I2I: Image to Icosahedral Projection for $\mathrm{SO}(3)$ Object Reasoning from Single-View Images

David M. Klee
Khoury College of Computer Science
Northeastern University
Boston, MA 02115
klee.d@northeastern.edu

Ondrej Biza
Khoury College of Computer Science
Northeastern University
Boston, MA 02115

Robert Platt
Khoury College of Computer Science
Northeastern University
Boston, MA 02115

Robin Walters
Khoury College of Computer Science
Northeastern University
Boston, MA 02115

Abstract

Reasoning about 3D objects based on 2D images is challenging due to large variations in appearance caused by viewing the object from different orientations. Ideally, our model would be invariant or equivariant to changes in object pose. Unfortunately, this is typically not possible with 2D image input because we do not have an a priori model of how the image would change under out-of-plane object rotations. The only $\mathrm{SO}(3)$-equivariant models that currently exist require point cloud input rather than 2D images. In this paper, we propose a novel model architecture based on icosahedral group convolution that reasons in $\mathrm{SO}(3)$ by projecting the input image onto an icosahedron. As a result of this projection, the model is approximately equivariant to rotation in $\mathrm{SO}(3)$. We apply this model to an object pose estimation task and find that it outperforms reasonable baselines.

1 Introduction

In many applications such as robotic manipulation, autonomous vehicle navigation, scene segmentation, or 3D modeling, it is useful to reason about 3D geometry of objects in the world using sensor data captured from only a single point-of-view. The most widely available such data type which can be collected under the fewest assumptions is an image, either in grey-scale, with 3 RGB channels, or with a depth channel. These applications have thus motivated much work in computer vision dedicated to inferring 3D information such as object orientation, position, and size [1, 2, 3, 4, 5] from images. These methods, however, fail to capture and exploit the symmetry of 3D space. While some computer vision models incorporate 2D symmetry [6], it is very difficult to incorporate 3D symmetries.

Alternatively, $\mathrm{SO}(3)$-equivariant neural networks have emerged as a powerful tool for modeling 3D geometry which directly incorporates 3D symmetries into the model. As a consequence, equivariant neural networks demonstrate improved generalization properties, data efficiency, and provable consistency [7, 8, 9]. In order to apply such symmetry-aware methods to pose estimation, however, the input data must be $\mathrm{SO}(3)$-transformable and thus until now such methods have been restricted to either point clouds [10], spherical images [11], or highly structured multi-view images [12]. Moreover, such methods are computationally expensive and not necessarily robust to occlusions or partial inputs.

Preprint. Under review.

We propose Image2Icosahedral (I2I), a method that allows for $\mathrm{SO}(3)$-equivariant reasoning on embeddings learned only from single-view image inputs. The embeddings and downstream model can be trained in an end-to-end manner. Since our model incorporates 3D symmetries, it is able to generalize better than image-based models over transformations induced by changes in camera viewpoint or object orientation. Moreover, I2I is faster and more memory efficient than other methods as it exploits convolution over the discrete icosahedral group $I_{60} \subset \mathrm{SO}(3)$. Similar to work on $\mathrm{SO}(2)$-equivariance, we find that using a discrete subgroup leads to faster performance and better optimization with minimal loss in accuracy or equivariance [6].

I2I has two novel components which separate the problems of feature extraction and object-level memory, (1) a projector which maps images to spherical dynamic filters and (2) a dynamic filter icosohedral convolution. The spherical filter projector projects 2D images onto 6 vertices of an icosohedron, representing the partially observed object features and the camera viewpoint. We implement this projector using an $\mathrm{E}(2)$-equivariant CNN [6]. Thus the projector does not need to directly reason about how the object looks from other angles. The output of the projection is then used as a dynamic filter which is convolved over the set of vertices of the icosohedron with learned object feature maps. The object feature maps are signals over the entire icosahedron and thus represent, in latent space, the fully observed object in an explicitly $\mathrm{SO}(3)$-transformable manner.

We evaluate our method on two tasks: object pose estimation where we estimate the $\mathrm{SO}(3)$ orientation of a novel in-class object from a single image and object classification where we infer object category from an image. While our proposed method outperforms state of the art baselines in both cases, our performance on the pose estimation task is particularly compelling. We outperform other pose-from-image baselines handily, by an order of magnitude in some cases. Our ablations show that our method is insensitive to minor variations in the algorithm – the general idea of reasoning over the icosahedron seems to be the key reason for the success of our method.

To summarize, our contribution are

- We propose a novel model that implements a discrete approximation to an $\mathrm{SO}(3)$-equivariant dynamic convolution filter over S^2.
- We propose a novel method of projecting a 2D image onto an equivariant spherical convolutional filter.
- When applied to the problem of estimating object pose in $\mathrm{SO}(3)$ from a monocular image, our method outperforms relevant baselines convincingly.
- Our method is faster and less memory intensive than relevant baselines and can easily be integrated into traditional CNN object detection pipelines.

2 Related Work

3D Shape Analysis Research into 3D object recognition has been facilitated by large datasets of common object models [13, 14, 15, 16]. The recognition problem has been studied for several input modalities including multiview images [17, 18, 12], point clouds [19, 20, 10], and projections of 3D geometry [21, 22, 23]. In the challenging setting where objects are not aligned to a canonical orientation, methods that reason about 3D space can achieve better performance. RotationNet [18] classifies objects while implicitly reasoning about the view point of each image, while EMVN [12] captures images from structured view points so that information can be processed in a $\mathrm{SO}(3)$ equivariant manner. In contrast, our method reasons about 3D rotation using a single image input with no restrictions on the camera view point.

6D Pose Estimation Reasoning about the 3D position and orientation of objects in a single image is challenging, yet has important applications in robotics [24] and autonomous vehicles [25]. If 3D CAD models are available, traditional computer vision methods like iterative closest point [26] and template matching [27] can be effective. Pose estimation has also been approached using convolutional networks which are trained to predict the 3D bounding boxes [5], visual keypoints [4, 28] or 3D object coordinates [2, 3, 29], with which the pose can be extracted based on known properties of the objects such as 3D size, appearance, or canonical instance. In contrast, our method does not require any information about object at inference time and it explicitly incorporates 3D rotational symmetry. Our method can be viewed as complementary to the existing 6D object pose estimation literature,

since it can be integrated into existing convolutional pipelines to provide improved reasoning of 3D rotations.

Rotation Equivariance Neural networks with components equivariant to 3D rotations (the $\mathrm{SO}(3)$ abstract group) have found applications in a multitude of domains including quantum chemistry [30, 31, 9, 32], trajectory prediction [33] and, most significant for our work, shape classification and pose estimation for 3D objects [23, 34, 30, 11, 12, 35, 9, 36, 37, 38, 10, 39, 40, 41]. One class of approaches is based on the spherical convolution [23]. A point cloud is projected onto a sphere and processed using fully $\mathrm{SO}(3)$ equivariant spherical convolutions [23, 11, 42]. Spherical projection was also used with single views of objects in the Homeomorphic VAE [34]. Homeomorphic VAE is approximately $\mathrm{SO}(3)$ equivariant, as there is no linear transformation of the pixels of the input image that would correspond to a 3D rotation of the depicted object [1]. Spherical latent spaces with an approximate $\mathrm{SO}(3)$ equivariance have also been used in [37, 41]. Here, each latent state is a single point on a sphere, instead of a signal over the entire sphere. Compared to [34, 37, 41], we also approximate $\mathrm{SO}(3)$ equivariance by projecting a single image of an object onto a hemi-icosohedral (a discretization of a hemisphere). Our model fits hundreds of object instances across several classes, whereas [34, 37, 41] are limited to a cube with multi-colored faces and a single 3D model of the Utah teapot. Other works have used the icosohedral group [43] for pose estimation in the structured multi-view setting [12] and with point cloud data [10, 39]. Alternatively, prior works represented a point cloud as a graph and used harmonic functions to model $\mathrm{SO}(3)$ equivariant interactions between nearby points [30, 35, 9, 38, 40]. These models are fully $\mathrm{SO}(3)$ equivariant, but as a trade-off, they require full point clouds with an approximately uniform distribution of points. Rotational equivariance was also proposed for capsule networks [44] operating on point cloud data [36]. Our focus is on unstructured single-view inputs.

3 Background

Our method provides a mechanism to learn a mapping from features in image space to a 3D rotational embedding. I2I combines layers which are explicitly constrained to be equivariant to 2D and 3D rotations as well as layers with learned equivariance. We introduce the concept of equivariance and how neural networks can be designed with equivariant layers to preserve symmetry.

3.1 Equivariance

The concept of equivariance to group transformations formalizes when a map preserves symmetry. A symmetry group is a set of transformations that preserves some structure. Given symmetry group G that acts on spaces $\mathcal{X}$ and $\mathcal{Y}$ via $\mathcal{T}_g$ and $\mathcal{T}'_g$, respectively, for all $g \in G$, a mapping, $f\colon \mathcal{X} \to \mathcal{Y}$ is equivariant to G if

$$f(\mathcal{T}_g x) = \mathcal{T}'_g f(x). \tag{1}$$

That is, an equivariant mapping commutes with the group transformation. Invariance is a special case of equivariance when $\mathcal{T}'_g$ is the identity mapping (i.e. applying group transformations to the input of an invariant mapping does not affect the output).

3.2 Group convolution over Homogeneous Spaces

The group convolution operation is a linear equivariant mapping which can be equipped with trainable parameters and used to build equivariant neural networks [45]. Convolution[2] is performed by computing the dot product of a signal with a filter that is shifted across a range of shifts. In standard 2D convolution, the shifting corresponds to a translation in pixel space. Group convolution [45] generalizes this idea to arbitrary symmetry groups, with the filter transforming by elements of the group. Let G be a group and $\mathcal{X}$ be a homogeneous space, i.e. a space on which G acts transitively, for example, $\mathcal{X} = G$ or $\mathcal{X} = G/H$ for a subgroup H. We can compute the group convolution between two functions, $f, \psi\colon \mathcal{X} \to \mathbb{R}^k$, as follows:

$$[f \star \psi](g) = \sum_{x \in \mathcal{X}} f(x) \cdot \psi(\mathcal{T}_g x). \tag{2}$$

[1] In contrast, a point cloud can be rotated in 3D by multiplying the individual points with a rotation matrix.

[2] Technically, this is group correlation, but is commonly called convolution in the context of neural networks.

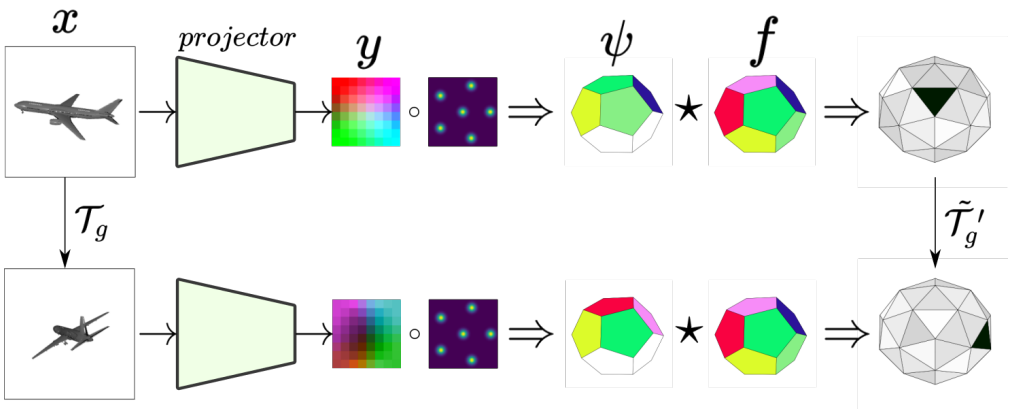

Figure 1: Illustration of our proposed model, Image2Icosahedral (I2I). We project from the input image x onto a feature map y using a C_4-equivariant ResNet model. Then, we project y onto ψ, a dynamic filter on an icosahedral approximation of the sphere S^2. Finally, we convolve ψ with a feature map f using an icosahedral group convolution. The downward pointing arrows indicate that the end-to-end model is approximately equivariant over the icosahedral group $I_{60} \subset \mathrm{SO}(3)$.

Note that the output of the convolution operation is defined for each group element $g \in G$, while the inputs, f and ψ, are defined over a homogeneous space of G. By parameterizing either f or ψ, group convolution may be used as a trainable layer in an equivariant model.

3.3 Icosahedral Group

The continuous transformation group $\mathrm{SO}(3)$ contains all 3D rotations. For reasoning about physical objects, it is desirable to learn functions that are equivariant to $\mathrm{SO}(3)$. However, to make the group convolution operation computationally tractable, discrete subgroups can be used to achieve approximate equivariance to continuous groups. Moreover, discrete subgroups are compatible with common pointwise activations, and there is evidence they may be easier to optimize over [6]. Unlike $\mathrm{SO}(2)$, which contains arbitrarily fine subgroups C_n, for $\mathrm{SO}(3)$, the largest discrete subgroup which is not contained a planar rotation group is the icosahedral group I_{60} which is composed of 60 group elements. The group can be conceptualized as the orientation-preserving symmetries of a regular icosahedron (12 vertices, 20 faces) or its dual polyhedron, the regular dodecahedron (20 vertices, 12 faces).

In our work, we perform group convolution between signals that live on a quotient space of the icosahedral group, i.e. an I_{60}-homogeneous space. Specifically, we use the quotient space $V_{12} = I_{60}/C_5$, which arises from the action of I_{60} on the twelve vertices of the icosahedron. The quotient space V_{12} is a discrete approximation of the 2-sphere, $\mathrm{S}^2 = \mathrm{SO}(3)/\mathrm{SO}(2)$.

4 Method

The Image2Icosahedral method (I2I) is designed to solve tasks which require reasoning about 3D geometry using only 2D images as input. For example, consider the task of mapping a 2D image of an object to its orientation in $\mathrm{SO}(3)$. Methods which take 3D representations such as point clouds as input can solve this task efficiently by exploiting $\mathrm{SO}(3)$ symmetry using equivariant neural networks. We would like to be accomplish something similar when taking 2D images as input also. Unfortunately, this is not possible using existing methods because this would require knowing a priori how the image would transform under 3D rotations of the object. With the exception of $\mathrm{SO}(2)$ rotations in the plane of the camera, we do not know how to compute the new 2D image $\mathcal{T}_g x$ which would result from a rotation g.

Our method, I2I, has two parts: (1) a projector from the image x onto a dynamic convolutional filter over the sphere S^2; and (2) convolution of this filter over S^2. The projector maps the 2D image into a 3D representation which we know how to transform using $\mathrm{SO}(3)$. This enables us to use $\mathrm{SO}(3)$-equivariant convolution over S^2 to then solve the task. The projector uses an $\mathrm{SO}(2)$-

equivariant ResNet to project onto S^2 (the pipeline going from x to y to ψ on the left side of Figure 1). For the convolution, we apply equivariant group convolution over S^2 of the dynamic filter with a class-specific object representation over the sphere (the convolution between ψ and f on the right side of Figure 1).

4.1 $\mathrm{SO}(2)$-Equivariant Projector onto a 2D Feature Map

The projector part of our model maps an image x onto a dynamic filter ψ defined over S^2. Since $\mathrm{SO}(3)$ can transform signals on S^2, this enables $\mathrm{SO}(3)$-equivariance downstream. The projector has two parts. First, a fully convolution model maps the input image x to a 2D feature map y using an $\mathrm{SO}(2)$ equivariant model. Then, we project that map onto S^2. We encode the input image x to a 2D feature map y using a ResNet-style architecture [46]. Since the input images do not have 3D rotational symmetry, we cannot impose $\mathrm{SO}(3)$-equivariance on this part of the model. However, we can enforce symmetry with respect to $\mathrm{SO}(2)$ rotations in the image plane. That is, rotating the object along the roll axis of the camera corresponds to $\mathrm{SO}(2)$ rotations of the pixels in the image plane. We encode this symmetry into the ResNet model by replacing the standard convolution layers with steerable convolutions defined with respect to the discrete cyclic subgroup C_n of rotations by multiples of $2\pi/n$, implemented using the E2CNN framework [6].

4.2 Projection onto the Vertices of the Icosahedron

The learned 2D feature map y is then orthographically projected onto S^2 to give a feature map ψ on S^2 which we will use as a dynamic filter. In practice, it is necessary to discritize S^2 to encode features. Recall that $V_{12} = I_{60}/C_5$ denotes the quotient space which is a discrete approximation of S^2 over the 12 vertices of the icosohedron. We use this set as a discritization of S^2 since it forms a regular mesh and is compatible with the action of the icosohedral group I_{60}. Therefore, the dynamic filter is the map $\psi\colon V_{12} \to \mathbb{R}^k$. In fact, we are able to capture more refined spatial resolution by subdividing the faces of the icosohedral [43] and sampling along this submesh. We still index signals along this submesh using V_{12} by grouping them along the channel dimension at each of vertices in V_{12}.

Similar to 2D convolutional filters, our dynamic filter is localized in that it is 0 except on the side of the sphere facing the camera. The non-occluded vertices are projected onto the image plane of the feature map, such that the identity group element is located in the center. Features for the vertices are generated by applying a Gaussian kernel centered at the vertices' projected pixel location to the feature map. The projection is illustrated in Figure 2. We find this simple projection operation is effective and useful since it preserves spatial information about the location of features in the image.

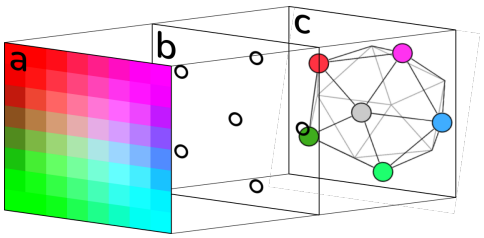

Figure 2: Projection of features from image space to quotient group of I_{60}. (a) dense feature map visualized as 3-channel image; (b) locations of vertices orthographically projected onto image space, local Gaussian kernels are used to sample from feature space; (c) resulting features on homogenous space V_{12}, note that only non-occluded vertices are assigned features during projection.

4.3 Dynamic Filter Icosohedral Convolution

The dynamic filters ψ generated by the projector may now be used to solve the given task using $\mathrm{SO}(3)$-equivariant group convolution. In group convolution, trainable parameters can be used in either the signal or filter. Since the learned features are locally supported on V_{12}, we treat it as a dynamic filter and convolve it over a signal parametrized by trainable weights. We refer to this learnable signal over V_{12} as the feature sphere, since it contains features over the discretized 2-sphere.

After projection, the features are converted into a dynamic filter. That is, the vectors at each vertex are reshaped into matrices so that the learned feature maps $\psi\colon V_{12} \to \mathbb{R}^k$ become $\psi\colon V_{12} \to \mathbb{R}^{m\times n}$ assuming $k = nm$. The convolution operation between the filter and the feature sphere $f : V_{12} \to \mathbb{R}^n$ is defined:

$$[f \star \psi](g) = \sum_{v\in\mathcal{N}} f(\mathcal{T}_g^{-1}v) \cdot \psi(v) \tag{3}$$

where $\mathcal{N}$ is a subset of V_{12} that defines the local neighborhood around the identity element where the filter is non-zero (i.e. all non-occluded vertices during projection). This formulation can be seen to be equivalent to the equation in Eqn. 2 by re-indexing the sum $w = \mathcal{T}_g^{-1}v$, but it is computationally faster since it avoids multiplying by elements v outside the support of the filter. The result of this operation is a vector space over the full icosahedral group, that is $[f \star \psi] : I_{60} \to \mathbb{R}^m$. For the orientation prediction task, normalizing the output using softmax gives a probability distribution over orientations of the input object.

For certain applications (e.g. object classification), it is desirable to learn features that are invariant to the rotation of the object or the image view point. An invariant representation can be achieved by adding a group pooling operation at the end of the model. In group pooling, we take a max (or average) along the dimension of the group, e.g. over all orientations in I_{60}. This is exactly analogous to spatial pooling in a 2D convolutional network.

4.4 Reasoning Over Continuous Orientation

In many applications, for example object pose estimation, we need to infer continuous object pose in $\mathrm{SO}(3)$. This may seem to be a problem for our method because we our output gives values over a discretization of $\mathrm{SO}(3)$, i.e. the icosahedral group. However, we handle this problem by also inferring rotational offsets as part of the feature vector associated with each group element. This is analogous to the idea of anchor boxes, used in object detection, where the network selects from discrete anchor locations then predicts continuous refinements to them [47].

We therefore decompose the $\mathrm{SO}(3)$ regression problem into two parts: first classify the nearest group element of I_{60} and then regress to the rotational offset from the group element. Therefore, we define the output of the icosahedral convolution over $\mathbb{R}^7$, i.e. each element of the icosahedral group is associated with a different vector in $\mathbb{R}^7$. One element of this vector is interpreted as an (unnormalized) probability $p(\hat{g})$ that the true rotation is closest to the associated group element $\hat{g} \in I_{60}$. The other six elements of the vector are used to construct an orthogonal rotation matrix $\hat{R}^{\Delta}_{\hat{g}}$ (i.e. the offset matrix) using the Gram-Schmidt orthonormalization process proposed in [34]. The estimated orientation in $\mathrm{SO}(3)$ is thus $\hat{R}^{\Delta}_{\hat{g}} \cdot \hat{g}$. We use the following loss function, which was introduced in [10]:

$$\mathcal{L}(g, R^{\Delta}_{\hat{g}}, p(\hat{g}), \hat{R}^{\Delta}_{\hat{g}}) = \mathcal{L}_{cls}(\delta_g(\hat{g}), p(\hat{g})) + \lambda\mathcal{L}_2(R^{\Delta}_g, \hat{R}^{\Delta}_g) \tag{4}$$

where g is the closest element in I_{60} to the ground truth orientation with ground truth offset R^{Δ}_g. The first term in Equation 4 is the cross entropy loss between the predicted distribution p and the delta distribution at g. The second term is an L2 loss on the elementwise difference between the two offset rotation matrices. Note that even though an offset is predicted for each element in I_{60}, this loss depends only on the group element closest to the ground truth oreintation. While the rotational offset $\hat{R}^{\Delta}_g$ need not exceed $2\pi/5$ in magnitude (this is the orientation difference between neighboring elements on the icosahedron), we find that enforcing limit this is unnecessary and decreases accuracy.

5 Experiments

5.1 Overview

Architecture: We use a ResNet-18 convolutional architecture [46] for the spherical filter projector. We reduce the downsampling of the network such that it outputs a dense feature map, rather than a single vector. The convolutional layers are instantiated with C_4-symmetric kernels using the $e2cnn$ library [6]. The input and output feature map have features with a trivial representation type whereas the hidden features use a regular representation. The dense feature map is projected onto the icosahedron vertices by applying Gaussian kernels with $\sigma = 0.2$ to their respective locations in

pixel space. We project onto 16 vertices of the icosahedron which is a discrete approximation of a hemisphere (out of 42 vertices total on the icosahedron). This filter is convolved with a dense feature sphere for which the weights of all 42 vertices are learnable. For the pose estimation task, our model is fully convolutional and outputs an $\mathbb{R}^7$ feature vector for each element of the icosahedral group (see Section 4.4). For the shape classification task, we perform a group pooling operation and output a vector of class labels.

Training: We train our method using SGD with Nestorov momentum. The initial learning rate is 1e-3 and decays with steps of 0.1 to 1e-5 during training. We use a batch size of 64 which consumes around 5 GB of RAM for single-class prediction and 8 GB of RAM for multi-class prediction on ModelNet40. Our method is trained for 40 epochs for orientation prediction and 20 epochs for shape classification. A single epoch takes $\approx$ 1.7min for orientation prediction and $\approx$ 50 min for shape classification on NVIDIA 2080Ti graphics card. We augment images during training with random translations of ± 3 pixels which reduces overfitting.

Datasets: We evaluate our method on the ModelNet40 dataset using the original split of 9,843 training objects and 2,468 test objects [13]. The objects were aligned manually per category by [48], and we apply random $\mathrm{SO}(3)$ rotations before rendering. We render 60, 128×128 images of each object in the train and test set. Depth images are rendered with trimesh [49] and grayscale images are rendered with Pyrender [50]. When collecting images for symmetric objects, we only sample object rotations that are orthogonal to the axis of symmetry.

5.2 Inferring Object Orientation

Experiment: Inferring object orientation in $\mathrm{SO}(3)$ for novel in-class objects from a single image is a challenging problem for which a variety of methods exist, including [20, 10, 51]. These methods can be categorized based on the modality of the input data. Some methods take point cloud input [20, 10]. Others take monocular images, e.g. [51]. Table 1 shows a comparison of these methods against our proposed method. In this experiment, we compare the ability of our method to correctly infer object orientation for novel objects from ten object categories in ModelNet40. In all cases, we assume the input data was produced by a single depth camera or a grayscale intensity camera. There are three categories of comparison. The methods in the first and second categories take single (monocular) images, depth images, and grayscale images respectively, as input. In each category, we compare our method (I2I) with the following three baselines. The first baseline (CNN) is a standard CNN backbone which predicts a rotation matrix using a linear layer after a ResNet encoder. The second baseline (CNN+IER) is the EfficientPose method of [51] which performs iterative error refinement by adding residuals to the initial rotation prediction over three additional linear layers. The third baseline (E2CNN-Eq) uses a C_4-equivariant ResNet architecture and then predicts rotation as a combination of a $SO(2)$-equivariant prediction of rotations in the camera plane and an $SO(3)$-equivariant prediction of the rotational component out of the camera plane (a technique first suggested by [34]). The methods in the third category take point clouds as input. These include KPConv [20] which uses a deformable convolutional operation, and EPN [10] which encodes end-to-end equivariance to the icosahedral group. These methods are not directly comparable to I2I because they reason over point clouds rather than images, but we include them here for reference.

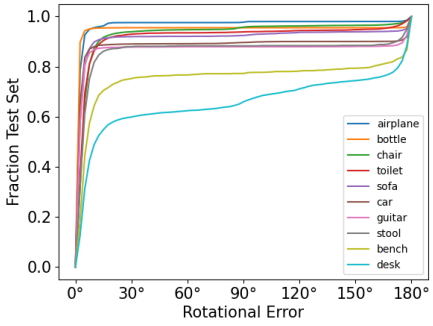

Figure 3: Amount of rotation error in I2I pose predictions for different objects.

Results and Discussion: Table 1 shows the results. Qualitatively, the results from the first two categories of comparison (depth and grayscale) are similar. Our method (I2I) outperforms on both modalities for all ten object categories. It is particularly noteworthy that our proposed method, I2I, outperforms E2CNN-Eq as significantly as it does. Since our method essentially combines a C_4-equivariant ResNet with the icosahedral convolution, E2CNN-Eq shows that our method would perform much worse without the icosahedral piece. Clearly, the icosahedral convolution is important, presumably because it enables us to reason explicitly in $\mathrm{SO}(3)$ while E2CNN-Eq is equivariant

input	method	desks	bottles	sofas	toilets	cars	chairs	stools	airplanes	guitars	benches
Depth Img	CNN	96.3	2.18	29.17	16.19	21.23	11.1	10.79	6.63	34.23	64.36
	CNN+IER [51]	89.89	2.44	14.74	15.37	38.94	10.64	11.03	6.7	31.47	71.62
	E2CNN-Eq [6]	75.8	1.77	7.27	8.5	15.88	7.98	8.45	4.25	31.78	68.71
	I2I (ours)	10.63	0.54	2.12	3.27	1.85	3.60	3.92	1.37	2.18	5.80
Grayscale Img	CNN	92.44	5.77	44.72	27.33	50.39	22.11	19.43	10.64	76.05	84.38
	CNN+IER [51]	92.93	6.76	47.25	30.05	61.53	20.44	17.75	10.56	42.33	87.24
	E2CNN-Eq [6]	87.46	4.17	23.76	23.64	68.54	16.34	16.76	6.62	36.54	94.0
	I2I (ours)	43.62	2.58	3.95	5.75	5.58	6.7	6.95	2.71	6.01	15.26
Partial Point Cloud	KPConv [20]	–	2.0	16.9	–	113.4	14.6	–	1.54	–	–
	EPN [10]	–	15.4	3.01	–	93.2	3.2	–	4.4	–	–

Table 1: Median rotation error for single-class object orientation prediction. Results are separated based on the input modality: depth images, grayscale images, and partial point clouds. Data for some objects are omitted for point cloud baselines due to the unreasonable amount of time it took to run these models on those objects.

only over a subgroup of $\mathrm{SO}(2)$. It could also be that the weight sharing scheme of our icosahedral convolution helps our model generalize better to rotations not seen during training. In some cases, the degree to which we outperform is significant, e.g. cars and guitars, due to significant front/back symmetry in these objects which can confuse models which do not reason in $\mathrm{SO}(3)$. It is also noteworthy that grayscale images are worse than depth images as a modality for pose estimation, presumably due to a lack of texture. We would expect all methods to do better with RGB input.

Error Analysis: In Fig. 3, we plot the distribution of errors by magnitude for our method trained on depth. The errors are tend to cluster at approximately 90 degrees and 180 degrees, reflecting symmetries in the object where it is natural for pose errors to occur. Notice that there are no discernible patterns of errors occurring at discrete pose intervals corresponding to the discretization encoded by the icosahedron (where group elements are separated by rotations of $2\pi/5$). This suggests that even though our method is based on a discretization of $\mathrm{SO}(3)$, our method of learning rotation offsets enables it to infer pose accurately over the continuous $\mathrm{SO}(3)$ group.

5.3 Shape Classification

Experiment: In shape classification, the task is to infer the category of an object given a point cloud or image of its appearance or shape. The task is challenging because the object is presented in an orientation selected uniformly at random. We compare our method with standard baselines in four categories (Table 2). In the first two categories, we classify shape based on a single depth or grayscale image, respectively. Here, we baseline against a standard ResNet backbone (CNN) and a C_4-equivariant ResNet model [6] (both models are configured with similar numbers of parameters as ours). We also compare against some multi-view methods, RotationNet-20 [18] and EMVN-12 [12]. These methods are not directly comparable to ours because I2I takes only a single image as input. Finally, we baseline against PointNet++ [19], KPConv [20], and EPN [10]. These are unfair comparisons because these models take *complete, i.e. unoccluded* point clouds as input.

Results: Table 2 shows our results. The key observation to make is that our method (I2I) outperforms the two baslines (CNN and E2CNN-Inv) for classification from single depth images and single grayscale images (the first two categories in Table 2). The fact that I2I outperforms E2CNN-Inv is particularly significant because it suggests that our method can reason about rotation symmetry beyond the planar rotation symmetry present in the image. Surprisingly, we find that our method, when trained on depth images, can even outperform RotationNet-20 (which has access to multiple views of the same object) and PointNet++ (which has access to a complete point cloud) in terms of mean average precision. These results suggest that our model is able to leverage its ability to reason in $\mathrm{SO}(3)$ to improve its classification of objects presented in novel orientations.

5.4 Comparison of Model Variations

In Table 3, we compare multiple variations on our model in the context of the object pose estimation task from Section 5.2 for the ten ModelNet40 object categories described there. We test the following

Input	Method	Acc.	mAP
Single Depth	CNN	76.5	65.5
	E2CNN-Inv [6]	80.4	70.7
	I2I (ours)	81.5	74.5
Single Gray	CNN	70.0	57.8
	E2CNN-Inv [6]	75.1	64.3
	I2I (ours)	76.4	67.8
Multi-View	RotationNet-20 [18]	80.0	74.2
	EMVN-12 [12]	88.5	79.6
Point Cloud	PointNet++ [19]	85.0	70.3
	KPConv [20]	86.7	77.5
	EPN [10]	88.3	79.7

Table 2: ModelNet40 shape classification results. The methods are separated by input modality: single depth and grayscale means the input is a single image rendered from randomly selected view point; multi-view take multiple grayscale images from structured view points; point cloud means full point cloud generated over entire mesh; . We report percent accuracy (Acc.) and mean average precision (mAP), including standard deviation over three random seeds for our method.

method	desk	bottle	sofa	toilet	car	chair	stool	airplane	guitar	bench
I2I (ours)	10.63	0.54	2.12	3.27	1.85	3.60	3.92	1.37	2.18	5.80
Sparse Filter	12.15	0.51	2.04	3.39	1.91	3.7	3.73	1.41	2.34	5.62
Vector Filter	10.46	0.52	2.18	3.42	1.83	3.58	3.64	1.33	2.29	6.06
Ico Filter	11.13	0.53	2.08	3.02	2.09	3.32	3.52	1.36	2.57	7.57

Table 3: Median orientation accuracy for different image to icosahedral projection schemes.

variations. For *Sparse Filter*, only a subset of the weights in the icosahedral convolution are trainable. Here, the icosahedral filter uses only 6 vertices instead of the full 16 and the icosahedral feature sphere uses only 12 vertices rather than the full 42. For *Vector Filter*, we reverse the roles of the dynamic filter and the feature sphere in the icosahedral convolution. Here, the $\mathrm{SO}(2)$-equivariant projector feeds into the feature sphere and the convolution filter has non-dynamic learnable weights (the dynamic filter is vector and feature sphere is matrix). For *Ico Filter*, the ResNet outputs a single 1×1 feature map that is convolved with the feature sphere. Essentially, the 1×1 feature map is assigned to the identity element of the dynamic filter and thus the feature sphere is supported over the full group. We balance the number of parameters in all methods to achieve similar computational cost. Overall, we find that on average our method does slightly better on the orientation prediction task than the variations considered above. However, all four variations have roughly similar performance, suggesting that it is the use of the icosahedral convolution in the first place which is the key element of our method.

6 Conclusion

We present a novel method, Image2Icosahedral, for learning 3D representations of objects from single-view 2D images. This 3D representation allows us to apply $\mathrm{SO}(3)$-equivariant techniques to tasks such as orientation inference and shape classification even for image inputs. Our method outperforms baselines in these tasks using less memory and compute time. Our model is limited by the task to relying on learned symmetry in the spherical filter projector as the group action is unknown in image space. Since we give the ground truth orientation of objects as a point estimate and our model can only output a distribution over I_{60} and not $\mathrm{SO}(3)$, we cannot perfectly describe ambiguities in orientation for objects with symmetry such as a bottle. In future work, we plan to address this by explicitly returning a distribution over $\mathrm{SO}(3)$, effectively giving our model the ability to estimate orientation and implicitly discover symmetry in objects. We will also integrate our method with bounding box methods such as Mask R-CNN [52] to create a single method capable of estimating the 6D pose of multiple objects in a cluttered scene.

References

[1] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European conference on computer vision*, pages 405–421. Springer, 2020.

[2] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2642–2651, 2019.

[3] Zhigang Li, Gu Wang, and Xiangyang Ji. Cdpn: Coordinates-based disentangled pose network for real-time rgb-based 6-dof object pose estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7678–7687, 2019.

[4] Yisheng He, Wei Sun, Haibin Huang, Jianran Liu, Haoqiang Fan, and Jian Sun. Pvn3d: A deep point-wise 3d keypoints voting network for 6dof pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11632–11641, 2020.

[5] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[6] Maurice Weiler and Gabriele Cesa. General E(2)-Equivariant Steerable CNNs. In *Conference on Neural Information Processing Systems (NeurIPS)*, 2019.

[7] Bryn Elesedy and Sheheryar Zaidi. Provably strict generalisation benefit for equivariant models. In *International Conference on Machine Learning*, pages 2959–2969. PMLR, 2021.

[8] Simon Batzner, Albert Musaelian, Lixin Sun, Mario Geiger, Jonathan P Mailoa, Mordechai Kornbluth, Nicola Molinari, Tess E Smidt, and Boris Kozinsky. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):1–11, 2022.

[9] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020.

[10] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3d point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14514–14523, 2021.

[11] 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.

[12] Carlos Esteves, Yinshuang Xu, Christine Allen-Blanchette, and Kostas Daniilidis. Equivariant multi-view networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1568–1577, 2019.

[13] 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.

[14] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[15] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019*, pages 1588–1597. IEEE, 2019.

[16] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Stephen J. Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *Int. J. Comput. Vis.*, 129(12):3313–3337, 2021.

[17] 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.

[18] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019, 2018.

[19] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[20] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6411–6420, 2019.

[21] Song Bai, Xiang Bai, Zhichao Zhou, Zhaoxiang Zhang, and Longin Jan Latecki. Gift: A real-time and scalable 3d shape search engine. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5023–5032, 2016.

[22] Mohsen Yavartanoo, Eu Young Kim, and Kyoung Mu Lee. Spnet: Deep 3d object classification and retrieval using stereographic projection. In *Asian conference on computer vision*, pages 691–706. Springer, 2018.

[23] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[24] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stan Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. *arXiv preprint arXiv:1809.10790*, 2018.

[25] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012.

[26] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, pages 586–606. Spie, 1992.

[27] Stefan Hinterstoisser, Stefan Holzer, Cedric Cagniart, Slobodan Ilic, Kurt Konolige, Nassir Navab, and Vincent Lepetit. Multimodal templates for real-time detection of texture-less objects in heavily cluttered scenes. In *2011 international conference on computer vision*, pages 858–865. IEEE, 2011.

[28] Lucas Manuelli, Wei Gao, Peter Florence, and Russ Tedrake. kpam: Keypoint affordances for category-level robotic manipulation. *arXiv preprint arXiv:1903.06684*, 2019.

[29] Sergey Zakharov, Ivan Shugurov, and Slobodan Ilic. Dpod: 6d pose object detector and refiner. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1941–1950, 2019.

[30] Nathaniel Thomas, Tess E. 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. *CoRR*, abs/1802.08219, 2018.

[31] Brandon M. Anderson, Truong-Son Hy, and Risi Kondor. Cormorant: Covariant molecular neural networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 14510–14519, 2019.

[32] Johannes Brandstetter, Rob Hesselink, Elise van der Pol, Erik Bekkers, and Max Welling. Geometric and physical quantities improve E(3) equivariant message passing. *CoRR*, abs/2110.02905, 2021.

[33] Robin Walters, Jinxi Li, and Rose Yu. Trajectory prediction using equivariant continuous convolution. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[34] Luca Falorsi, Pim de Haan, Tim R Davidson, Nicola De Cao, Maurice Weiler, Patrick Forré, and Taco S Cohen. Explorations in homeomorphic variational auto-encoding. *arXiv preprint arXiv:1807.04689*, 2018.

[35] Adrien Poulenard, Marie-Julie Rakotosaona, Yann Ponty, and Maks Ovsjanikov. Effective rotation-invariant point CNN with spherical harmonics kernels. In *2019 International Conference on 3D Vision, 3DV 2019, Québec City, QC, Canada, September 16-19, 2019*, pages 47–56. IEEE, 2019.

[36] Yongheng Zhao, Tolga Birdal, Jan Eric Lenssen, Emanuele Menegatti, Leonidas J. Guibas, and Federico Tombari. Quaternion equivariant capsule networks for 3d point clouds. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part I*, volume 12346 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2020.

[37] Robin Quessard, Thomas D. Barrett, and William R. Clements. Learning disentangled representations and group structure of dynamical environments. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[38] Adrien Poulenard and Leonidas J. Guibas. A functional approach to rotation equivariant non-linearities for tensor field networks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pages 13174–13183. Computer Vision Foundation / IEEE, 2021.

[39] Xiaolong Li, Yijia Weng, Li Yi, Leonidas Guibas, A Lynn Abbott, Shuran Song, and He Wang. Leveraging se(3) equivariance for self-supervised category-level object pose estimation. *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021.

[40] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021.

[41] Jung Yeon Park, Ondrej Biza, Linfeng Zhao, Jan-Willem van de Meent, and Robin Walters. Learning symmetric embeddings for equivariant world models. *CoRR*, abs/2204.11371, 2022.

[42] Carlos Esteves, Avneesh Sud, Zhengyi Luo, Kostas Daniilidis, and Ameesh Makadia. Cross-domain 3d equivariant image embeddings. In *International Conference on Machine Learning*, pages 1812–1822. PMLR, 2019.

[43] Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International Conference on Machine Learning*, pages 1321–1330. PMLR, 2019.

[44] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 3856–3866, 2017.

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

[46] 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.

[47] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[48] Nima Sedaghat, Mohammadreza Zolfaghari, Ehsan Amiri, and Thomas Brox. Orientation-boosted voxel nets for 3d object recognition. *arXiv preprint arXiv:1604.03351*, 2016.

[49] Dawson-Haggerty et al. trimesh.

[50] Matthew Matl et al.

[51] Yannick Bukschat and Marcus Vetter. Efficientpose: An efficient, accurate and scalable end-to-end 6d multi object pose estimation approach. *arXiv preprint arXiv:2011.04307*, 2020.

[52] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

Appendix

A Additional Implementation Details

I2I I2I uses a ResNet-18 encoder with four layers of two blocks. The base width is 64, which doubles every layer such that there are 512 dimensions after the fourth layer. The striding is set to 1 in the final two layers, resulting in an 8-by-8 feature map. This feature map is processed with a 1×1 convolution to project to the necessary dimensionality for the dynamic filter. All convolutional layers are instantiated with C_4-equivariant kernels. The projection to V_{12} is performed with Gaussian kernels ($\sigma = 0.2$) such that the icosahedron's diameter covers 90% of the feature map. For training on object orientation prediction using Eqn. 4, we set λ to 100. For training on shape classification, we perform group-pooling on the output of the icosahedral group convolution to generate a probability distribution over the classes.

CNN The CNN baseline uses the same ResNet-18 backbone as I2I, except the layers perform traditional (non-equivariant) 2d convolutions and the final feature map is average pooled to generate a feature vector. This feature vector is processed with a linear layer to produce the desired output

CNN+IER The CNN+IER adds three linear layers to CNN, that produce residuals on the rotation prediction. That is, each layer produces a delta rotation that is added to the prediction so it has 3 chances to update the prediction.

E2CNN-Inv The baselines E2CNN-Inv uses the same C_4 equivariant ResNet-18 architecture as I2I, but performs group-pooling to generate a C_4-invariant feature vector. This vector is processed with a linear layer to produce a probability distribution over classes.

E2CNN-Eq The baselines E2CNN-Eq uses the same C_4 equivariant ResNet-18 architecture as I2I. After the encoder, the representation is processed separately by: equivariant layers that predict rotations in the image plane; and invariant layers that predict arbitrary rotations. The output of the method is the product of the in-plane and out-of-plane rotation matrices. This idea of separating the prediction into equivariant and invariant components was initially suggested in [34] where the equivariance was enforced with regularization.

B Measuring Equivariance Error

Our method is designed to generate an embedding that exhibits $SO(3)$-equivariance, that is, if the view point of an image is rotated, then we expect the output signal over I_{60} to transform similarly. To test the equivariance error present in our model, we measure deviations from the equivariance property of Eqn. 1 (for $T_g x$, we have access to object models and rotate them by g before rendering). We separately report errors for group actions, g, that act in the image plane (In-plane) and for arbitrary 3D rotations (Out-of-plane). The results are reported in Table 4 as the median rotation difference between the RHS and LHS of Eqn. 1 using 200 datapoints generated with the ModelNet40 airplane test set.

We find that I2I exhibits the least equivariance error when compared to methods that do not encode equivariance (CNN) or encode $\mathrm{SO}(2)$-equivariance only (E2CNN-Eq). We also include a comparison to I2I using a non-equivariant projector (I2I w/o E2CNN), and find that this method has lower equivariance error than E2CNN. Surprisingly, this is true even for in-plane rotations. This can be explained by the fact that the icosahedron group convolution is equivariant to C_5 in the image plane, and thus part I2I w/o E2CNN has equivariance to in-plane rotations which may help it to learn end-to-end equivariance to in-plane rotations. Note the equivariance error to in-plane rotations for E2CNN-Eq is not 0 due to discretization of the rotation group; we only enforce equivariance to C_4 subgroup.

Method	Median Equivariance Error	
	In-plane	Out-of-plane
CNN	13.4	19.9
E2CNN-Eq	7.2	11.4
I2I w/o E2CNN	4.4	7.0
I2I	3.2	5.8

Table 4: Comparison of equivariance errors, reported as median rotational differences in degrees. Grayscale images are generated using objects from ModelNet40 airplane test set. In-plane means that the objects were rotated by random $\mathrm{SO}(2)$ rotations in the image plane; out-of-plane means that the objects were rotated by random $\mathrm{SO}(3)$ rotations.

C Additional Shape Classification Results

To better understand the error modes of our method, we report the confusion matrix when trained on depth images in Fig. 4. As expected, similar objects, like flower-pots and plants or cups and vases, are common mis-classifications. We also observe errors between seemingly distinct categories such as radios and dressers; however, these objects can have similar appearance when viewed in a single image especially without any cue to their relative size.

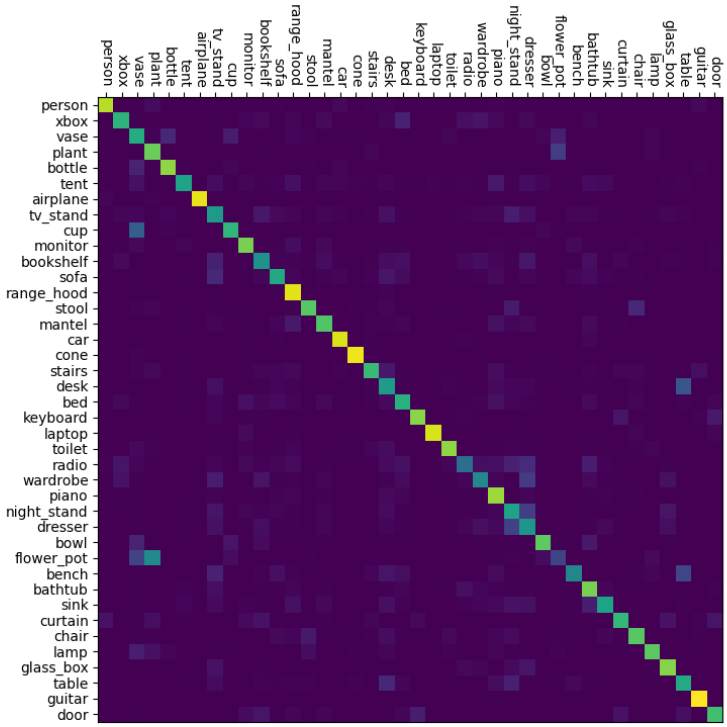

Figure 4: Confusion matrix of I2I on ModelNet40 shape classification task from Table 2. The ground truth labels are on the y-axis and the predicted labels are on the x-axis (i.e. the first row shows the probability distribution over classes for an input image of a person).

D Visualizing Object Orientation Predictions

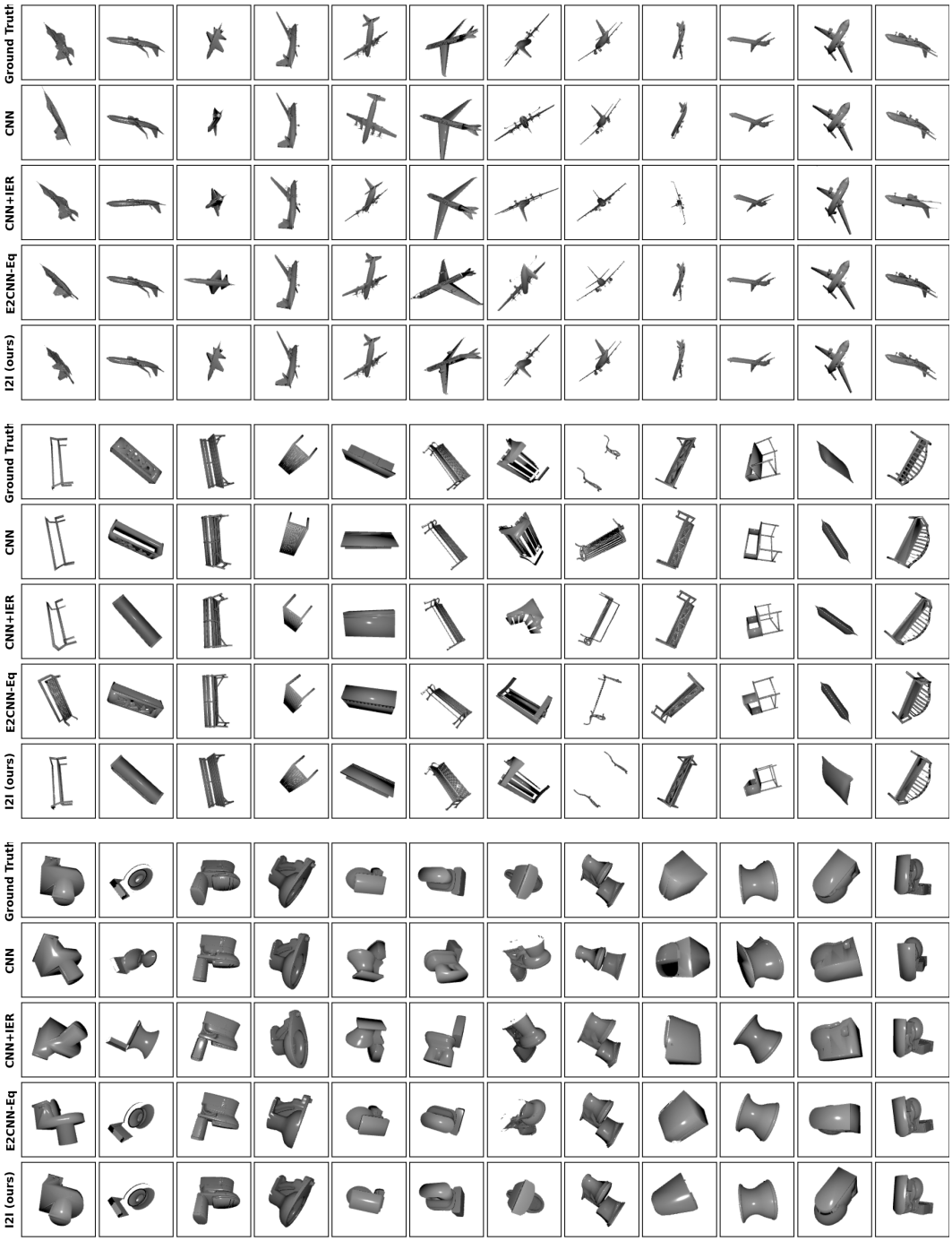

Figure 5: Example predictions for object orientation prediction task from Table 1 (Grayscale Img). To illustrate the predictions, we render new images of the respective objects using the $\mathrm{SO}(3)$ rotation predicted by each method. While I2I generally produces accurate predictions, large errors can be seen in columns 2 and 5 of the bench images.

E Object Orientation Prediction from RGB Images

In this section, we evaluate our method using single RGB images of six objects from the ShapeNet55 dataset [14]. For each object class, we perform an 80-20 train-test split; for chair and airplane, we cap the number of objects used to 1,000. We render images using Blender (open sourced under GPL), by applying 60 random $\mathrm{SO}(3)$ rotations to each object in the train and test sets. We compare our method to the same baselines introduced in Section 5.

method	guitar	bed	bottle	bowl	clock	chair	file-cabinet	airplane
CNN	12.63	70.38	11.64	12.87	22.11	41.31	42.40	17.77
CNN+IER	12.55	83.68	10.38	12.01	26.76	39.40	41.40	17.78
E2CNN-Eq	10.21	63.47	8.58	10.30	17.89	38.09	41.92	14.57
I2I	5.81	46.23	4.22	4.85	9.78	26.03	22.13	8.95

Table 5: Median accuracy (°) on object orientation prediction task. Object classes were selected from ShapeNet55 dataset and rendered as RGB images from random $\mathrm{SO}(3)$ views.

The results in Table 5 show that our method consistently outperforms the baseline methods. I2I achieves a median accuracy less than 10° for five out of eight objects. The accuracy of our method on ShapeNet55 objects support our claims from Section 5 that the icosahedral group convolution improves reasoning about 3D rotations of objects, despite the 2D nature of the input images. In addition, these results demonstrate that our method is flexible enough to accommodate different input modalities including depth, gray-scale, and RGB images.

F Effects of Object Centering

We test whether our method is shift-invariant, a necessary property in order to integrate it into multi-object 6D pose prediction pipelines. In such a pipeline, such as PoseCNN, the first step is to extract bounding boxes of relevant objects in the scene, which may have some positional noise. Thus, we test how robust I2I's orientation predictions are to object centering in the image by inserting random shifts at test time. In Table 6, we show results when I2I is trained on depth images of airplanes with different amounts of pixel shifts during both training and evaluation. While highest accuracy is achieved with a centered object, we find that I2I's accuracy remains high even up to pixel shifts up to 30 pixels (for input images that are 128-by-128 pixels). We hypothesis the robustness is due to the shift-invariance of the ResNet projector, which has a downsampling factor of 16. These results suggest that our method could be trained end-to-end in conjunction with a bounding box prediction method such as Mask R-CNN.

Pixel Shift	Median Error (°)
±5px	1.44
±10px	1.47
±15px	1.56
±20px	1.64
±25px	1.77
±30px	1.88

Table 6: Orientation prediction of I2I on ModelNet40 airplane class (depth images) under varying levels of pixel shifts. The pixel shifts are randomly sampled in the labeled ranges and applied to the input images during training and evaluation.