CENTER FOR
Brains
Minds+
Machines

CBMM Memo No. 075 **December 31, 2017**

3D Object-Oriented Learning: An End-to-end Transformation-Disentangled 3D Representation

by

Qianli Liao and Tomaso Poggio
Center for Brains, Minds, and Machines, McGovern Institute for Brain Research,
Massachusetts Institute of Technology, Cambridge, MA, 02139.

Abstract:

We provide more detailed explanation of the ideas behind a recent paper on "Object-Oriented Deep Learning" [1] and extend it to handle 3D inputs/outputs. Similar to [1], every layer of the system takes in a list of "objects/symbols", processes it and outputs another list of objects/symbols. In this report, the properties of the objects/symbols are extended to contain 3D information — including 3D orientations (i.e., rotation quaternion or yaw, pitch and roll) and one extra coordinate dimension (z-axis or depth). The resultant model is a novel end-to-end interpretable 3D representation that systematically factors out common 3D transformations such as translation and 3D rotation. As first proposed by [1] and discussed in more detail in [2], it offers a "symbolic disentanglement" solution to the problem of transformation invariance/equivariance. To demonstrate the effectiveness of the model, we show that it can achieve perfect performance on the task of 3D invariant recognition by training on one rotation of a 3D object and test it on 3D rotations (i.e., at arbitrary angles of yaw, pitch and roll). Furthermore, in a more realistic case where depth information is not given (similar to viewpoint invariant object recognition from 2D vision) our model generalizes reasonably well to novel viewpoints while ConvNets fail to generalize.

This work was supported by the Center for Brains, Minds and Machines (CBMM), funded by NSF STC award CCF - 1231216.

Contents

1 Introduction

A new framework of "Object-Oriented Learning" has been proposed in [1] to deal with geometric transformations. It offers a new solution to the problem of invariance, equivariance and disentanglement of transformations (See [2] for further discussion on these concepts).

The main ideas in [1] were illustrated with intuitive figures. Experiments on CIFAR-10 supported the argument that the model disentangles in-plane rotations and generalizes to all rotations after training only on one rotation. In this paper, we are going to further explain some ideas behind [1] in a more principled fashion and show exactly how it disentangles transformations.

Furthermore, with the more general and principled view of this approach, we extend the model to 3D, by disentangling 3D rotations. The resultant model is **exactly equivariant and disentangled** (See [2] for definition) throughout all layers with respect to 3D translations and rotations of inputs. We assume here and in the following that the inputs to the first layer are objects with explicitely given poses given.

2 Hierarchical Transformation of Coordinate Frames

In this section, we give a more detailed and principled description of the Object-Oriented Voting depicted in the Figure 2 of [1] (For reference, it is shown again in Figure 1 of this report). This approach can be generally called **"Hierarchical Transformation of Coordinate Frames"** or equivalently **"Hierarchical Composition of Poses"**.

As described before, an important use of the idea of Object-Oriented Learning is to disentangle the transformations of the object and package them as fields/properties of the object.

The key insight of our system is that higher layer objects' poses are children of lower layer poses. When lower layer objects are transformed, higher layer objects are transformed in an exactly equivariant way. The details are as follows:

First, we define a pose/transformation and a rule of how to compose poses.

Definition 1: Pose A pose is a coordinate frame (or frame for short). It can be defined as an arbitrarily arranged coordinate frame A_o or the transformation required to transform the canonical world frame (world origin + standard basis) into this frame. Unless stated otherwise, we usually refer to absolute pose (w.r.t. world frame) when using the term pose.

Definition 2: Relative Pose A relative pose is again a coordinate frame but with respect to some frame of reference, instead of the world's frame.

To avoid ambiguity, whenever talking about a relative pose/frame, one needs to specify the coordinate frame that is its frame of reference. The only exception is the world coordinate frame, since we do not care about anything else "outside the world". Everything can be understood with respect to world frame.

World Coordinate Frame: It is the root frame of the hierarchy of poses. Depending on specific applications, can define anything to be the world frame. In object recognition, the camera/viewer's frame is used as the world frame.

Degrees of Freedom: A pose can be a frame that is transformed by any homogeneous transformation from the world frame (or frame of reference, if relative). This will in principle support a wide range of transformations including translation, rotation, scaling, shearing, stretch, projective transformation, etc.

In current implementations, we have not yet tested many degrees of freedom — one object o only contains the following transformation parameters: the reference point coordinates x,y (and z if in 3D), orientation (in-plane rotation angle in 2D or quaternion if in 3D) and scaling factor. This constitutes a subset of affine transformations.

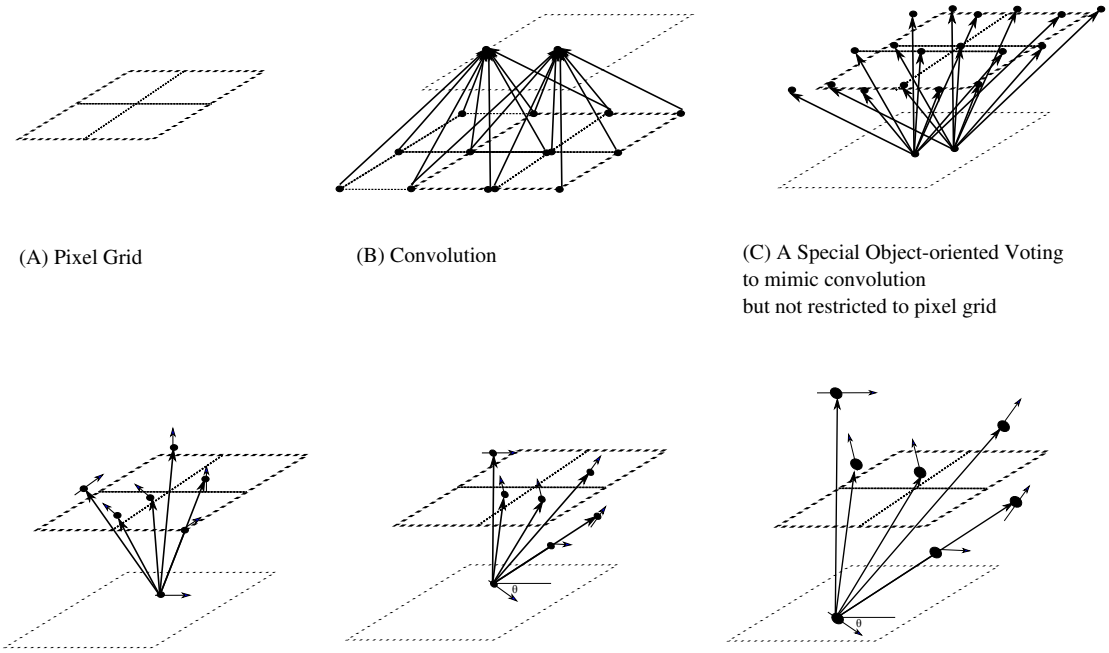

Figure 1: Figure 2 of [1]. We use our formulation of *Hierarchical Transformation of Coordinate Frames'*. The idea is that the pose (coordinate frame) of the input object affects the pose (coordinate frame) of predicted object. As an intuitive illustration, in (F) when the input object is rotated by θ degrees and scaled by $2x$, the same predicted points in the object's local coordinate frame are eventually mapped to points that are more spread out and rotated by θ degrees in the world coordinate frame, because the object's coordinate frame (i.e., transformation) is applied to all of its children (i.e., all downstream objects predicted by it).

In future work, we will test additional degrees of freedom.

Hierarchical Composition of Poses: An object o can predict/vote for the existence of another object p and its relative pose R_p with respect to object o's own coordinate frame. The predicted R_p is invariant to object o's absolute pose A_o.

The absolute pose of object p A_p can be computed by composing A_o and R_p.

If A_o and R_p are represented by transformation matrices, then:

$$A_p = A_o R_p \tag{1}$$

Note that it is not always required (and efficient) to represent A_o and R_p as transformation matrices. It depends on how many degrees of freedoms in pose an object is allowed. For example, in the 2D case, if we only have translation X_p, Y_p, one rotation angle rot_p and a scale $scal_p$. Let Abs and Rel denotes absolute and relative poses, respectively.

We have: $A_p = \{AbsX_p, AbsY_p, AbsRot_p, AbsScal_p\}$, $R_p = \{RelX_p, RelY_p, RelRot_p, RelScal_p\}$. One can calculate A_p by:

$$AbsX_p = AbsX_o + RelX_p \tag{2}$$
$$AbsY_p = AbsY_o + RelY_p \tag{3}$$
$$AbsRot_p = AbsRot_o + RelRot_p \tag{4}$$
$$AbsScal_p = AbsScal_o * RelScal_p \tag{5}$$

In 3D, we currently predict 3 Euler angles to represent a relative rotation of the predicted object from the predicting object. The Euler angles are then converted into quaternion and composed with the rotation of the predicting object to get the absolute rotation.

Once A_p is computed from A_o, the prediction of object p from object o is complete. This is all a voting layer does.

In practice, one object o does not only predict one object p, but rather a number of them. In current implementation, one object predicts a 8 object on a circle (as in 2D) or 26 objects on a sphere (3D) around it (the number 8 and 26 are flexible), in addition to an object in the center of the circle/sphere. Optionally, we also tried adding a trainable deformation/offset to each predicted location, which results in similar performance to the vanilla case.

Proposition 1: Commutative Property Assume the input x are a list of objects in 3D (instead of pixels), the voting layer $V()$ commutes with the transformation $T()$. That is

$$T(V(x)) = V(T(x)) \tag{6}$$

Note that T(x) means that the same transformation is applied to every element of the list of objects x.

2.1 Binding Layer

We used the same binding algorithm described in [2]. There is again a commutative property as follows: **Proposition 2: Commutative Property of Binding Layer** Assume the input x are a list of objects in 3D (instead of pixels), the binding layer $B()$ commutes with the transformation $T()$. That is

$$T(B(x)) = B(T(x)) \tag{7}$$

Note that T(x) means that the same transformation is applied to every element of the list of objects x.

Proposition 1 and 2 jointly guarantees the exact equivariance and disentanglement property described in [2].

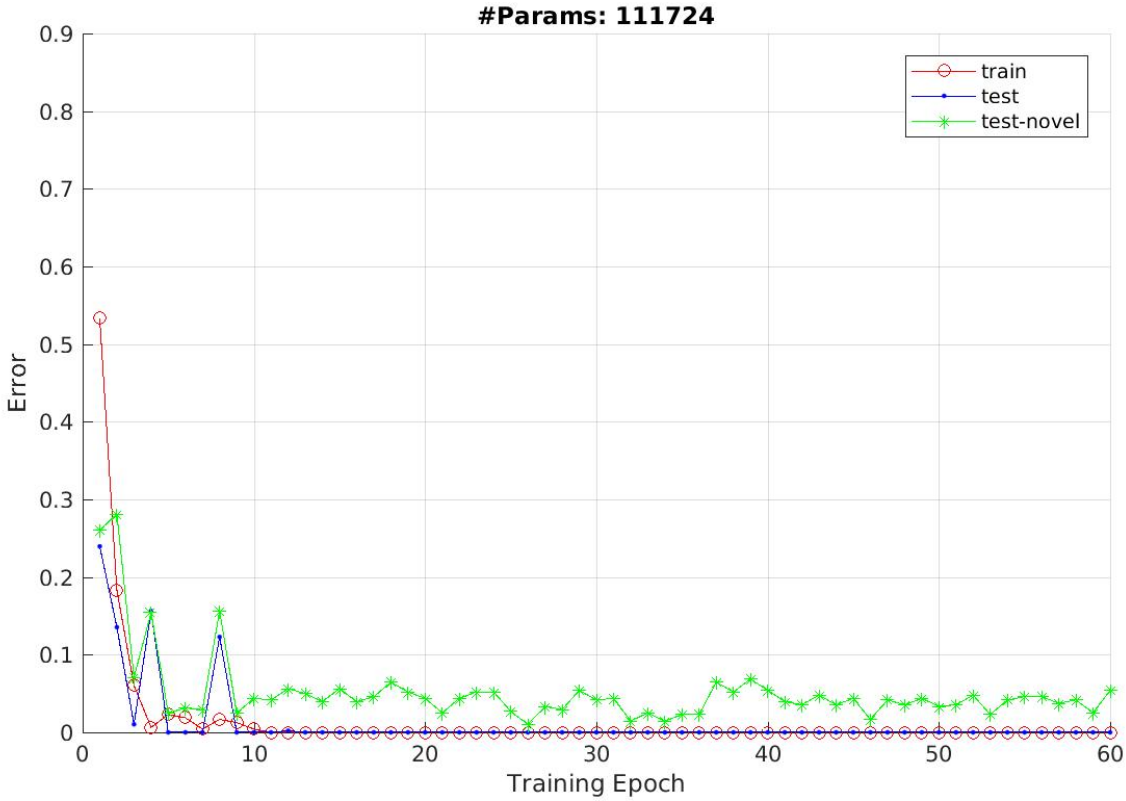

Figure 2: Performance of our model on recognizing 3D patterns. Training and testing are performed on one fixed rotation/view of the objects. "Test-novel" refers to testing the model on novel 3D rotations of the object. The inputs to the model are 3D objects (i.e., signature, x,y,z, pose).

3 Experiments

3.1 Invariant Recognition of 3D Objects from 3D Inputs

We generate synthetic 3D patterns (like point clouds) using a procedure similar to generating the 2D stimuli in [2]. There are 10 different arrangements 7 objects in 3D, representing 10 classes. Training and testing are performed on a fixed viewpoint of the data. We also tested novel viewpoints by randomly rotating the 3D pattern. The results are shown in Figure 2. The model generalizes to all novel rotations that it did not see during training.

3.2 Invariant Recognition of 3D Objects from 2D Inputs

Similar to the above experiment in Figure 2. As our final goal is 2D vision based object recognition, we tried a more natural setting where the all depth information is discarded (by setting the z coordinate to 0, corresponding to orthographic projection). To make the problem slightly more realistic, we provide two different but fixed viewpoints for training. The results are shown in 3. Since there is no z dimension in the inputs, we can also test ConvNets on this task, as shown in 4.

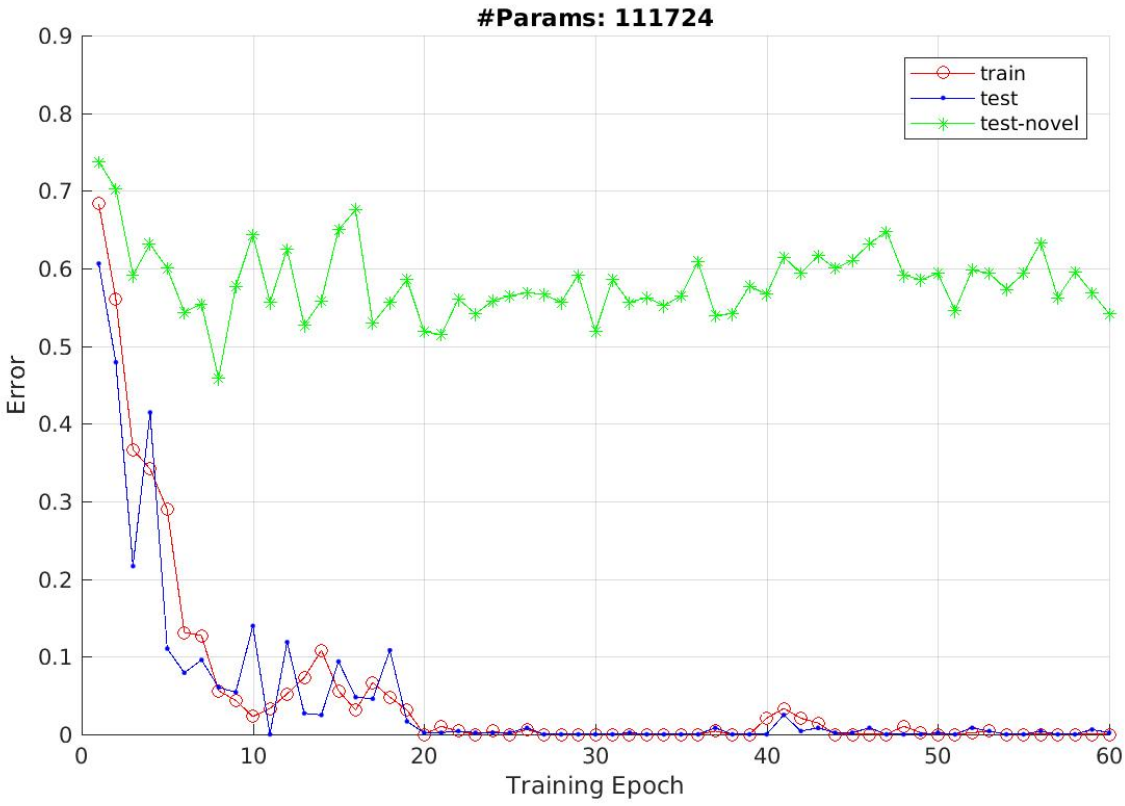

Figure 3: Performance of our model in the recognition of 3D patterns without depth information. Training and testing are performed on two different but fixed rotations/views of the objects. "Test-novel" refers to testing the model on novel 3D rotations of the object. The inputs to the model are 3D objects with depth information set to 0 (i.e., signature, x,y and pose. z=0). To compare with ConvNet, the coordinates x,y are rounded to integers. Even without depth information, our model shows a reasonable performance when generalizing to new viewpoints of the object.

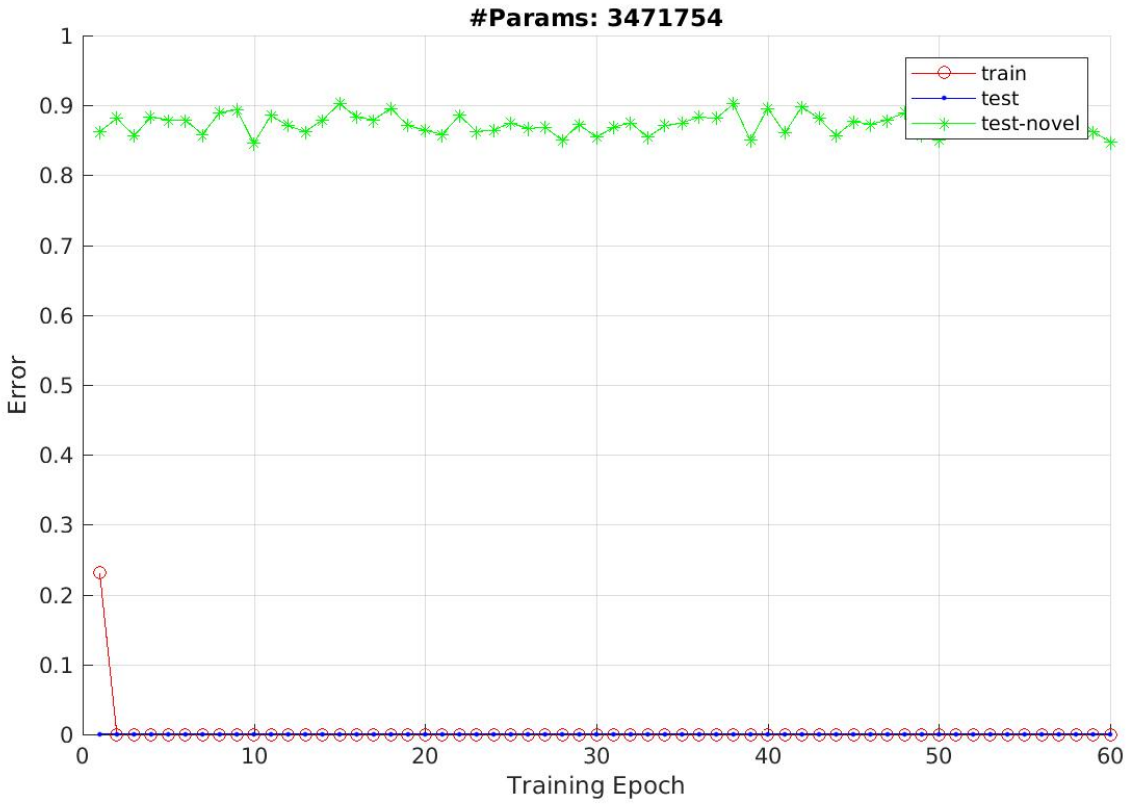

Figure 4: Performance of **ConvNet** (a simple ResNet) on recognizing 3D patterns without depth information. Training and testing are performed on two different but fixed rotations/views of the objects. "Test-novel" refers to testing the model on novel 3D rotations of the object. The inputs to the ConvNet are 3D objects orthographically projected into an image. Since depth information is set to 0 (same as Figure 3), the signatures and poses of objects correspond to the (x,y) locations of the image. While ConvNet converges very fast to 0 training error, there is no generalization to novel viewpoints of 3D patterns.

References

[1] Q. Liao and T. Poggio, "Object-oriented deep learning," tech. rep., Center for Brains, Minds and Machines (CBMM), https://dspace.mit.edu/handle/1721.1/112103, October, 2017.

[2] Q. Liao and T. Poggio, "Exact equivariance, disentanglement and invariance of transformations," tech. rep., Center for Brains, Minds and Machines (CBMM), 2017.