Equivariant Multi-View Networks

Carlos Esteves[*], Yinshuang Xu[*], Christine Allen-Blanchette, Kostas Daniilidis
GRASP Laboratory, University of Pennsylvania
{machc,xuyin,allec,kostas}@seas.upenn.edu

Abstract

Several popular approaches to 3D vision tasks process multiple views of the input independently with deep neural networks pre-trained on natural images, achieving view permutation invariance through a single round of pooling over all views. We argue that this operation discards important information and leads to subpar global descriptors. In this paper, we propose a group convolutional approach to multiple view aggregation where convolutions are performed over a discrete subgroup of the rotation group, enabling, thus, joint reasoning over all views in an equivariant (instead of invariant) fashion, up to the very last layer. We further develop this idea to operate on smaller discrete homogeneous spaces of the rotation group, where a polar view representation is used to maintain equivariance with only a fraction of the number of input views. We set the new state of the art in several large scale 3D shape retrieval tasks, and show additional applications to panoramic scene classification.

1. Introduction

The proliferation of large scale 3D datasets for objects [39, 3] and whole scenes [2, 8] enables training of deep learning models producing global descriptors that can be applied to classification and retrieval tasks.

The first challenge that arises is how to represent the inputs. Despite numerous attempts with volumetric [39, 24], point-cloud [27, 32] and mesh-based [23, 26] representations, using multiple views of the 3D input allows switching to the 2D domain where all the recent image based deep learning breakthroughs (*e.g.* [15]) can be directly applied, facilitating state of the art performance [33, 20].

Multi-view (MV) based methods require some form of view-pooling, which can be (1) pixel-wise pooling over some intermediate convolutional layer [33], (2) pooling over the final 1D view descriptors [34], or (3) combining the final logits [20], which can be seen as independent voting. These operations are usually invariant to view permutations.

Our key observation is that conventional view pooling is performed before any joint processing of the set of views and will inevitably discard useful features, leading to subpar descriptors. We solve the problem by first realizing that each view can be associated with an element of the rotation group $\mathbf{SO}(3)$, so the natural way to combine multiple views is as a function on the group. A traditional CNN is applied to obtain view descriptors that compose this function. We design a group-convolutional network (G-CNN, inspired by [5]) to learn representations that are equivariant to transformations from the group. This differs from the invariant representations obtained through usual view-pooling that discards information. We obtain invariant descriptors useful for classification and retrieval by pooling over the last G-CNN layer. Our G-CNN has filters with localized support on the group and learns hierarchically more complex representations as we stack more layers and increase the receptive field.

We take advantage of the finite nature of multiple views and consider finite rotation groups like the icosahedral, in contrast with [6, 10] which operate on the continuous group. To reduce the computational cost of processing one view per group element, we show that by considering views in canonical coordinates with respect to the group of in-plane dilated rotations (log-polar coordinates), we can greatly reduce the number of views and obtain an initial representation on a homogeneous space (H-space) that can be lifted via correlation, while maintaining equivariance.

We focus on 3D shapes but our model is applicable to any task where multiple views can represent the input, as demonstrated by an experiment on panoramic scenes.

Figure 1 illustrates our model. Our contributions are:

- We introduce a novel method of aggregating multiple views whether "outside-in" for 3D shapes or "inside-out" for panoramic views. Our model exploits the underlying group structure, resulting in equivariant features that are functions on the rotation group.
- We introduce a way to reduce the number of views while maintaining equivariance, via a transformation to

[*] Equal contribution.
http://github.com/daniilidis-group/emvn

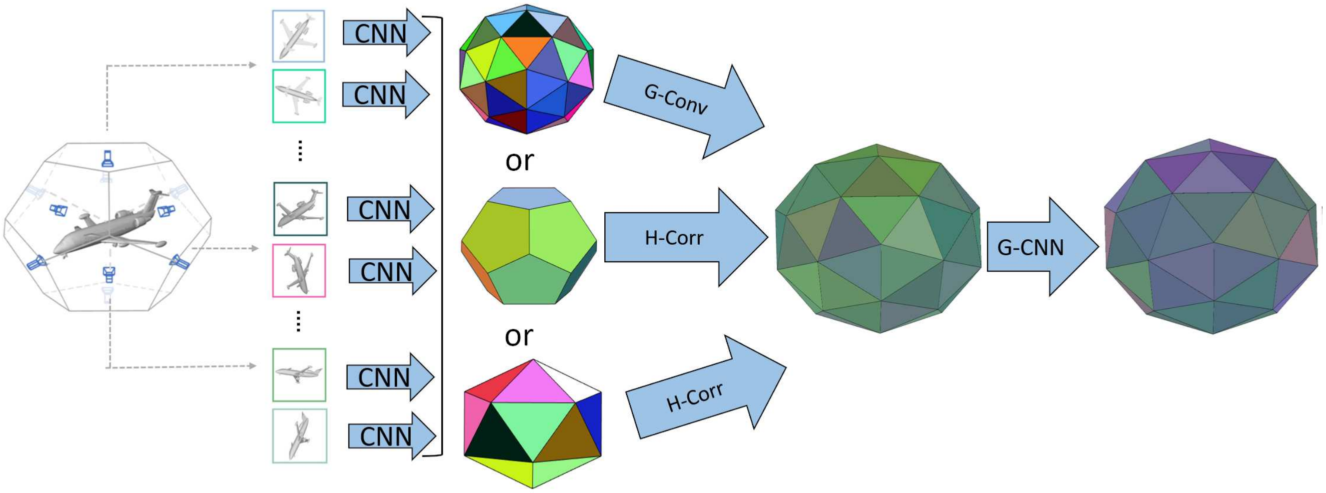

Figure 1: Our Equivariant Multi-View Network aggregates multiple views as functions on rotation groups that are processed via group convolutions. This guarantees equivariance to 3D rotations and allows jointly reasoning over all views, leading to superior shape descriptors. Vector-valued functions on the icosahedral group are shown on the pentakis dodecahedron, and functions on the corresponding homogeneous spaces (H-spaces) are shown on the dodecahedron and icosahedron. Each view is first processed by a CNN and resulting descriptors are associated with a group (or H-space) element. When views are identified with an H-space, the first operation is a correlation that lifts features to the group. Once we have an initial representation on the group, a Group-CNN can be applied.

canonical coordinates of in-plane rotation followed by homogeneous space convolution.

- We explore the finite rotation groups and homogeneous spaces and present a discrete G-CNN model on the largest group to date, the icosahedral group. We further explore the concept of filter localization for this group.
- We achieve state of the art performance on multiple shape retrieval benchmarks, both in canonical poses and perturbed with rotations, and show applications to panoramic scene classification.

2. Related work

3D shape analysis Performance of 3D shape analysis is heavily dependent on the input representation. The main representations are volumetric, point cloud and multi-view.

Early examples of volumetric approaches are [3], which introduced the ModelNet dataset and trained a 3D shape classifier using a deep belief network on voxel representations; and [24], which presents a standard architecture with 3D convolutional layers followed by fully connected layers.

Su *et al*. [33] realized that by rendering multiple views of the 3D input one can transfer the power of image-based CNNs to 3D tasks. They show that a conventional CNN can outperform the volumetric methods even using only a single view of the input, while a multi-view (MV) model further improves the classification accuracy.

Qi *et al*. [28] study volumetric and multi-view methods and propose improvements to both; Kanezaki *et al*. [20] introduces an MV approach that achieves state-of-the-art classification performance by jointly predicting class and pose, though without explicit pose supervision.

GVCNN [12] attempts to learn how to combine different view descriptors to obtain a view-group-shape representation; they refer to arbitrary combinations of features as “groups”. This differs from our usage of the term “group” which is the algebraic definition.

Point-cloud based methods [27] achieve intermediate performance between volumetric and multi-view, but are much more efficient computationally. While meshes are arguably the most natural representation and widely used in computer graphics, only limited success has been achieved with learning models operating directly on them [23, 26].

In order to better compare 3D shape descriptors we will focus on the retrieval performance. Recent approaches show significant improvements on retrieval: You *et al*. [41] combines point cloud and MV representations; Yavartanoo *et al*. [40] introduces multi-view stereographic projection; and Han *et al*. [14] implements a recurrent MV approach.

We also consider more challenging tasks on rotated ModelNet and SHREC’17 [29] retrieval challenge which contains rotated shapes. The presence of arbitrary rotations motivates the use of equivariant representations.

Equivariant representations A number of workarounds have been introduced to deal with 3D shapes in arbitrary orientations. Typical examples are training time rotation augmentation and/or test time voting [28] and learning an initial rotation to a canonical pose [27]. The view-pooling in [33] is invariant to permutations of the set of input views.

A principled way to deal with rotations is to use representations that are equivariant by design. There are mainly three ways to embed equivariance into CNNs. The first way is to constrain the filter structure, which is similar to Lie generator based approach [30, 17]. Worral *et al.* [38] take advantage of circle harmonics to have both translational and 2D rotational equivariance into CNNs. Similarly, Thomas *et al.* [35] introduce a tensor field to keep translational and rotational equivariance for 3D point clouds.

The second way is through a change of coordinates; [11, 18] take the log-polar transform of the input and transfer rotational and scaling equivariance about a single point to translational equivariance.

The third way is to make use of an equivariant filter orbit. Cohen and Welling propose group convolution (G-CNNs) with the square rotation group [5], later extended to the hexagon [19]. Worrall and Brostow [37] proposed CubeNet using Klein's Four-group on 3D voxelized data. Winkels *et al.* [36] implement 3D group convolution on Octahedral symmetry group for volumetric CT images. Cohen *et al.* [7] very recently considered functions on the icosahedron, however their convolutions are on the cyclic group and not on the icosahedral as ours. Esteves *et al.* [10] and Cohen *et al.* [6] focus on the infinite group $\mathbf{SO}(3)$, and use the spherical harmonic transform for the exact implementation of the spherical convolution or correlation. The main issue with these approaches is that the input spherical representation does not capture the complexity of an object's shape; they are also less efficient and face bandwidth challenges.

3. Preliminaries

We seek to leverage symmetries in data. A *symmetry* is an operation that preserves some structure of an object. If the object is a discrete set with no additional structure, each operation can be seen as a permutation of its elements.

The term *group* is used in its classic algebraic definition of a set with an operation satisfying the closure, associativity, identity, and inversion properties. A transformation group like a permutation is the "missing link between abstract group and the notion of symmetry" [25].

We refer to *view* as an image taken from an oriented camera. This differs from *viewpoint* that refers to the optical axis direction, either *outside-in* for a moving camera pointing at a fixed object, or *inside-out* for a fixed camera pointing at different directions. Multiple *views* can be taken from the same *viewpoint*; they are related by in-plane rotations.

Equivariance Representations that are equivariant by design are an effective way to exploit symmetries. Consider a set $\mathcal{X}$ and a transformation group G. For any $g \in G$, we can define group action applied on the set, $\mathcal{T}_g^{\mathcal{X}} : \mathcal{X} \to \mathcal{X}$, which has property of homomorphism, $\mathcal{T}_g^{\mathcal{X}} \mathcal{T}_h^{\mathcal{X}} = \mathcal{T}_{gh}^{\mathcal{X}}$. Consider a map $\Phi : \mathcal{X} \to \mathcal{Y}$. We say Φ is equivariant to G if

$$\Phi(\mathcal{T}_g^{\mathcal{X}}(x)) = \mathcal{T}_g^{\mathcal{Y}}(\Phi(x)), \quad \forall x \in \mathcal{X}, g \in G. \tag{1}$$

In the context of CNNs, $\mathcal{X}$ and $\mathcal{Y}$ are sets of input and feature representations, respectively. This definition encompasses the case when $\mathcal{T}_g^{\mathcal{Y}}$ is the identity, making Φ invariant to G and discarding information about g. In this paper, we are interested in non-degenerate cases that preserve information.

Convolution on groups We represent multiple views as a functions on a group and seek equivariance to the group, so group convolution (G-Conv) is the natural operation for our method. Let us recall planar convolution between $f, h \colon \mathbb{R}^2 \mapsto \mathbb{R}$, which is the main operation of CNNs:

$$(f * h)(y) = \int_{x \in \mathbb{R}^2} f(x) h(y - x)\, dx. \tag{2}$$

It can be seen as an operation over the group of translations on the plane, where the group action is addition of coordinate values; it is easily shown to be equivariant to translation. This can be generalized to any group G and $f, h \colon G \mapsto \mathbb{R}$,

$$(f * h)(y) = \int_{g \in G} f(g) h(g^{-1} y)\, dg, \tag{3}$$

which is equivariant to group actions from G.

Convolution on homogeneous spaces For efficiency, we may relax the requirement of one view per group element and consider only one view per element of a homogeneous space of lower cardinality. For example, we can represent the input on the 12 vertices of the icosahedron (an H-space), instead of on the 60 rotations of the icosahedral group.

A homogeneous space $\mathcal{X}$ of a group G is defined as a space where G acts transitively: for any $x_1, x_2 \in \mathcal{X}$, there exists $g \in G$ such that $x_2 = g x_1$.

Two convolution-like operations can be defined between functions on homogeneous spaces $f, h \colon \mathcal{X} \mapsto \mathbb{R}$:

$$(f * h)(y) = \int_{g \in G} f(g\eta) h(g^{-1} y)\, dg, \tag{4}$$

$$(f \star h)(g) = \int_{x \in \mathcal{X}} f(gx) h(x)\, dx, \tag{5}$$

where $\eta \in \mathcal{X}$ is an arbitrary canonical element. We denote (4) "homogeneous space convolution" (H-Conv), and (5) "homogeneous space correlation" (H-Corr). Note that convolution produces a function on the homogeneous space $\mathcal{X}$ while correlation lifts the output to the group G.

We refer to [22, 4] for expositions on group and homogeneous space convolution in the context of neural networks.

Finite rotation groups Since our representation is a finite set of views that can be identified with rotations, we will deal with finite subgroups of the rotation group $\mathbf{SO}(3)$. A finite subgroup of $\mathbf{SO}(3)$ can be the cyclic group $\mathcal{C}_k$ of multiples of $2\pi/k$, the dihedral group $\mathcal{D}_k$ of symmetries of a regular k-gon, the tetrahedral, octahedral, or icosahedral group [1].

Our main results are on the icosahedral group $\mathcal{I}$, the 60-element non-abelian group of symmetries of the icosahedron (illustrated in the supplementary material). The symmetries can be divided in sets of rotations around a few axes. For example, there are 5 rotations around each axis passing through vertices of the icosahedron or 3 rotations around each axis passing through its faces centers.

Equivariance via canonical coordinates Some configurations produce views that are related by in-plane rotations. We leverage this to reduce the number of required views by obtaining rotation invariant view descriptors through a change to canonical coordinates followed by a CNN.

Segman *et al*. [30] show that changing to a canonical coordinate system allows certain transformations of the input to appear as translations of the output. For the group of dilated rotations on the plane (isomorphic to $\mathbf{SO}(2) \times \mathbb{R}^+$), canonical coordinates are given by the log-polar transform.

Since planar convolutions are equivariant to translation, converting an image to log-polar and applying a CNN results in features equivariant to dilated rotation, which can be pooled to invariant descriptors on the last layer [11, 18].

4. Method

Our first step is to obtain $|G|$ views of the input where each view x_i is associated with a group element $g_i \in G$[1]. Each view is fed to a CNN Φ_1, and the 1D descriptors extracted from the last layer (before projection into the number of classes) are combined to form a function on the group $y \colon G \mapsto \mathbb{R}^n$, where $y(g_i) = \Phi_1(x_i)$. A group convolutional network (G-CNN) Φ_2 operating on G is then used to process y, and global average pooling on the last layer yields an invariant descriptor that is used for classification or retrieval. Training is end-to-end. Figure 1 shows the model.

The MVCNN with late-pooling from [20], which outperforms the original [33], is a special case of our method where Φ_2 is the identity and the descriptor is y averaged over G.

4.1. View configurations

There are several possible view configurations of icosahedral symmetry, basically consisting of vertices or faces of solids with the same symmetry. Two examples are associating viewpoints with faces/vertices of the icosahedron, which are equivalent to the vertices/faces of its dual, the dodecahedron. These configurations are based on platonic solids, which guarantee a uniform distribution of viewpoints. By selecting viewpoints from the icosahedron faces, we obtain 20 sets of 3 views that differ only by 120 deg in plane rotations; we refer to this configuration as 20 × 3. Similarly, using the dodecahedron faces we obtain the 12 × 5 configuration.

In the context of 3D shape analysis, multiple viewpoints are useful to handle self-occlusions and ambiguities. Views that are related by in-plane rotations are redundant in this sense, but necessary to keep the group structure.

To minimize redundancy, we propose to associate viewpoints with the 60 vertices of the truncated icosahedron (which has icosahedral symmetry). There is a single view per viewpoint in this configuration. This is not a uniformly spaced distribution of viewpoints, but the variety is beneficial. Figure 3 shows some view configurations we considered.

Note that our configurations differ from both the 80-views from [33] and 20 from [20] which are not isomorphic to any rotation group. Their 12-views configuration is isomorphic to the more limited cyclic group.

4.2. Group convolutional networks

The core of the group convolutional part of our method is the discrete version of (3). A group convolutional layer with c_i, c_j channels in the input and output and nonlinearity σ is then given by

$$f_j^{\ell+1}(y) = \sigma\left(\sum_{i=1}^{c_i}\sum_{g\in G} f_i^{\ell}(g)h_{ij}(g^{-1}y)\right), \qquad (6)$$

where f_i^ℓ is the channel i at layer ℓ and h_{ij} is the filter between channels i and j, where $1 \le j \le c_j$. This layer is equivariant to actions of G.

Our most important results are on the icosahedral group $\mathcal{I}$ which has 60 elements and is the largest discrete subgroup of the rotation group $\mathbf{SO}(3)$. To the best of our knowledge, this is the largest group ever considered in the context of discrete G-CNNs. Since $\mathcal{I}$ only coarsely samples $\mathbf{SO}(3)$, equivariance to arbitrary rotations is only approximate. Our results show, however, that the combination of invariance to local deformations provided by CNNs and exact equivariance by G-CNNs is powerful enough to achieve state of the art performance in many tasks.

When considering the group $\mathcal{I}$, inputs to Φ_2 are $60 \times n$ where n is the number of channels in the last layer of Φ_1 (n=512 for ResNet-18). There are $c_i \times c_j$ filters per layer each with up to the same cardinality of the group.

We can visualize both filters and feature maps as functions on the faces of the pentakis dodecahedron, which is the dual polyhedron of the truncated icosahedron. It has icosahedral symmetry and 60 faces that can be identified with elements of the group. The color of the face associated with $g_i \in \mathcal{I}$

[1] Alternatively, we can use $|\mathcal{X}|$ views for a homogeneous space $\mathcal{X}$ as shown in 4.3.

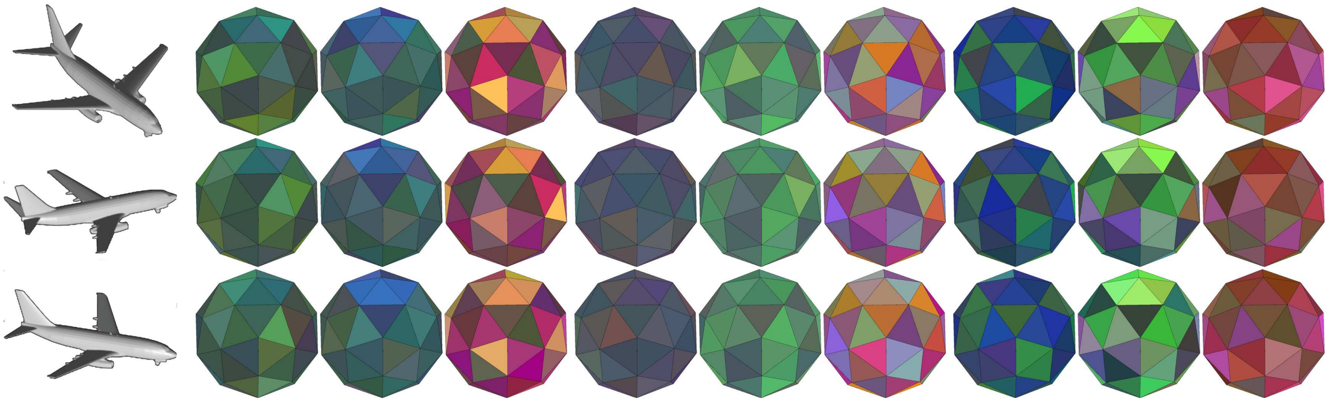

Figure 2: Features learned by our method are visualized on the pentakis dodecahedron, which has icosahedral symmetry so its 60 faces can be identified with elements of the discrete rotation group $\mathcal{I}$. Columns show learned features from different channels/layers. The first two rows are related by a rotation of 72 deg in $\mathcal{I}$. Equivariance is exact in this case, as can be verified by the feature maps rotating around the polar axis (notice how the top 5 cells shift one position). The first and third row are related by a rotation of 36 deg around the same axis, which is in the midpoint between two group elements. Equivariance is approximate in this case, and features are a mixture of the two above.

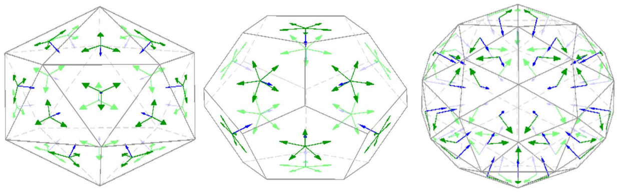

Figure 3: Outside-in camera configurations considered. Left to right: 20×3, 12×5, and 60×1. Blue arrows indicate the optical axis and green, the camera up direction. Object is placed at the intersection of all optical axes. Only the 60×1 configuration avoids views related by in-plane rotations.

reflects $f(g_i)$, which is vector valued. Figure 2 shows some equivariant feature maps learned by our method.

4.3. Equivariance with fewer views

As illustrated in Figure 3, the icosahedral symmetries can be divided in sets of rotations around a few axes. If we arrange the cameras such that they lie on these axes, images produced by each camera are related by in-plane rotations.

As shown in Section 3, converting one image to canonical coordinates can transform in-plane rotations in translations. We'll refer to converted images as "polar images". Since fully convolutional networks can produce translation-invariant descriptors, by applying them to polar images we effectively achieve invariance to in-plane rotations [11, 18], which makes only one view per viewpoint necessary. These networks require circular padding in the angular dimension.

When associating only a single view per viewpoint, the input is on a space of points instead of a group of rotations[2]. In fact, the input is a function on a homogeneous space of the group; concretely, for the view configurations we consider, it is on the icosahedron or dodecahedron vertices.

We can apply discrete versions of convolution and correlation on homogeneous spaces as defined in Section 3:

$$^{*}f_j^{\ell+1}(y) = \sigma\left(\sum_{i=1}^{c_i}\sum_{g\in G} f_i^{\ell}(g\eta)h_{ij}(g^{-1}y)\right), \qquad (7)$$

$$^{\star}f_j^{\ell+1}(g) = \sigma\left(\sum_{i=1}^{c_i}\sum_{x\in \mathcal{X}} f_i^{\ell}(gx)h_{ij}(x)\right). \qquad (8)$$

The benefit of this approach is that since it uses 5x (3x) fewer views when starting from the 12×5 (20×3) configuration, it is roughly 5x (3x) faster as most of the compute is done before the G-CNN. The disadvantage is that learning from polar images can be challenging. Figure 4 shows one example of polar images produced from views.

When inputs are known to be aligned (in canonical pose), an equivariant intermediate representation is not necessary; in this setting, we can use the same method to reduce the number of required views, but without the polar transform.

4.4. Filter localization

G-CNN filters are functions on G, which can have up to $|G|$ entries. Results obtained with deep CNNs during the past few years show the benefit from limited support filters (many architectures use 3×3 kernels throughout). The advantages are two-fold: (1) convolution with limited support

[2] They are isomorphic for the 60×1 configuration.

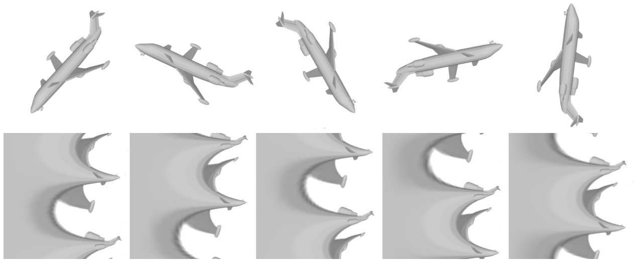

Figure 4: One subset of in-plane related views from the 12×5 configuration and correspondent polar images. Note how the polar images are related by circular vertical shifts so their CNN descriptors are approximately invariant to the in-plane rotation. There are 12 such subsets for the 12×5 configuration and 20 for the 20×3; this allows us to maintain equivariance with 12 or 20 views instead of 60.

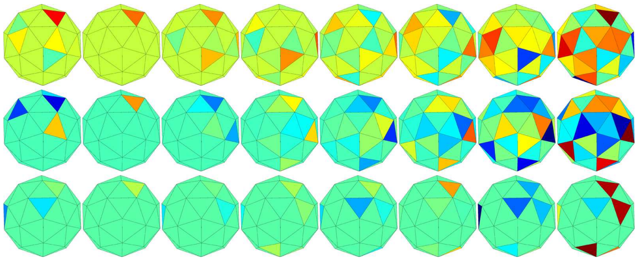

Figure 5: Localized filters and their receptive fields as we stack more layers. First column shows the filter, second the input, and others are results of stacking group convolutions with the same filter. Top row filter has 12 nonzero elements; middle and bottom have 5. The support for the bottom row contains elements of a 12 element subgroup, so its receptive field cannot cover the full input space.

is computationally more efficient, and (2) it allows learning of hierarchically more complex features as layers are stacked. Inspired by this idea, we introduce localized filters for discrete G-CNNs[3]. For a filter $h\colon G \mapsto \mathbb{R}$, we simply choose a subset S of G that is allowed to have nonzero filter values while $h(G-S)$ is set to zero. Since S is a fixed hyperparameter, we can compute (6) more efficiently:

$$f_j^{\ell+1}(y) = \sigma\left(\sum_{i=1}^{c_i}\sum_{g\in S} f_i^{\ell}(yg^{-1})h_{ij}(g)\right). \tag{9}$$

To ensure filter locality, it is desirable that elements of S are close to each other in the manifold of rotations. The 12 smallest rotations in $\mathcal{I}$ are of 72 deg. We therefore choose S to contain the identity and a number of 72 deg rotations.

One caveat of this approach is that we need to make sure S spans G, otherwise the receptive field will not cover the whole input no matter how many layers are stacked, which can happen if S belongs to a subgroup of G (see Figure 5). In practice this is not a challenging condition to satisfy; for our heuristic of choosing only 72 deg rotations we only need to guarantee that at least two are around different axes.

5. Experiments

We evaluate on 3D shape classification, retrieval and scene classification, and include more comparisons and an ablation study in the supplementary material. First, we discuss the architectures, training procedures, and datasets.

Architectures We use a ResNet-18 [15] as the view processing network Φ_1, with weights initialized from ImageNet [9] pre-training. The G-CNN part contains 3 layers with 256 channels and 9 elements on its support (note that the number of parameters is the same as one conventional 3×3 layer). We project from 512 to 256 channels so the number of parameters stay close to the baseline. When the method in Section 4.3 is used to reduce the number of views, the first G-Conv layer is replaced by a H-Corr.

Variations of our method are denoted Ours-X, and Ours-R-X. The R suffix indicate retrieval specific features, that consist of (1) a triplet loss[4] and (2) reordering the retrieval list so that objects classified as the query's predicted class come first. Before reordering, the list is sorted by cosine distance between descriptors. For SHREC'17, choosing the number N of retrieved objects is part of the task – in this case we simply return all objects classified as the query's class.

For fair assessment of our contributions, we implement a variation of MVCNN, denoted MVCNN-M-X for X input views, where the best-performing X is shown. MVCNN-M-X has the same view-processing network, training procedure and dataset as ours; the only difference is that it performs pooling over view descriptors instead of using a G-CNN.

Training We train using SGD with Nesterov momentum as the optimizer. For ModelNet experiments we train for 15 epochs, and 10 for SHREC'17. Following [16], the learning rate linearly increases from 0 to lr in the first epoch, then decays to zero following a cosine quarter-cycle. When training with 60 views, we set the batch size to 6, and lr to 0.0015. This requires around 11 Gb of RAM. When training with 12 or 20 views, we linearly increase both the batch size and lr.

Training our 20-view model on ModelNet40 for one epoch takes $\approx 353s$ on an NVIDIA 1080 Ti, while the corresponding MVCNN-M takes $\approx 308s$. Training RotationNet [20] for one epoch under same conditions takes $\approx 1063s$.

[3] Localization for the continuous case was introduced in [10].

[4] Refer to the supplementary material for details.

Datasets We render 12 × 5, 20 × 3 and 60 × 1 camera configurations (Section 4.1) for ModelNet and the ShapeNet SHREC'17 subset, for both rotated and aligned versions. For the aligned datasets, where equivariance to rotations is not necessary, we fix the camera up-vectors to be in the plane defined by the object center, camera and north pole. This reduces the number of views from 12 × 5 to 12 and from 20 × 3 to 20. For the rotated datasets, all renderings have 60 views and follow the group structure. Note that the rotated datasets are not limited to the discrete group and contain continuous rotations from $\mathbf{SO}(3)$. We observe that the 60 × 1 configuration performs best so those are the numbers shown for "Ours-60". For the experiment with fewer views, we chose 12 from 12×5 and 20 from 20×3 that are converted to log-polar coordinates (Section 4.3). For the scene classification experiment, we sample 12 overlapping views from panoramas. No data augmentation is performed.

5.1. SHREC'17 retrieval challenge

The SHREC'17 large scale 3D shape retrieval challenge [29] utilizes the ShapeNet Core55 [3] dataset and has two modes: "normal" and "perturbed" which correspond to "aligned" and "rotated" as we defined in Section 5.2. The challenge was carried out in 2017 but there has been recent interest on it, especially on the "rotated" mode [6, 10, 21].

Table 1 shows the results. N is the number of retrieved elements, which we choose to be the objects classified as the same class as the query. The Normalized Discounted Cumulative Gain (NDGC) score uses ShapeNet subclasses to measure relevance between retrieved models. Methods are ranked by the mean of micro (instance-based) and macro (class-based) mAP. Several extra retrieval metrics are included in the supplementary material. Only the best performing methods are shown; we refer to [29] for more results.

Our model outperforms the state of the art for both modes even without the triplet loss, which, when included, increase the margins. We consider this our most important result, since it is the largest available 3D shape retrieval benchmark and there are numerous published results on it.

5.2. ModelNet classification and retrieval

We evaluate 3D shape classification and retrieval on variations of ModelNet [39]. In order to compare with most publicly available results, we evaluate on "aligned" ModelNet, and use all available models with the original train/test split (9843 for training, 2468 for test). We also evaluate on the more challenging "rotated" ModelNet40, where each instance is perturbed with a random rotation from $\mathbf{SO}(3)$.

Tables 2 and 3 show the results. We show only the best performing methods and refer to the ModelNet website[5] for complete leaderboard. Classification performance is given by accuracy (acc) and retrieval by the mean average precision (mAP). Averages are over instances. We include class-based averages on the supplementary material.

		micro		macro	
Method	score	mAP	G@N	mAP	G@N
RotatNet [20]	67.8	77.2	86.5	58.3	65.6
ReVGG [29]	61.8	74.0	82.8	49.6	55.9
DLAN [13]	57.0	66.3	76.2	47.7	56.3
MVCNN-M-12	69.1	74.9	83.8	63.2	70.3
Ours-12	70.7	77.7	86.3	63.6	70.8
Ours-20	71.4	*77.9*	*86.8*	64.9	71.9
Ours-60	*71.7*	77.8	86.4	**65.6**	**72.3**
Ours-R-20	**72.2**	**79.1**	**87.5**	*65.4*	**72.3**
DLAN [13]	56.6	65.6	75.4	47.6	56.0
ReVGG [29]	55.7	*69.6*	78.3	41.8	47.9
MVCNN-M-60	57.5	64.1	75.9	50.9	59.7
Ours-12	58.1	66.4	76.7	49.8	58.6
Ours-20	59.3	66.9	77.0	51.7	60.2
Ours-60	*62.1*	*69.6*	*79.6*	*54.6*	*63.0*
Ours-R-60	**63.5**	**71.8**	**81.1**	**55.1**	**63.3**

Table 1: SHREC'17 retrieval results. Top block: aligned dataset; bottom: rotated. We show mean average precision (mAP) and normalized discounted cumulative gain (G). We set the new state of the art by a large margin. Even our 12-view model outperforms the baseline, which shows the potential of keeping equivariance with fewer views.

We outperform the retrieval state of the art for both ModelNet10 and ModelNet40, even without retrieval-specific features. When including such features (triplet loss and reordering by class label), the margin increases significantly.

We focus on retrieval and do not claim state of the art on classification, which is held by RotationNet [20]. While ModelNet retrieval was not attempted by [20], the SHREC'17 retrieval was, and we show significantly better performance on it (Table 1).

5.3. Scene classification

We have shown experiments for object-centric configurations (outside-in), but our method is also applicable to camera-centric configurations (inside-out), which is demonstrated on the Matterport3D [2] scene classification from panoramas task. We sample multiple overlapping azimuthal views from the panorama and apply our model over the cyclic group of 12 rotations, with a filter support of 6. Table 4 shows the results; the full table with accuracies per class and input samples are in the supplementary material.

The MV approach is superior to operating directly on panoramas because (1) it allows higher overall resolution

[5] http://modelnet.cs.princeton.edu

	M40 (aligned)		M10 (aligned)	
	acc	mAP	acc	mAP
MVCNN-12 [33]	90.1	79.5	-	-
SPNet [40]	92.63	85.21	*97.25*	94.20
PVNet [41]	93.2	89.5	-	-
SV2SL [14]	93.40	89.09	94.82	91.43
PANO-ENN [31]	**95.56**	86.34	96.85	93.2
MVCNN-M-12	94.47	89.13	96.33	93.54
Ours-12	94.51	*91.82*	96.33	95.30
Ours-20	*94.69*	91.42	**97.46**	*95.74*
Ours-60	94.36	91.04	96.80	95.25
Ours-R-12	94.67	**93.56**	96.78	**96.18**

Table 2: Aligned ModelNet classification and retrieval. We only compare with published retrieval results. We achieve state of the art retrieval performance even without retrieval-specific model features. This shows that our view aggregation is useful even when global equivariance is not necessary.

	M40 (rotated)	
	acc	mAP
MVCNN-80 [33]	86.0	-
RotationNet [20]	80.0	74.20
Spherical CNN [6]	86.9	-
MVCNN-M-60	90.68	78.18
Ours-12	88.50	79.58
Ours-20	89.98	80.73
Ours-60	*91.00*	*82.61*
Ours-R-60	**91.08**	**88.57**

Table 3: Rotated ModelNet40 classification and retrieval. Note that gap between "Ours" and "MVCNN-M" is much larger than in the aligned dataset, which demonstrates the advantage of our equivariant representation.

while sharing weights across views, and (2) views match the scale of natural images so pre-training is better exploited. Our MVCNN-M outperforms both baselines, and our proposed model outperforms it, which shows that the group structure is also useful in this setting. In this task, our representation is equivariant to azimuthal rotations; a CNN operating directly on the panorama has the same property.

	single [2]	pano [2]	MV-M-12	Ours-12
acc [%]	33.3	41.0	51.9	53.8

Table 4: Scene classification class-based accuracy.

5.4. Discussion

Our model shows state of the art performance on multiple 3D shape retrieval benchmarks. We argue that the retrieval problem is more appropriate to evaluate shape descriptors because it requires a complete rank of similarity between models instead of only a class label.

Our results for aligned datasets show that the full set of 60 views is not necessary and may be even detrimental in this case; but even when equivariance is not required, the principled view aggregation with G-Convs is beneficial, as direct comparison between MVCNN-M and our method show. For rotated datasets, results clearly show that performance increases with the number of views, and that the aggregation with G-Convs brings huge improvements.

Interestingly, our MVCNN-M baseline outperforms many competing approaches. The differences with respect to the original MVCNN [33] are (1) late view-pooling, (2) use of ResNet, (3) improved rendering, and (4) improved learning rate schedule. These significant performance gains were also observed in [34], and attest to the representative potential of multi-view representations.

One limitation is that our feature maps are equivariant to discrete rotations only, and while classification and retrieval performance under continuous rotations is excellent, for tasks such as continuous pose estimation it may not be. Another limitation is that we assume views to follow the group structure, which may be difficult to achieve for real images. Note that this is not a problem for 3D shape analysis, where we can render any arbitrary view.

6. Conclusion

We proposed an approach that leverages the representational power of conventional deep CNNs and exploits the finite nature of the multiple views to design a group convolutional network that performs an exact equivariance in discrete groups, most importantly the icosahedral group. We also introduced localized filters and convolutions on homogeneous spaces in this context. Our method enables joint reasoning over all views as opposed to traditional view-pooling, and is shown to surpass the state of the art by large margins in several 3D shape retrieval benchmarks.

7. Acknowledgments

We are grateful for support through the following grants: NSF-IIP-1439681 (I/UCRC), NSF-IIS-1703319, NSF MRI 1626008, ARL RCTA W911NF-10-2-0016, ONR N00014-17-1-2093, ARL DCIST CRA W911NF-17-2-0181, the DARPA-SRC C-BRIC, and by Honda Research Institute.

References

[1] Michael Artin. *Algebra*, volume 50. Academic Press, 2010. 4

[2] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *CoRR*, 2017. 1, 7, 8, 11, 13, 14

[3] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2, 7

[4] Taco Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *CoRR*, 2018. 3

[5] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999, 2016. 1, 3

[6] Taco S Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *arXiv preprint arXiv:1801.10130*, 2018. 1, 3, 7, 8

[7] Taco S. Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. *CoRR*, 2019. 3

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. *CoRR*, 2017. 1

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 6

[10] Carlos Esteves, Christine Allen-Blanchette, Ameesh Makadia, and Kostas Daniilidis. Learning so (3) equivariant representations with spherical cnns. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–68, 2018. 1, 3, 6, 7

[11] Carlos Esteves, Christine Allen-Blanchette, Xiaowei Zhou, and Kostas Daniilidis. Polar transformer networks. *arXiv preprint arXiv:1709.01889*, 2017. 3, 4, 5

[12] Yifan Feng, Zizhao Zhang, Xibin Zhao, Rongrong Ji, and Yue Gao. Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2018. 2

[13] Takahiko Furuya and Ryutarou Ohbuchi. Deep aggregation of local 3d geometric features for 3d model retrieval. In *BMVC*, pages 121–1, 2016. 7, 12

[14] Zhizhong Han, Mingyang Shang, Zhenbao Liu, Chi-Man Vong, Yu-Shen Liu, Matthias Zwicker, Junwei Han, and CL Philip Chen. Seqviews2seqlabels: Learning 3d global features via aggregating sequential views by rnn with attention. *IEEE Transactions on Image Processing*, 28(2):658–672, 2019. 2, 8

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 6

[16] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. *CoRR*, 2018. 6

[17] Yacov Hel-Or and Patrick C Teo. Canonical decomposition of steerable functions. *Journal of Mathematical Imaging and Vision*, 9(1):83–95, 1998. 3

[18] Joao F Henriques and Andrea Vedaldi. Warped convolutions: Efficient invariance to spatial transformations. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1461–1469. JMLR. org, 2017. 3, 4, 5

[19] Emiel Hoogeboom, Jorn WT Peters, Taco S Cohen, and Max Welling. Hexaconv. *arXiv preprint arXiv:1803.02108*, 2018. 3

[20] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 4, 6, 7, 8, 11, 12

[21] Risi Kondor, Zhen Lin, and Shubhendu Trivedi. Clebsch–gordan nets: a fully fourier space spherical convolutional neural network. In *Advances in Neural Information Processing Systems*, pages 10138–10147, 2018. 7

[22] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. *CoRR*, 2018. 3

[23] Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *The IEEE International Conference on Computer Vision (ICCV) Workshops*, December 2015. 1, 2

[24] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 922–928. IEEE, 2015. 1, 2

[25] Willard Miller. *Symmetry groups and their applications*, volume 50. Academic Press, 1973. 3

[26] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proc. CVPR*, volume 1, page 3, 2017. 1, 2

[27] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 1, 2, 3

[28] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016. 2, 3

[29] Manolis Savva, Fisher Yu, Hao Su, Asako Kanezaki, Takahiko Furuya, Ryutarou Ohbuchi, Zhichao Zhou, Rui Yu, Song Bai, Xiang Bai, Masaki Aono, Atsushi Tatsuma, S. Thermos, A. Axenopoulos, G. Th. Papadopoulos, P. Daras, Xiao Deng, Zhouhui Lian, Bo Li, Henry Johan, Yijuan Lu, and Sanjeev Mk. Shrec'17 track: Large-scale 3d shape retrieval from

shapenet core55. In *10th Eurographics workshop on 3D Object retrieval*, pages 1–11, 2017. 2, 7, 12

[30] Joseph Segman, Jacob Rubinstein, and Yehoshua Y Zeevi. The canonical coordinates method for pattern deformation: Theoretical and computational considerations. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12):1171–1183, 1992. 3, 4

[31] Konstantinos Sfikas, Ioannis Pratikakis, and Theoharis Theoharis. Ensemble of panorama-based convolutional neural networks for 3d model classification and retrieval. *Computers & Graphics*, 71:208–218, 2018. 8

[32] Martin Simonovsky and Nikos Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proc. CVPR*, 2017. 1

[33] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 1, 2, 3, 4, 8, 11, 12, 13

[34] Jong-Chyi Su, Matheus Gadelha, Rui Wang, and Subhransu Maji. A deeper look at 3d shape classifiers. *CoRR*, 2018. 1, 8

[35] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018. 3

[36] Marysia Winkels and Taco S Cohen. 3d g-cnns for pulmonary nodule detection. *arXiv preprint arXiv:1804.04656*, 2018. 3

[37] Daniel Worrall and Gabriel Brostow. Cubenet: Equivariance to 3d rotation and translation. *arXiv preprint arXiv:1804.04458*, 2018. 3

[38] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, volume 2, 2017. 3

[39] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1912–1920, 2015. 1, 7

[40] Mohsen Yavartanoo, Eu Young Kim, and Kyoung Mu Lee. Spnet: Deep 3d object classification and retrieval using stereographic projection. *CoRR*, 2018. 2, 8

[41] Haoxuan You, Yifan Feng, Rongrong Ji, and Yue Gao. Pvnet: A joint convolutional network of point cloud and multi-view for 3d shape recognition. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1310–1318. ACM, 2018. 2, 8