feature channels zero-mean (Eq. (1)), which makes prediction infeasible. For other kinds of poolings, inserting all IN layers produces the best performance, and thus we used the all IN layers.

Table 2. Performance comparison. (a) w/o pooling, (b) spatial and (c) channel poolings, where **bold**, **blue**, and **red** numbers show the best scores in each category. **ORG** and **Styled** represents original and style transfered test datasets, respectively.

		Person X					TUD				
		ORG		Styled			ORG		Styled		
	Pool	$\text{Acc}_{10^\circ}\uparrow$	MAE ↓	$\text{Acc}_{10^\circ}\uparrow$	MAE ↓	SIM ↑	$\text{Acc}_{10^\circ}\uparrow$	MAE ↓	$\text{Acc}_{10^\circ}\uparrow$	MAE ↓	SIM ↑
(a)	Full (w/o IN)	**95.2**	**3.6**	69.3	9.7	97.0	35.6	27.1	18.2	49.0	62.5
	Full	94.9	3.6	**84.6**	**5.9**	**99.2**	**38.9**	**25.9**	**24.9**	**33.3**	**80.8**
(b)	Avg (w/o IN)	96.1	3.4	75.5	7.7	98.1	28.6	31.5	15.7	51.6	59.7
	Avg (w/o Top-IN)	56.6	16.0	47.0	21.5	90.9	31.3	28.3	23.6	37.8	71.7
	COOC [42]	**96.1**	**3.3**	83.5	6.5	98.4	26.6	34.6	22.1	39.9	74.7
	BSP [32]	95.8	3.4	84.5	6.3	98.7	31.9	26.0	25.7	33.1	81.2
	MPN-BSP [29]	94.4	4.0	81.9	6.8	98.7	29.8	33.8	19.8	46.1	63.4
	GSP [54]	95.9	**3.3**	**86.0**	**5.9**	**99.0**	**38.5**	**24.6**	**29.8**	**32.2**	**82.8**
(c)	Avg (w/o IN)	95.0	3.7	76.8	8.2	97.8	27.9	33.5	16.5	49.5	60.0
	Avg	94.0	4.0	82.2	6.8	98.6	34.3	26.6	27.3	36.8	75.2
	Std	95.0	3.7	83.4	6.5	98.9	29.6	28.0	25.6	33.3	77.4
	Max	95.2	3.6	82.7	6.6	98.8	31.4	25.9	26.4	30.3	85.7
	Avg+Std+Max	96.0	3.3	84.9	6.0	98.9	27.7	32.0	21.8	40.6	77.1
	BCP	95.2	3.5	**88.2**	**5.4**	**99.3**	35.5	24.2	27.6	32.1	79.3
	MPN-BCP	95.6	3.4	87.5	5.7	98.9	37.5	24.4	**33.1**	26.3	84.6
	GCP (w/o IN)	**96.0**	**3.2**	79.4	7.4	98.1	**42.5**	25.7	26.0	34.1	78.9
	GCP	95.6	3.4	87.2	5.6	99.2	35.8	**21.6**	**33.1**	**25.5**	**87.6**

results verify that the accuracy of the orientation estimation could be increased by summarizing redundant channel information. Meanwhile, the limited discriminative ability of naive channel pooling suffers from insufficient performance improvements.

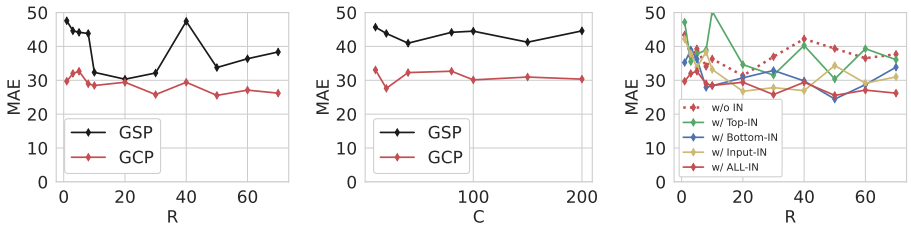
(3) Among spatial poolings, bilinear pooling methods (BSP/MPN-BSP/GSP) show high performance. Especially, GSP performs the best on the Styled datasets, achieving 0.4° and 0.9° lower MAE than BSP on the Person X Styled and the TUD Styled datasets, respectively. These results are due to neglecting the singular values that are affected by style changes.

(4) Comparing bilinear spatial pooling (BSP/MPN-BSP/GSP) and bilinear channel pooling (BCP/MPN-BCP/GCP), the latter methods demonstrate superior performance. For example, BCP achieves 0.9° and 1.0° lower MAE than BSP on the Person X Styled and the TUD Styled datasets, respectively. These results confirm the robustness of the spatial bases compared with the channel bases against style changes.

(5) GCP performs the second best on the Person-X Styled dataset, achieving 0.2° higher MAE than BCP. Meanwhile, GCP significantly outperforms other methods on the TUD Styled dataset, achieving 6.6° lower MAE than BCP. These results verify that GCP has a strong discriminative ability in channel pooling and higher robustness to style changes due to selecting the leading singular vectors and neglecting the singular values.

5.4 Parameter Sensitivity Analysis and Ablation Study

Number of bases/channels. Fig. 3 (a) and (b) compare the performance of GSP with GCP when varying the number R of bases and C of channels, respectively. Here C is the output dimension of the 1×1 convolution layer. For (a) and (b), we used $C = 80$



(a) Number of bases (b) Number of channels (c) Places of IN

Fig. 3. Sensitivity to the parameters and ablation study on the TUD Styled dataset.

Table 3. Comparison of spatial size of the input feature map for GCP .

	Person X					TUD				
	ORG		Styled			ORG		Styled		
$(H \times W)$	$\text{Acc}_{10^\circ} \uparrow$	MAE ↓	$\text{Acc}_{10^\circ} \uparrow$	MAE ↓	SIM ↑	$\text{Acc}_{10^\circ} \uparrow$	MAE ↓	$\text{Acc}_{10^\circ} \uparrow$	MAE ↓	SIM ↑
(28×28)	95.5	**3.3**	**87.5**	**5.6**	99.1	**40.6**	24.2	30.1	30.9	82.6
(14×14)	**95.6**	3.4	87.2	**5.6**	**99.2**	35.8	**21.6**	**33.1**	**25.5**	**87.6**
(7×7)	**95.6**	3.4	85.4	6.1	98.8	36.9	21.7	31.9	28.9	83.5

and $R = 5$ respectively. GCP performs better as the number R increases. Also, the performance of GCP is more stable than GSP with lower MAEs in any R and C.

Place of IN. Fig. 3 (c) compares the places of the IN layer: Top-IN, Bottom-IN, Input-IN, and ALL-IN (all of them). The results indicate that inserting IN anywhere improves the performance of the styled dataset, and ALL-IN tends to perform best.

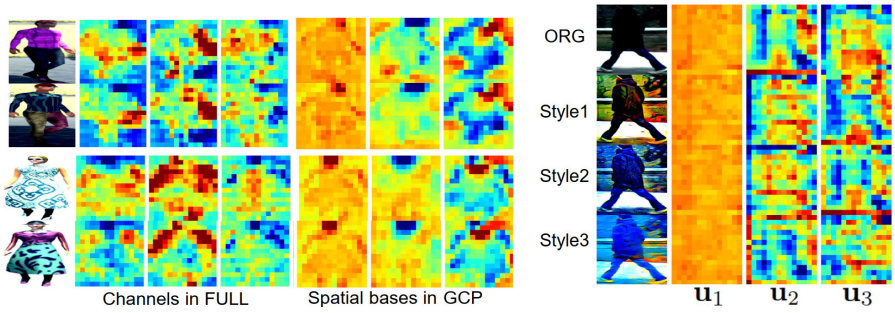
Spatial size. Table 3 compares the spatial sizes of the feature maps. For 28×28 and 7×7, we resized the high-resolution and low-resolution maps similarly as 14×14. The results indicate that GCP is effective when the spatial size is larger than 14×14.

5.5 Qualitative Analysis

Robustness examples. Figure 4 shows several example images to demonstrate robustness. In this example, we omitted IN so that the difference became clear, and we used GCP learned with $R = 5$. In (a), we showed the feature map of the first 3 channels of Full and the leading 3 spatial bases of GCP. We observe that the variation of clothing textures affects several channel activations of Full. Meanwhile, the spatial bases of GCP tend to be consistent. In (b), we observe that the style variation alters the order of similar bases; however, GCP is insensitive to this order.

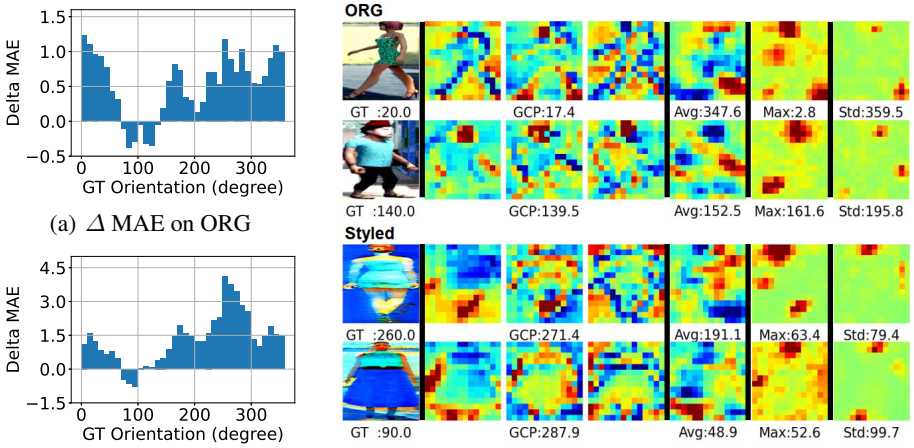
Analysis per orientation. Fig. 5 (a) and (b) show performance gain of GCP from Avg in terms of MAE obtained by Δ MAE_θ = MAE_θ(Avg) - MAE_θ(GCP), where MAE_θ is MAE calculated per ground truth orientation θ (higher Δ_θ MAE is better). We used GCP ($R = 5$) with ALL-IN[4]. We see that the improvement on 20° and 140° are the highest on the original test dataset. On the Styled dataset, 260° is the highest. Fig. 5 (c) shows several estimated images on these orientations. We observe that the global

[4] Without IN, the elements of the first spatial basis were mostly positive values as shown in the hot colors of Fig. 1 and Fig. 4. Meanwhile, IN changed this property in Fig. 5 (c)



(a) Channel of Full vs. spatial bases (b) Robustness against style changes

Fig. 4. Example images of robustness.



(a) Δ MAE on ORG (b) Δ MAE on Styled (c) Resulting examples with predicted orientations

Fig. 5. Qualitative results on Person X. GT stands for ground truth orientation.

human body shape is well expressed in GCP in several basis vectors compared to other channel pooling methods. On the styled dataset, Max and Std poolings wrongly detected a front face for the back pose ($260°$), and thus the estimation errors are significant. The bottom example in Fig. 5 (c) shows the opposite case, where GCP failed to estimate the frontal pose ($90°$) to the opposite orientation ($287.9°$). The failure may be due to GCP weakened the impact of local clues, such as the front head. However, Fig. 5 (b) implies that the failure cases of GCP are much fewer than the success cases.

6 Conclusions

We have proposed a Grassmann Channel Pooling (GCP), which summarizes a feature map as a linear subspace of its spatial bases for robust shape representation. GCP has

invariance to permutations of feature channels and variations in activation magnitude, enabling it to extract robust features for style shifts. Also, GCP corresponds to a dual form of GSP [54], which has an interesting connection to existing bilinear pooling methods. Furthermore, GCP works complementarily with Instance Normalization [50], which enhances domain generalization. Experiments conducted on the PersonX and TUD datasets confirmed the superior performance of GCP compared to other pooling methods.

Though GCP has the above advantages, it struggles with handling a higher-resolution input feature map, e.g., 56×56, directly due to its high computational cost. Also, GCP would be vulnerable when large spatial misalignment exists within the same pose, e.g., MEBOW dataset [56]. To address these limitations, we require advanced architectures. One possible design would be to apply GCP as local pooling and then integrate them globally. In addition, we would like to test the applicability of GCP to other vision tasks beyond the person orientation estimation. A careful selection of the target tasks is mandatory as GCP requires spatial alignment within the same visual concept.

Acknowledgements This work is supported by JSPS KAKENHI JP20K11890.

References

1. Absil, P., Mahony, R.E., Sepulchre, R.: Optimization algorithms on matrix manifolds. Princeton University Press (2008)
2. Andriluka, M., Roth, S., Schiele, B.: Monocular 3D pose estimation and tracking by detection. In: CVPR (2010)
3. Ben-David, S., Blitzer, J., Crammer, K., Pereira, F.: Analysis of representations for domain adaptation. In: NeurIPS. pp. 137–144 (2006)
4. Beyer, L., Hermans, A., Leibe, B.: Biternion Nets: Continuous head pose regression from discrete training labels. In: GCPR (2015)
5. Boureau, Y.L., Ponce, J., LeCun, Y.: A theoretical analysis of feature pooling in visual recognition. In: ICML. pp. 111–118 (2010)
6. Cai, H., Li, J., Hu, M., Gan, C., Han, S.: EfficientViT: Lightweight multi-scale attention for high-resolution dense prediction. In: ICCV (2023)
7. Chang, T., Yang, P., Luo, X., Ji, P., Wang, M.: Learning style-invariant robust representation for generalizable visual instance retrieval. In: ACMMM (2023)
8. Chen, L., Zhang, Y., Song, Y., van den Hengel, A., Liu, L.: Domain generalization via rationale invariance. In: ICCV (2023)
9. Chen, T.W., Yoshinaga, M., Gao, H., Tao, W., Wen, D., Liu, J., Osa, K., Kato, M.: Condensation-Net: Memory-efficient network architecture with cross-channel pooling layers and virtual feature maps. In: CVPR Workshop (2019)
10. Chen, X., Wang, S., Wang, J., Long, M.: Representation subspace distance for domain adaptation regression. In: ICML (2021)
11. Edelman, A., Arias, T., Smith, S.T.: The geometry of algorithms with orthogonality constraints. SIAM Journal on Matrix Analysis and Applications **20**(2), 303–353 (1998)
12. Fan, X., Wang, Q., Ke, J., Yang, F., Gong, B., Zhou, M.: Adversarially adaptive normalization for single domain generalization. In: CVPR (2021)
13. Fang, P., Zhou, J., Roy, S.K., Petersson, L., Harandi, M.: Bilinear attention networks for person retrieval. In: ICCV (2019)

14. Gatys, L.A., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: NuerIPS (2015)
15. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR (2016)
16. Geng, Z., Guo, M., Chen, H., Li, X., Wei, K., Lin, Z.: Is attention better than matrix decomposition? In: ICLR (2021)
17. Hamm, J., Lee, D.D.: Grassmann discriminant analysis: A unifying view on subspace-based learning. In: ICML (2008)
18. Hara, K., Chellappa, R.: Growing regression tree forests by classification for continuous object pose estimation. International Journal of Computer Vision **122**(2), 293–312 (2017)
19. Hara, K., Vemulapalli, R., Chellappa, R.: Designing deep convolutional neural networks for continuous object orientation estimation. In: arXiv:1702.01499 (2017)
20. Huang, Y., Sun, X., Lu, M., Xu, M.: Channel-max, channel-drop and stochastic max-pooling. In: CVPR Workshop (2015)
21. Huang, Z., Gool, L.V.: Building deep networks on Grassmann manifolds. In: AAAI (2018)
22. Jadon, A., Patil, A.: A comprehensive survey of evaluation techniques for recommendation systems (2024), https://arxiv.org/abs/2312.16015
23. Jegou, H., Douze, M., Schmid, C.: On the burstiness of visual elements. In: CVPR (2009)
24. Jia, J., Ruan, Q., Hospedales, T.M.: Frustratingly easy person re-identification: Generalizing person re-id in practice. In: BMVC (2019)
25. Jin, X., Lan, C., Zheng, W., Chen, Z.: Style normalization and restitution for domain generalization and adaptation. IEEE Transactions on Multimedia **24**, 3636–3651 (2022)
26. Kim, S.S., Gwak, I.Y., Lee, S.W.: Coarse-to-fine deep learning of continuous pedestrian orientation based on spatial co-occurrence feature. IEEE Trans. on ITS **21**(6), 2522–2533 (2020)
27. Koniusz, P., Zhang, H., Porikli, F.: A deeper look at power normalizations. In: CVPR (2018)
28. Li, H., Ye, M., Du, B.: WePerson: Learning a generalized re-identification model from all-weather virtual data. In: ACMMM. pp. 3115–3123. ACM (2021)
29. Li, P., Xie, J., Wang, Q., Zuo, W.: Is second-order information helpful for large-scale visual recognition? In: CVPR (2017)
30. Lin, T.Y., Maji, S.: Improved bilinear pooling with CNNs. In: BMVC (2017)
31. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear CNN models for fine-grained visual recognition. In: CVPR (2015)
32. Lin, T.Y., RoyChowdhury, A., Maji, S.: Bilinear convolutional neural networks for fine-grained visual recognition. IEEE Trans. on PAMI **40**(6), 1309–1322 (2018)
33. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. In: CVPR (2022)
34. Ma, Z., Chang, D., Xie, J., Ding, Y., Wen, S., Li, X., Si, Z., Guo, J.: Fine-grained vehicle classification with channel max pooling modified CNNs. IEEE Transactions on Vehicular Technology **68**(4), 3324–3233 (2019)
35. Marcos, D., Volpi, M., Tuia, D.: Learning rotation invariant convolutional filters for texture classification. In: ICPR (2016)
36. Nam, H., Kim, H.E.: Batch-instance normalization for adaptively style-invariant neural networks. In: NuerIPS (2018)
37. (Ning)Xia, B., Gong, Y., Zhang, Y., Poellabauer, C.: Second-order non-local attention networks for person re-identification. In: ICCV (2019)
38. Prokudin, S., Gehler, P., Nowozin, S.: Deep directional statistics: Pose estimation with uncertainty quantification. In: ECCV (2018)
39. Raza, M., Chen, Z., Rehman, S.U., Wang, P., Bao, P.: Appearance based pedestrians' head pose and body orientation estimation using deep learning. Neurocomputing **272**(10), 647–659 (2018)

40. Roy, K., Simon, C., Moghadam, P., Harandi, M.: Subspace distillation for continual learning. Neural Netw. **167**(C), 65–79 (2024)
41. Seo, S., Suh, Y., Kim, D., Kim, G., Han, J., Han, B.: Learning to optimize domain specific normalization for domain generalization. In: ECCV (2020)
42. Shih, Y.F., Yeh, Y.M., Lin, Y.Y., Weng, M.F., Lu, Y.C., Chuang, Y.Y.: Deep co-occurrence feature learning for visual object recognition. In: CVPR (2017)
43. Simon, C., Koniusz, P., Nock, R., Harandi, M.: Adaptive subspaces for few-shot learning. In: CVPR (2020)
44. Souza, L.S., Sogi, N., Gatto, B.B., Kobayashi, T., Fukui, K.: Grassmannian learning mutual subspace method for image set recognition. Neurocomputing **517**(14), 20–33 (2023)
45. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: CVPR (2019)
46. Sun, X., Zheng, L.: Dissecting person re-identification from the viewpoint of viewpoint. In: CVPR (2019)
47. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR (2015)
48. Tomasini, U.M., Petrini, L., Cagnetta, F., Wyart, M.: How deep convolutional neural networks lose spatial information with training. Machine Learning: Science and Technology **4**, 1–18 (2023)
49. Turaga, P.K., Veeraraghavan, A., Srivastava, A., Chellappa, R.: Statistical computations on Grassmann and Stiefel manifolds for image and video-based recognition. IEEE Trans. Pattern Anal. Mach. Intell. **33**(11), 2273–2286 (2011)
50. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Instance normalization: The missing ingredient for fast stylization. arXiv **arXiv:1607.08022** (2016)
51. Wang, J., Yang, H., Fu, J., Yamasaki, T., Guo, B.: Fine-grained image style transfer with visual transformers. In: ACCV (2022)
52. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep high-resolution representation learning for visual recognition. IEEE Trans. Pattern Anal. Mach. Intell. **43**(10), 3349–3364 (2021)
53. Wang, Z., Li, W., Kao, Y., Zou, D., Wang, Q., Ahn, M., Hong, S.: HCR-Net: A hybrid of classification and regression network for object pose estimation. In: IJCAI (2018)
54. Wei, X., Zhang, Y., Gong, Y., Zhang, J., Zheng, N.: Grassmann pooling as compact homogeneous bilinear pooling for fine-grained visual classification. In: ECCV (2018)
55. Woo, S., Park, J., Lee, J.Y., Kweon, I.S.: CBAM: Convolutional block attention module. In: ECCV (2018)
56. Wu, C., Chen, Y., Luo, J., Su, C.C., Dawane, A., Hanzra, B., Deng, Z., Liu, B., Wang, J.Z., Kuo, C.: MEBOW: Monocular estimation of body orientation in the wild. In: CVPR (2020)
57. Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: HRFormer: High-Resolution vision transformer for dense predict. In: NuerIPS. pp. 7281–7293 (2021)
58. Zeiler, M.D.: ADADELTA: An adaptive learning rate method. CoRR **abs/1212.5701** (2012)
59. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: ECCV (2016)
60. Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C.: Domain generalization: A survey. IEEE Trans. Pattern Anal. Mach. Intell. **45**(4), 4396–4415 (2023)