Adjoint Rigid Transform Network: Task-conditioned Alignment of 3D Shapes

Keyang Zhou[1,2]   Bharat Lal Bhatnagar[1,2]   Bernt Schiele[2]   Gerard Pons-Moll[1,2]

[1]University of Tübingen, Germany

{keyang.zhou,gerard.pons-moll}@uni-tuebingen.de

[2]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
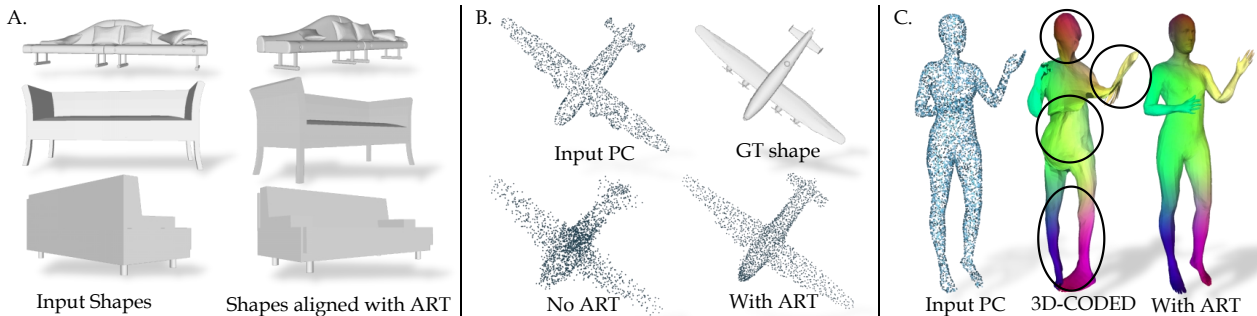
{bbhatnag, schiele}@mpi-inf.mpg.de

Figure 1: Aligning 3D data is a hard problem which typically requires manual intervention, and most 3D learning SotA methods rely on good alignment. We propose Adjoint Rigid Transform (ART) Network, a self-supervised module which can be added to existing 3D networks to boost performance on a variety of tasks. Here we show (A) automatic shape alignment with ART; and use ART to improve (B) shape auto-encoding with PointNet [39] and (C) human mesh registration with 3D-CODED [23].

Abstract

Most learning methods for 3D data suffer significant performance drops when the data is not carefully aligned to a canonical orientation. Aligning real world 3D data collected from different sources is non-trivial and requires manual intervention. In this paper, we propose the Adjoint Rigid Transform (ART) Network, a neural module which can be integrated with a variety of 3D networks to significantly boost their performance. ART learns to rotate input shapes to a learned canonical orientation, which is crucial for a lot of tasks such as shape reconstruction, interpolation, non-rigid registration, and latent disentanglement. ART achieves this with self-supervision and a rotation equivariance constraint on predicted rotations. With only self-supervision, ART facilitates learning a canonical orientation for both rigid and nonrigid shapes, which leads to a notable boost in performance of aforementioned tasks. Our code and model are available at [2].

1. Introduction

With the rise of consumer grade 3D sensors and the popularity of a wide range of applications such as virtual and augmented reality, digital humans [32, 36] and animals [59], general objects [13] and scenes [17], there is an increasing demand for learning powerful representations from 3D data. Architectures such as PointNet [39], PointNet++ [40], or EdgeConv [51] for processing point clouds, and Graph Convolutions [12, 22, 28] for processing 3D meshes allow researchers to obtain impressive results on challenging tasks such as shape encoding [41], disentangling object shape and pose [56], shape and pose interpolation [56] and 3D human mesh registration [23, 11, 10]. However, experiments show that the performance of these methods significantly decreases when 3D objects are not aligned to a common global orientation, which is a severe limitation since real world scanned objects are not aligned.

The question we pose here is: *Can we automatically learn a network module to align shapes with only self-supervision?* The ability to do so would allow existing 3D networks to consume unaligned raw data and still keep a good performance.

The most common way to align shapes is with preprocessing, as done in ShapeNet [13] and ModelNet [54]. But this is time-consuming and not completely automatic. The Spatial Transformer Networks (STN) [26] in 2D, and its 3D analogous incorporated in PointNet [39], allow the network
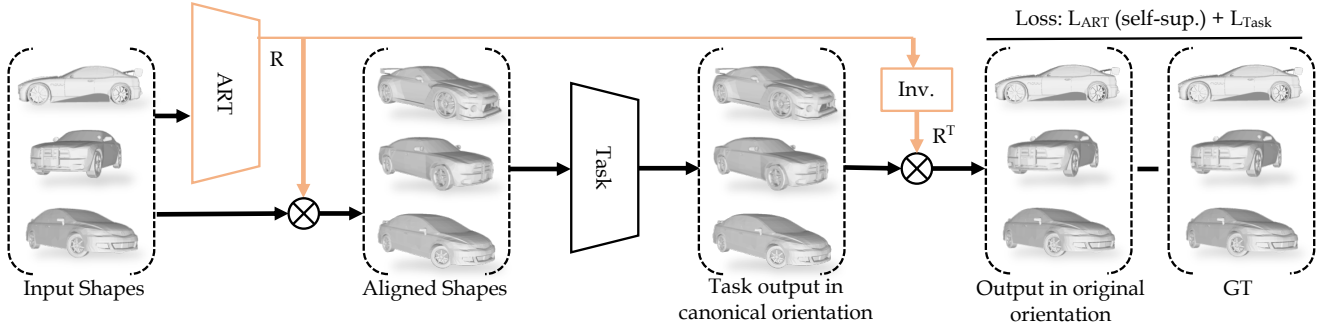
1

Figure 2: We propose a simple yet powerful module, Adjoint Rigid Transform (ART) Network that can automatically align 3D shapes to a common orientation. Given an input shape, ART predicts a rotation that maps the shape to a canonical orientation. The aligned shape is then fed to the target task and the output is rotated back to its original orientation. We show meshes here just for illustration purpose; the input can be either point cloud or mesh in practice.

to predict an affine transformation of the data to minimize the downstream loss. However, we experimentally show that STN alone does not achieve alignment. As a result, existing networks still struggle dealing with unaligned data.

We propose a simple yet powerful module, *Adjoint Rigid Transform (ART)* Network, which can be plugged into 3D networks with an auto-encoder backbone and learn to automatically align 3D shapes conditioned on tasks including shape reconstruction, shape interpolation and nonrigid registration, as shown in Fig. 1. Different from previous work on learning rotation equivariance [19, 43, 47], ART does not depend on any specially designed feature or architecture, hence directly applicable to existing models.

An overview of our method is shown in Fig. 2. Our first key idea is to learn to rotate the object to a canonical orientation, perform the target task, and rotate it back to its original pose, see Fig. 2. This leads to canonical orientations which are convenient for the task, but many such orientations can arise during learning due to local minima, which hinders the network from achieving optimal performance. To obtain a single canonical orientation per shape class, we show that the rotation predictor needs to be rotation equivariant. Hence our second key idea is to impose a self-supervised equivariance loss during training. The remarkable result is that by just using self-supervision, ART can discover a single canonical orientation of general objects, which leads to a significant boost in task performance. This is desirable as downstream networks can focus on the task instead of devoting capacity to learning rotation equivariance. Our contributions are as follows:

- We propose a novel rotation equivariance loss, which allows ART to learn a canonical orientation conditioned on the target task. The alignment quality of ART is shown to be superior to previous works.

- We demonstrate that a wide range of 3D tasks lose performance when the data is not aligned. ART signifi-

cantly improves the performance of existing methods on such tasks.

- We show the general applicability of ART spanning both rigid and non-rigid objects, and different input modalities such as meshes and point clouds.

2. Related Work

Existing methods lose performance for a wide range of 3D tasks when the data is not aligned to a canonical orientation. In this section we first discuss works that address this issue by learning model features invariant or equivariant to global orientation. Next, we discuss works that explicitly align 3D data to a canonical orientation. Our method is related to both categories as ART can be added to existing methods and can align 3D shapes using self-supervision, thus making the downstream task more robust to varying global orientations.

2.1. Rotation invariance/equivariance

An interesting research direction to achieve rotation invariance is to handcraft descriptors that are invariant to rotation by design. These works are typically based on PCA [49] and geometric properties such as distances between pairs of points [31] (in 2D) or spherical harmonics [4] (in 3D). A major limitation with the image-based descriptors is that they need to be manually designed per task which is non-trivial and difficult to scale and generalise to 3D tasks. Another approach for learning rotation invariant representation is to bake rotation invariance into the network filters. Rotation invariance within a deep network can be obtained by explicitly rotating the feature maps or convolutional filters [58, 18, 16, 52, 53, 45]. Recently, Sun *et al*. [46] proposed to learn a canonical frame by training with pairs of randomly rotated shapes. Our work follows a similar strategy, but we directly predict the canonical transformation from input shapes instead of using a capsule-based

architecture. Hence ART can be seamlessly integrated into different shape representations and backbone models

Spatial Transformers Jaderberg *et al*. [25] designed Spatial Transformer Network (STN), a differentiable module that manipulates the input image and feature maps by learned geometric transformations. Since it was first proposed, STN and its variants achieved promising results on image recognition [5], image morphing [21] and image alignment [8, 29].

More related to our approach are works that apply STN to point clouds. PointNet [39] inserted STN to both the input layer and feature maps as part of its architecture. ITN [55] constrained the predicted transformations to be rotations and took an iterative procedure to gradually transform the input point clouds in small steps. Wang *et al*. [50] learned a combination of affine, projective and deformable transformations to dynamically update local patches for feature aggregation. Recently, Fang *et al*. [20] encoded spatial direction information of points using spatial transformers and defined an anisotropic filter for point clouds.

STN freely predicts affine transformations to warp the input and the feature maps. Although rotation is a subset of affine transformations, STN takes no measure to ensure that the network maps all the input data to a canonical orientation, which is crucial for many 3D tasks. ART on the other hand predicts rotations and explicitly encourages data to be aligned in a canonical orientation, hence promoting better performance of downstream tasks.

2.2. 3D data alignment

Aligning data to a consistent frame of reference is a crucial preprocessing step for many applications [48]. When the correspondences between two shapes are available, the rigid transformation relating them can be analytically computed [6]. But in real world scenarios where correspondences are often unknown, Iterative Closest Points (ICP) [15, 9] and its variants [38] are more widely used. Since the iterative procedure of ICP starts from an initial guess of transformation, it can easily get trapped in local minima due to bad initialization. Recently, Liu *et al*. [30] proposed an approach to learn unsupervised correspondences which can be used as an alternative to ICP.

For aligning a collection of shapes, Principal Component Analysis (PCA) offers a naïve solution by matching the principal axes of every shape to those of the reference shape. Although aligning with PCA alone is error-prone, PCA is the basis for many alignment algorithms [42, 27]. For symmetric objects, reflection symmetry can also be used to further improve the alignment [37, 14, 34]. Huang *et al*. [24] discretized the transformation sampling space for each shape and formulated joint alignment as a Markov Random Field optimization problem. Averkiou *et al*. [7]
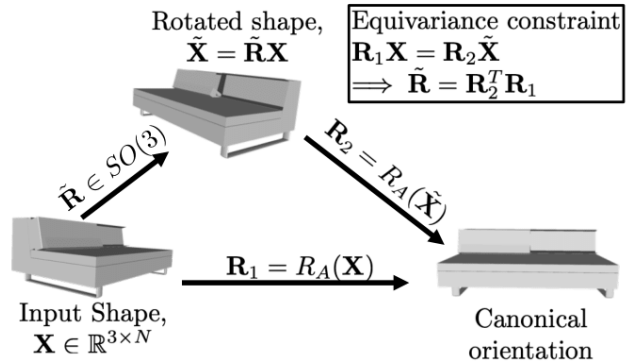

Figure 3: Equivariance constraint for ART. Given an input shape $\mathbf{X} \in \mathbb{R}^{3 \times N}$, we sample a random rotation $\tilde{\mathbf{R}}$ and obtain the rotated shape $\tilde{\mathbf{X}} = \tilde{\mathbf{R}}\mathbf{X}$. Our equivariance constraint forces the network R_A to map $\mathbf{X}$ and $\tilde{\mathbf{X}}$ to the same canonical space.

observed that ambiguities in alignment often arise in only a few candidate orientations. They reduced the search space and evaluated candidate alignments by analyzing the autocorrelation descriptors of shapes. We compare ART with a PCA baseline and Averkiou *et al*. [7]. We show that our method outperforms them both.

3. Adjoint Rigid Transform Network

In this section, we describe the general formulation of Adjoint Rigid Transform (ART) Network and how it integrates with the target task. We also introduce the equivariance loss which encourages the learned canonical orientation to be unique.

3.1. Canonical Rotation Prediction

Consider a function $g : \mathcal{X} \mapsto \mathcal{Y}$ which takes a matrix of points $\mathbf{X} \in \mathcal{X}$ as input, and let $\mathbf{R} \in \mathrm{SO}(3)$ be a rotation matrix. It is often useful to obtain g as a composition of functions

$$g(\mathbf{X}) = \mathbf{R}^T g'(\mathbf{R}\mathbf{X}). \qquad (1)$$

The idea is to rotate the input shape to a canonical orientation where processing $g'(\cdot)$ is more natural, and then rotate the output back with $\mathbf{R}^T$. This is what we refer to as an adjoint transform, as is popularly used in robotics [35].

The adjoint transform in Eq. 1 inspires the design of ART. In perception tasks however, objects can appear in multiple orientations, and the canonical coordinate frame is not defined a priori. Our task thus boils down to predicting a rotation $\mathbf{R}$ which aligns object $\mathbf{X}$ to a canonical view without using any pose labeling. Suppose we have $\mathbf{X} \in \mathbb{R}^{3 \times N}$ as input. $\mathbf{X}$ contains unordered point coordinates for point cloud input or ordered vertex coordinates for mesh input. We confine the predicted transformation to rotation for the following considerations: i) Orientation is often the most

notable factor of variation in unprocessed 3D data. ii) Rotation preserves the intrinsic geometry of the shape which is desirable for a number of tasks. iii) Unlike general affine transformations, rotation is guaranteed to be invertible. Rotations can be conveniently inverted by taking the transpose. This choice ensures the well-definedness and efficiency of adjoint transformations.

Hence, we learn a mapping $R_A(\mathbf{X}) : \mathcal{X} \mapsto \mathrm{SO}(3)$ from the input shape $\mathbf{X}$ to the canonical rotation $\mathbf{R}$. We denote the downstream 3D network by g' and denote the network with ART module as a whole by g. With the predicted rotation $R_A(\mathbf{X})$, Eq. 1 can be formulated as:

$$g(\mathbf{X}) = R_A^T(\mathbf{X})g'(R_A(\mathbf{X})\mathbf{X}) \qquad (2)$$

which replaces the known rotation in Eq. 1 with a learned one. Note that when the predicted rotation is identity $R_A = \mathbf{I}_{3\times3}$, we are left with a standard downstream network.

ART can be trained end-to-end with downstream tasks where the expected output orientation is the same as input (see Fig. 2). Shape auto-encoding is a typical use case for ART. In this case, the desired output is the input shape itself, which can be imposed with the following modified self-reconstruction loss

$$L_{\mathrm{recon}} = d\left(\mathbf{X}, R_A^T(\mathbf{X})g'(R_A(\mathbf{X})\mathbf{X})\right), \qquad (3)$$

where the loss function $d : \mathbb{R}^{3\times N} \times \mathbb{R}^{3\times M} \mapsto \mathbb{R}$ can be Chamfer distance for point clouds or vertex-to-vertex distance for meshes.

3.2. Rotation Equivariance

Predicting rotations as in Eq. 2 with an objective function from Eq. 3 does not ensure alignment. ART can learn multiple canonical orientations for a shape collection and easily get trapped in local minima. This is undesirable for both shape alignment and the target task. A necessary condition for learning unique canonical orientation is that the same shape in different input orientations should be transformed to the same canonical orientation, mathematically $R_A(\mathbf{X})\mathbf{X} = R_A(\mathbf{R}\mathbf{X})\mathbf{R}\mathbf{X} \ \forall \mathbf{R} \in \mathrm{SO}(3)$. From this it follows that the rotation predictor R_A should be equivariant to input orientations, *i.e.* $R_A(\mathbf{R}\mathbf{X}) = R_A(\mathbf{X})\mathbf{R}^T$. We enforce this constraint using a rotation equivariance loss (Fig. 3).

During training, given an input $\mathbf{X} \in \mathbb{R}^{3\times N}$, we uniformly sample a rotation $\tilde{\mathbf{R}} \in \mathrm{SO}(3)$ and obtain a different orientation of $\mathbf{X}$, $\tilde{\mathbf{X}} = \tilde{\mathbf{R}}\mathbf{X}$. Suppose we have $\mathbf{R}_1 = R_A(\mathbf{X})$ and $\mathbf{R}_2 = R_A(\tilde{\mathbf{X}})$. Then from the unique orientation constraint $\mathbf{R}_1\mathbf{X} = \mathbf{R}_2\tilde{\mathbf{X}}$ we can derive

$$\tilde{\mathbf{R}} = \mathbf{R}_2^T\mathbf{R}_1. \qquad (4)$$

Since $\tilde{\mathbf{R}}$ is known, we can turn Eq. 4 into a loss term

$$L_{\mathrm{rot\_matrix}} = \left\| \tilde{\mathbf{R}} - \mathbf{R}_2^T\mathbf{R}_1 \right\|_2^2. \qquad (5)$$

Note that $L_{\mathrm{rot\_matrix}}$ is imposed on rotation matrices. However, objects such as tables can have rotational symmetry. Suppose the rotation symmetry group of $\mathbf{X}$ is of order n. Then there are n possible rotation matrices that will leave shape $\mathbf{X}$ unchanged, so there is not a well-defined groundtruth. To handle the potential ambiguity in rotational symmetry, we add another loss term:

$$L_{\mathrm{rot\_chamfer}} = d_{\mathrm{CD}}\left(\tilde{\mathbf{X}}, \mathbf{R_2}^T\mathbf{R_1}\mathbf{X}\right), \qquad (6)$$

where d_{CD} is the symmetric Chamfer distance. The intuition behind this loss term is that shapes remain invariant to distinct elements of their symmetry groups.

Combining these two terms, we have

$$L_{\mathrm{ART}} = \lambda_1 L_{\mathrm{rot\_matrix}} + \lambda_2 L_{\mathrm{rot\_chamfer}}. \qquad (7)$$

3.3. Implementation Details

When the input shapes reside on the same plane, sampling one rotation $\tilde{R}$ per iteration suffices for Eq. 5. For general $\mathrm{SO}(3)$ rotations, we sample three $\tilde{R}$ instead. We also took the approach of [55] and iteratively apply ART to refine rotation predictions.

We adopt different architectures for Adjoint Rigid Transform Network according to different types of input. For point clouds, we use PointNet [39] as backbone. PointNet is more efficient in both training and inference compared with more sophisticated architectures for point cloud processing such as PointNet++ [40], adding minimal overhead to the downstream network. For meshes, we simply use mesh down-sampling layers [41] and fully-connected layers. We use the continuous rotation representation proposed in [57] for rotation prediction. Details about the architecture are in the supplementary.

We need to ensure that ART does not disturb training when the dataset is already in alignment, so we initialize it to predict the identity matrix. In the following experiments, we set $\lambda_1 = 0.02$, and set $\lambda_2 = 0.05$ for table category and $\lambda_2 = 0$ otherwise. Inputs are centered and normalized to fit within a unit ball.

4. Experiments

In this section, we evaluate the effectiveness of ART through extensive experiments. We demonstrate the usefulness of ART on several tasks for 3D data involving rigid (ShapeNet [13]) and non-rigid (humans [33]) objects. We also show that ART works seamlessly across different shape representations such as point clouds and meshes.

4.1. Point Cloud Auto-encoding

Shape auto-encoding is a fundamental task in unsupervised feature learning of point clouds. Prior works typically report performance on datasets consisting of axis-aligned
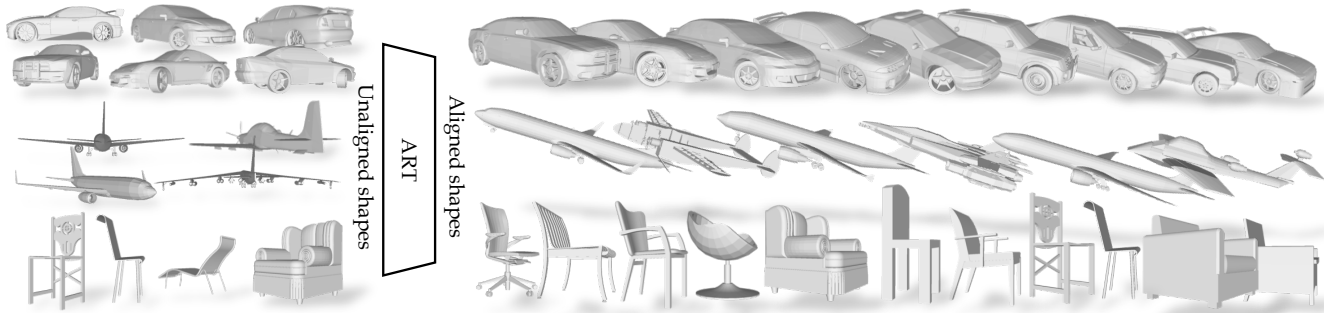
Figure 4: ART can align 3D shapes with just self-supervision. Given input shapes in arbitrary orientations (left), ART can align them to a common orientation (right).
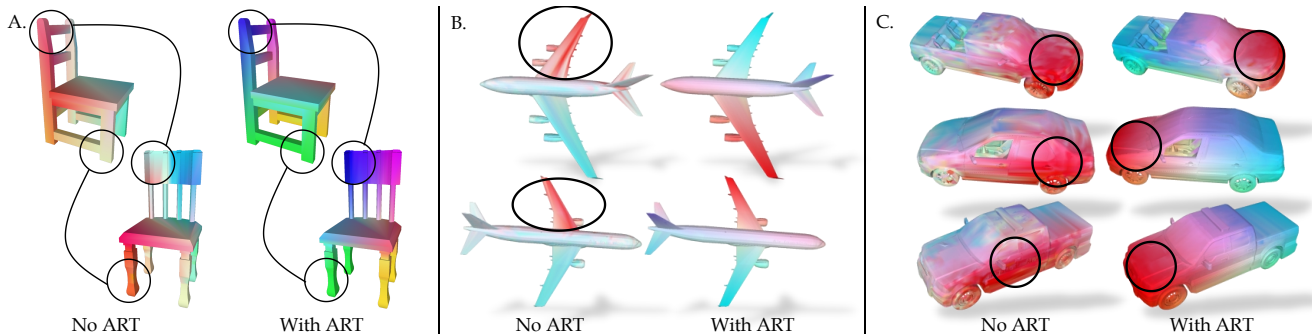

Figure 5: ART can establish correspondences between different objects within a ShapeNet category even when the shapes are not aligned. In A, we show that baseline [3] does not assign same colours to the corresponding parts. In B, we show that [3] gets confused between the left and the right wing whereas ART does not. In C, we can see that [3] cannot find consistent correspondences as the front of the car is mapped to the rear and side of the car respectively. ART on the other hand can establish correct correspondences.

shapes such as ShapeNet [13]. But in the real world, shapes are often collected from diverse sources and are therefore unaligned. To assess the capability of ART to handle unaligned data, we perturb shapes in ShapeNetCore [13] by applying random 2D rotations around the gravity axis as well as random 3D rotations. We evaluate auto-encoding in both single-category and multi-category settings. For single-category auto-encoding, we use plane, car, chair, table and sofa categories. For multi-category auto-encoding, we jointly train on these categories. For each shape, we uniformly sample 8000 points. The training/validation/testing splits are kept as 85%/5%/10%. We use symmetric Chamfer distance as both training loss and evaluation metric.

We use the network from Achlioptas *et al*. [3] as the auto-encoder backbone. It also serves as one of our baselines. On top of it, we add ITN [55] and ART to assess gains in performance brought by each module. Moreover, we designed a PCA baseline, where we naïvely align shapes by matching their principal axes and train the plain auto-encoder on PCA-aligned data.

Table 1 reports the numerical results for point cloud auto-reconstruction on both aligned and unaligned data. We can observe a significant increase in reconstruction error when the auto-encoder trains on unaligned data. Pre-

aligning shapes with PCA alleviates the problem, but there is still a performance gap. ITN performs on par with PCA, while ART outperforms all baselines on unaligned data. When the training data is perturbed by a 2D azimuthal rotation, ART matches or even beats the performance on pre-aligned data for most categories. This performance gain drops slightly for the case of 3D rotations. Qualitative examples are shown in Fig. 10.

4.2. Shape Alignment

One of the key advantages of our method is alignment of shapes to a canonical orientation (see Fig. 4). We evaluate alignment on ShapeNet [13] plane category, which we perturbed by applying random azimuthal rotations. The plane category has a well-defined criteria for exact alignment. It also doesn't suffer from ambiguities of rotational symmetry, making it suitable for benchmarking alignment accuracy. We add results on other categories in supplementary.

Here, we use the alignment evaluation metric proposed by Averkiou *et al*. [7]. We compute the angular distance between every pair of shapes using ShapeNet groundtruth orientation as reference. We compare ART to PCA, ITN [55], and the alignment method proposed by Averkiou *et al*. [7] which assumes consistent up vectors among shapes.
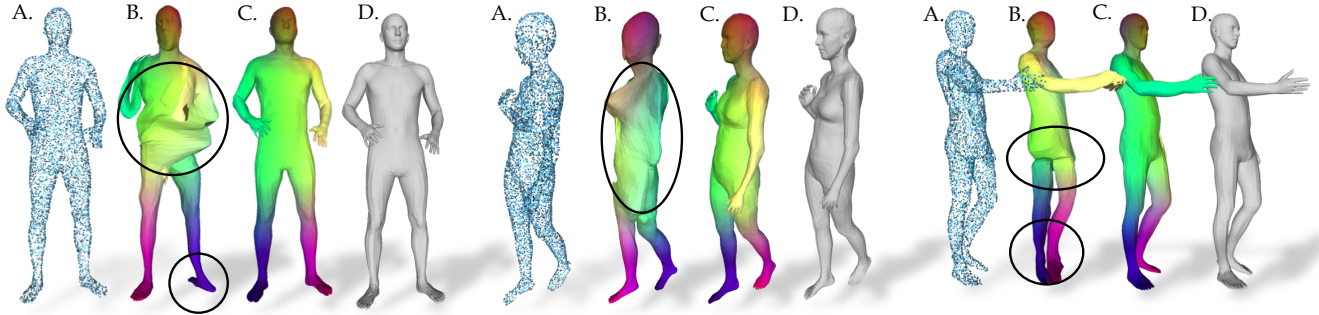
Figure 6: ART improves 3D human registration as compared to 3D-CODED [23] (single init.). In each set we show, (A) input point cloud, (B) registration with [23], (C) registration with [23]+ART (Our) and (D) GT mesh. Adding ART improves the performance of [23].
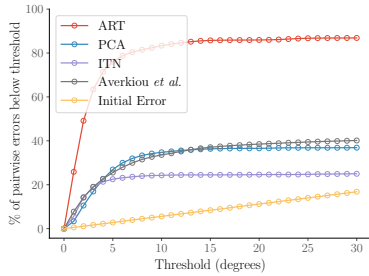

Figure 7: Percentage of shape pairs with an angular distance less than the given thresholds. ART outperforms the other three baselines by a large margin, with around 80% of shape pairs differing by less than $10°$.
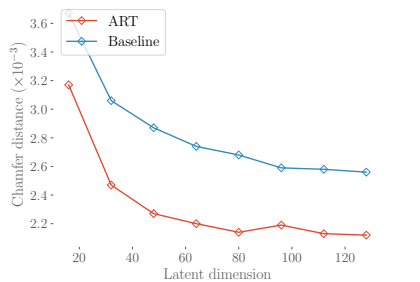

Figure 8: Reconstruction error (mean Chamfer distance) vs. size of latent code for ART and [3]. It can be seen that with a latent code of size 32, ART already outperforms [3] with a latent code of size 128.
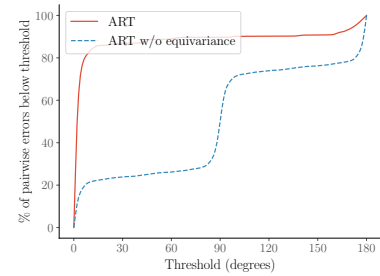

Figure 9: ART w/o equivariance constraint can learn multiple canonical orientations. Notice the sharp rise of the blue curve at $90°$ and $180°$ thresholds. It indicates the four modes of orientations on the plane category.

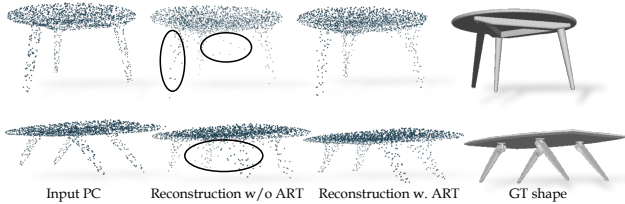Input PC — Reconstruction w/o ART — Reconstruction w. ART — GT shape

Figure 10: Shape auto-reconstruction. ART improves reconstruction quality when ShapeNet [13] objects are not aligned.

The cumulative distribution curve for pairwise errors is shown in Fig. 7. For completeness we also include the initial pairwise error for unaligned shapes. We can see that Averkiou *et al.* [7] perform on par with PCA on this dataset while ITN underperforms, with over half of the shape pairs differing by more than $30°$. This is because all these methods get confused by the near-symmetries of planes (e.g. matching plane tip to tail, or wings to fuselage). ART has two advantages in this regard, i) the downstream autoencoding task allows ART to learn semantically meaningful features to better disambiguate parts of the plane, and ii) our equivariance constraint forces ART to chose a unique canonical orientation for all planes. More qualitative results

on 2D and 3D alignment will be shown in supplementary.

4.3. Shape Interpolation

Shape interpolation is a challenging task for unaligned shapes as linear interpolation in feature space cannot handle the highly nonlinear global orientation. In Fig. 12, we show that interpolation with [3] severely distorts the shape when the source and target shapes have different orientations. Notably, ART does not suffer from this as it brings both the source shape and target shape to the same orientation where interpolation becomes meaningful.

4.4. Shape Correspondence Prediction

We show that our method can be used to predict dense correspondences between objects within a ShapeNet [13] category. The input to our shape auto-encoder is an unordered point cloud of an object in arbitrary orientation. It predicts a fixed number of points as output, which approximates the input point cloud. This allows us to establish correspondences across shapes as each shape is represented as deformations of the same fixed set of points. We compare the correspondences predicted by [3] with and without ART. Notice that the model doesn't have access to super-

Method	Data	Single-category					Multi-categories
		Plane	Chair	Car	Table	Sofa	
a) AE [3]	pre-aligned	1.16	2.20	1.81	2.37	2.21	2.02
b) AE	unaligned 2D	1.86	3.20	2.39	2.96	3.14	2.54
c) +PCA	unaligned 2D	1.34	2.42	1.91	3.14	2.41	2.26
d) +ITN [55]	unaligned 2D	1.37	2.29	1.89	2.38	2.20	2.10
e) +ART (Our)	unaligned 2D	**1.16**	**2.15**	**1.87**	**2.32**	**2.13**	**2.03**
f) AE	unaligned 3D	3.31	3.83	3.37	5.18	3.84	3.53
g) +PCA	unaligned 3D	1.38	2.44	1.97	3.46	2.61	2.41
h) +ITN	unaligned 3D	1.37	2.55	1.89	2.85	2.46	2.43
i) +ART (Our)	unaligned 3D	**1.22**	**2.41**	**1.88**	**2.38**	**2.16**	**2.26**

Table 1: We evaluate our approach on point cloud auto-encoding by comparing to multiple baselines in different settings. Numbers are reported in Chamfer distances ($\times 10^{-3}$). We show that performance of existing method [3] drops significantly between aligned and unaligned data. Aligning shapes with PCA and ITN [55] improves performance but our method clearly outperforms all the baselines, often even matching the *oracle* performance on pre-aligned data. We highlight the lowest error in each training setting with boldface.

vision of correspondence, but point-wise correspondences naturally arise when adding ART since the output space of the auto-encoder now has a consistent global orientation. The qualitative results are shown in Fig. 5.

4.5. Human Body Registration

3D-CODED, proposed by Groueix *et al*. [23], is a popular learning-based human mesh registration approach. Given an unordered point cloud as input, 3D-CODED learns to deform a pre-defined human body template mesh to match the point cloud. Since 3D-CODED was trained on synthetic shapes with consistent global orientations, it can only reconstruct shapes in that particular orientation. When dealing with real world scans without alignment, 3D-CODED applies multiple initializations to find the optimal orientation – it rotates the scan with ~100 uniformly sampled rotations around the gravity axis.

We show that adding ART to their method solves this problem. ART learns a canonical orientation for humans during training. At inference time, it only takes a single forward pass to transform the input to the canonical orientation, which is both faster and more accurate than sampling rotations. We compare the performance of ART with 3D-CODED [23] in both single-initialization and multiple-initialization cases. We test on registered meshes from Renderpeople [1] and AMASS [33]. We can see from Table 2 that ART consistently outperforms 3D-CODED by a large margin, even though we do not require 100 initializations. The qualitative results are shown in Fig. 6.

4.6. Human Pose Transfer

We use the SotA unsupervised model proposed by Zhou *et al*. [56] for the task of pose transfer. The input to their method is a registered 3D human mesh and they decompose the mesh deformations into shape and pose components. Like [56], we train on AMASS [33], a human motion cap-

Method	Dataset Renderpeople	AMASS
3D-CODED Single Init.	193.0	50.0
3D-CODED Multi Init.	23.8	33.9
ART (Our)	**16.9**	**16.8**

Table 2: Human mesh registration. We report vertex-to-vertex error in mm. 3D-CODED+ART with single initialization outperforms 3D-CODED [23] with multiple (~100) initializations.

Method	Supervised	Unsupervised
Zhou *et al*. [56]	15.44	19.43
ART (Our)	**9.16**	**17.98**

Table 3: ART for human pose transfer. ART improves the performance of [56] in both supervised and unsupervised setups. We report vertex-to-vertex error in mm.

ture dataset parametrized by SMPL model [32]. We evaluate model performance on the task of pose transfer, where we reconstruct a human body from the shape of one subject and the pose from another subject. Since we have access to the underlying SMPL parameters, we utilize SMPL model to generate pseudo-groundtruth for evaluation. Following the practice of [56], we also trained a supervised model from SMPL pseudo-groundtruth. Table 3 summarizes pose transfer errors. ART significantly lowers the pose transfer error in both supervised and unsupervised setups.

4.7. Human Pose Interpolation

Another limitation of [56] is that it cannot interpolate poses between two humans with very different global orientations. Fig. 11 shows pose interpolation results when there is a large global rotation between source and target. In [56], pose interpolation is done by linearly interpolating between source and target pose codes. However, the intermediate pose codes do not always lie on the pose manifold, caus-
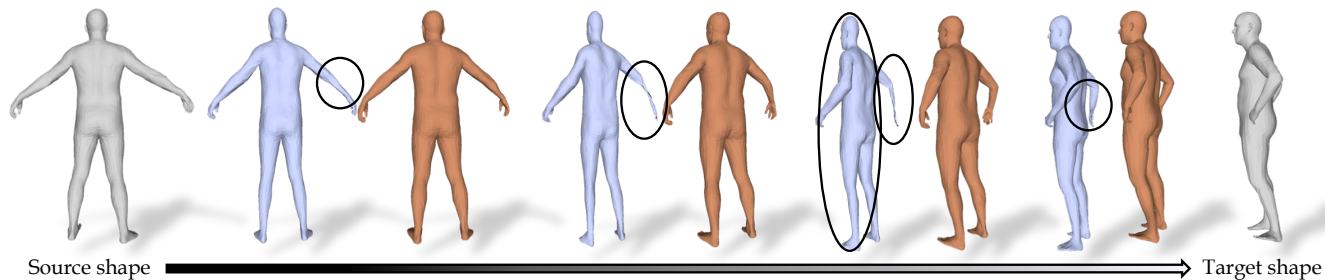
Figure 11: Pose interpolation *without* ART (blue) leads to squeezing artifacts whereas +ART (brown) handles global rotations well. We use Zhou *et al*. [56] for pose interpolation and compare performance with and without adding ART to the method.
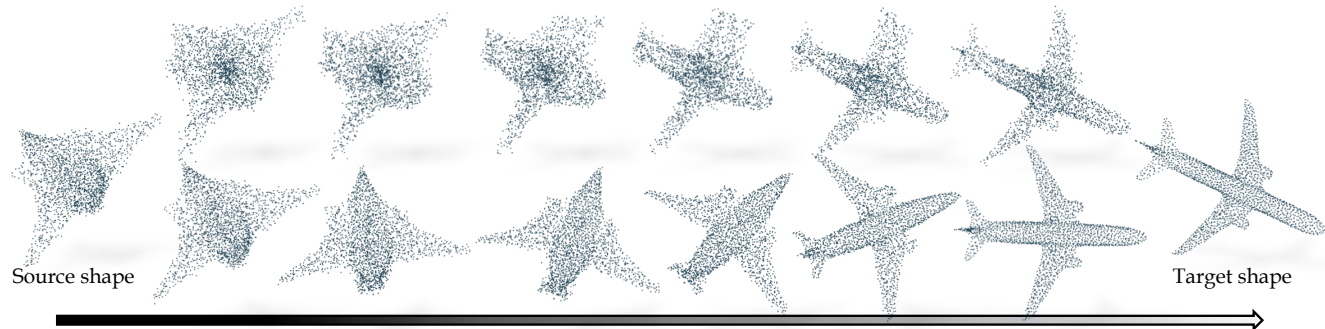

Figure 12: Shape interpolation *without* ART (top) leads to severe distortions due to different orientations between source and target shapes. +ART (bottom) on the other hand aligns the source and the target shape to a canonical orientation, resulting in smooth interpolation.

ing the squeezing artifact. Since ART can explicitly factor global rotations out, pose codes in our approach only represent articulation and not global rotation. To interpolate pose, we simply apply linear interpolation to pose codes, and spherical linear interpolation [44] to global rotations predicted by ART. Fig. 11 clearly demonstrates the strength of our method. See supplementary for more results.

4.8. Analysis and Ablation

Size of latent code vs. performance We show that a point cloud auto-encoder [3] cannot learn to encode shape orientation despite using a large latent dimension. In this experiment, we gradually increase the size of the latent code and study the performance improvement in the task of multi-category point cloud reconstruction. It can be seen from Fig. 8 that while the error drops with an increasing latent dimension, the gap between the baseline and ART remains. ART with 32 latent dimensions already outperforms [3] with 128 latent dimensions.

Importance of the equivariance constraint We implement a baseline where ART is free to predict arbitrary rotations, *i.e.* we do not enforce equivariance constraint. It can be seen from Fig. 9 that the quality of alignment without the equivariance constraint is poor as the network is free to pick multiple canonical orientations without the constraint.

5. Conclusion

Learning 3D representations is a challenging task and methods across a wide range of tasks rely on aligned data. Obtaining this alignment in real world scenarios is not trivial and often requires a lot of manual effort. We propose a simple module, *Adjoint Rigid Transform (ART) Network* that can automatically align 3D data conditioned on a set of target tasks. What makes ART effective is the semantic features learned by the auto-encoder coupled with a rotation equivariance constraint which results in a canonical orientation for input shapes. ART can be easily integrated into existing systems, can work with point clouds and meshes, and can be trained with self-supervision. We experimentally show that ART significantly boosts the performance of existing methods for shape auto-encoding, alignment, and interpolation on rigid objects, and human registration, pose transfer, and interpolation on non-rigid humans.

References

[1] Renderpeople. https://renderpeople.com/. 7

[2] https://virtualhumans.mpi-inf.mpg.de/art/. 1

[3] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018. 5, 6, 7, 8

[4] Yasseen Almakady, Sasan Mahmoodi, Joy Conway, and Michael Bennett. Rotation invariant features based on three dimensional gaussian markov random fields for volumetric texture classification. *Computer Vision and Image Understanding*, 194:102931, 2020. 2

[5] Roberto Annunziata, Christos Sagonas, and Jacques Calì. Destnet: Densely fused spatial transformer networks. *arXiv preprint arXiv:1807.04050*, 2018. 3

[6] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987. 3

[7] Melinos Averkiou, Vladimir G Kim, and Niloy J Mitra. Autocorrelation descriptor for efficient co-alignment of 3d shape collections. In *Computer Graphics Forum*, volume 35, pages 261–271. Wiley Online Library, 2016. 3, 5, 6

[8] Anil Bas, Patrik Huber, William AP Smith, Muhammad Awais, and Josef Kittler. 3d morphable models as spatial transformer networks. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 904–912, 2017. 3

[9] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992. 3

[10] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Combining implicit function learning and parametric models for 3d human reconstruction. In *European Conference on Computer Vision (ECCV)*. Springer, August 2020. 1

[11] Bharat Lal Bhatnagar, Cristian Sminchisescu, Christian Theobalt, and Gerard Pons-Moll. Loopreg: Self-supervised learning of implicit surface correspondences, pose and shape for 3d human mesh registration. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020. 1

[12] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Michael Bronstein, and Stefanos Zafeiriou. Neural 3d morphable models: Spiral convolutional networks for 3d shape representation learning and generation. In *The IEEE International Conference on Computer Vision (ICCV)*, 2019. 1

[13] 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. 1, 4, 5, 6

[14] M. Chaouch and A. Verroust-Blondet. A novel method for alignment of 3d models. In *2008 IEEE International Conference on Shape Modeling and Applications*, pages 187–195, 2008. 3

[15] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 3

[16] Taco S. Cohen and Max Welling. Steerable cnns. *International Conference on Learning Representations ICLR*, 2017. 2

[17] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 1

[18] Sander Dieleman, Jeffrey De Fauw, and Koray Kavukcuoglu. Exploiting cyclic symmetry in convolutional neural networks. volume 48 of *Proceedings of Machine Learning Research*, pages 1889–1898, New York, New York, USA, 20–22 Jun 2016. PMLR. 2

[19] 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. 2

[20] Yuan Fang, Chunyan Xu, Zhen Cui, Yuan Zong, and Jian Yang. Spatial transformer point convolution. *arXiv preprint arXiv:2009.01427*, 2020. 3

[21] N. Fish, R. Zhang, L. Perry, D. Cohen-Or, E. Shechtman, and C. Barnes. Image morphing with perceptual constraints and stn alignment. *Computer Graphics Forum*, 39(6):303–313, 2020. 3

[22] Shunwang Gong, Lei Chen, Michael Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019. 1

[23] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. 3d-coded : 3d correspondences by deep deformation. In *ECCV*, 2018. 1, 6, 7

[24] Qi-Xing Huang, Hao Su, and Leonidas Guibas. Fine-grained semi-supervised labeling of large shape collections. *ACM Transactions on Graphics (TOG)*, 32(6):1–10, 2013. 3

[25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015. 3

[26] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2017–2025, 2015. 1

[27] Michael M Kazhdan. *Shape representations and algorithms for 3D model retrieval*. PhD thesis, Princeton University Princeton, 2004. 3

[28] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*, 2017. 1

[29] Chen-Hsuan Lin and Simon Lucey. Inverse compositional spatial transformer networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2568–2576, 2017. 3

[30] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence. In *Advances in Neural Information Processing Systems (NeurIPS)*, December 2020. 3

[31] Yinlong Liu, Chen Wang, Zhijian Song, and Manning Wang. Efficient global point cloud registration by matching rotation invariant features through translation search. In *Computer Vision – ECCV 2018*, pages 460–474, Cham, 2018. Springer International Publishing. 2

[32] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graphics (Proc. SIGGRAPH Asia)*, 34(6):248:1–248:16, Oct. 2015. 1, 7

[33] Naureen Mahmood, Nima Ghorbani, Nikolaus F. Troje, Gerard Pons-Moll, and Michael J. Black. AMASS: Archive of motion capture as surface shapes. In *International Conference on Computer Vision*, pages 5442–5451, Oct. 2019. 4, 7

[34] P. Minovic, S. Ishikawa, and K. Kato. Symmetry identification of a 3-d object represented by octree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(5):507–514, 1993. 3

[35] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994. 3

[36] Leonid Pishchulin, Stefanie Wuhrer, Thomas Helten, Christian Theobalt, and Bernt Schiele. Building statistical shape spaces for 3d human modeling. *Pattern Recognition*, 67:276–286, 2017. 1

[37] Joshua Podolak, Philip Shilane, Aleksey Golovinskiy, Szymon Rusinkiewicz, and Thomas Funkhouser. A planar-reflective symmetry transform for 3d shapes. *ACM Transactions on Graphics (TOG)*, 25(3):549–559, July 2006. 3

[38] François Pomerleau, Francis Colas, and Roland Siegwart. A review of point cloud registration algorithms for mobile robotics. *Foundations and Trends® in Robotics*, 4(1):1–104, 2015. 3

[39] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 3, 4

[40] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1, 4

[41] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 704–720, 2018. 1, 4

[42] Konstantinos Sfikas, Theoharis Theoharis, and Ioannis Pratikakis. Rosy+: 3d object pose normalization based on pca and reflective object symmetry with application in 3d object retrieval. *International Journal of Computer Vision*, 91(3):262–279, 2011. 3

[43] Wen Shen, Binbin Zhang, Shikun Huang, Zhihua Wei, and Quanshi Zhang. 3d-rotation-equivariant quaternion neural networks. *arXiv preprint arXiv:1911.09040*, 2019. 2

[44] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th annual conference on Computer graphics and interactive techniques*, pages 245–254, 1985. 8

[45] Riccardo Spezialetti, Federico Stella, Marlon Marcon, Luciano Silva, Samuele Salti, and Luigi Di Stefano. Learning to orient surfaces by self-supervised spherical cnns. *Advances in Neural Information Processing Systems*, 33:5381–5392, 2020. 2

[46] Weiwei Sun, Andrea Tagliasacchi, Boyang Deng, Sara Sabour, Soroosh Yazdani, Geoffrey E Hinton, and Kwang Moo Yi. Canonical capsules: Self-supervised capsules in canonical pose. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[47] 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. 2

[48] Victor Villena-Martinez, Sergiu Oprea, Marcelo Saval-Calvo, Jorge Azorin-Lopez, Andres Fuster-Guillo, and Robert B Fisher. When deep learning meets data alignment: A review on deep registration networks (drns). *arXiv preprint arXiv:2003.03167*, 2020. 3

[49] D. V. Vranic. An improvement of rotation invariant 3d-shape based on functions on concentric spheres. In *Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429)*, volume 3, pages III–757, 2003. 2

[50] Jiayun Wang, Rudrasis Chakraborty, and Stella X Yu. Spatial transformer for 3d points. *arXiv preprint arXiv:1906.10887*, 2019. 3

[51] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 1

[52] Maurice Weiler and Gabriele Cesa. General e(2)-equivariant steerable cnns. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 14334–14345. Curran Associates, Inc., 2019. 2

[53] Maurice Weiler, Fred A. Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR*, abs/1711.07289, 2017. 2

[54] 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

[55] Wentao Yuan, David Held, Christoph Mertz, and Martial Hebert. Iterative transformer network for 3d point cloud. *arXiv preprint arXiv:1811.11209*, 2018. 3, 4, 5, 7

[56] Keyang Zhou, Bharat Lal Bhatnagar, and Gerard Pons-Moll. Unsupervised shape and pose disentanglement for 3d meshes. In *European Conference on Computer Vision (ECCV)*, August 2020. 1, 7, 8

[57] Yi Zhou, Connelly Barnes, Lu Jingwan, Yang Jimei, and Li Hao. On the continuity of rotation representations in neural networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 4

[58] Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. Oriented response networks. In *CVPR*, 2017. 2

[59] Silvia Zuffi, Angjoo Kanazawa, David W Jacobs, and Michael J Black. 3d menagerie: Modeling the 3d shape and pose of animals. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6365–6373, 2017. 1