ScrewNet: Category-Independent Articulation Model Estimation From Depth Images Using Screw Theory

Ajinkya Jain
Department of Mechanical Engineering
University of Texas at Austin, USA
ajinkya@utexas.edu

Rudolf Lioutikov
Department of Computer Science
University of Texas at Austin, USA
lioutikov@utexas.edu

Scott Niekum
Department of Computer Science
University of Texas at Austin, USA
sniekum@cs.utexas.edu

Abstract: Robots in human environments will need to interact with a wide variety of articulated objects such as cabinets, drawers, and dishwashers while assisting humans in performing day-to-day tasks. Existing methods either require objects to be textured or need to know the articulation model category a priori for estimating the model parameters for an articulated object. We propose ScrewNet, a novel approach that estimates an object's articulation model directly from depth images without requiring a priori knowledge of the articulation model category. ScrewNet uses screw theory to unify the representation of different articulation types and perform category-independent articulation model estimation. We evaluate our approach on two benchmarking datasets and compare its performance with a current state-of-the-art method. Results demonstrate that ScrewNet can successfully estimate the articulation models and their parameters for novel objects across articulation model categories with better on average accuracy than the prior state-of-the-art method.

Keywords: Articulation Model Estimation from depth images, Object Model Learning

1 Introduction

Human environments are populated with objects that contain functional parts, such as refrigerators, drawers, and staplers. These objects are known as articulated objects and consist of multiple rigid bodies connected via mechanical joints such as hinge joints or slider joints. A service robot will need to interact with these objects frequently while assisting humans. For manipulating such objects safely, the robot must reason about the articulation properties of the object. Safe manipulation policies for these interactions can be obtained directly either by using expert-defined control policies [1, 2] or by learning them through the robot's interactions with the objects [3, 4]. However, this approach may fail to provide good manipulation policies for all articulated objects that the robot might interact with, due to the vast diversity of articulated objects in human environments and the limited availability of interaction time. An alternative is to estimate the articulation models for such objects through observations, and then use a planning [5] or model-based RL method [4] to manipulate them effectively.

Existing methods for estimating articulation models of objects from visual data either use fiducial markers to track the relative movement between the object parts [6–8] or require textured objects so that feature tracking techniques can be used to observe this motion [9–11]. These requirements severely restrict the class of objects on which these methods can be used. An alternative approach is to use deep networks to extract relevant features from the raw images automatically for model estimation [12, 13]. However, these methods assume prior knowledge of the articulation model category (revolute or prismatic) to estimate the category-specific model parameters, which may not

be readily available for novel objects encountered by a robot in human environments. Addressing this limitation, we propose a novel architecture, ScrewNet, which uses screw theory priors to perform articulation model estimation directly from depth images without requiring prior knowledge of the articulation model category. ScrewNet unifies the representation of different articulation categories by leveraging the fact that the common articulation model categories (namely revolute, prismatic, and rigid) can be seen as specific instantiations of a general constrained relative motion between two objects about a fixed screw axis. This unified representation enables ScrewNet to estimate the object articulation models independent of the model category.

ScrewNet garners numerous benefits over existing approaches. First, it can estimate articulation models directly from raw depth images without requiring a priori knowledge of the articulation model category. Second, due to the screw theory priors, a single network suffices for estimating models for all common articulation model categories unlike prior methods [12, 13]. Third, ScrewNet can also estimate an additional articulation model category, the helical model, without making any changes in the network architecture or the training procedure.

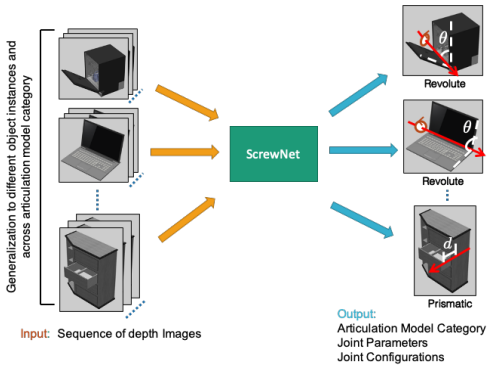


Figure 1: ScrewNet estimates the articulation model for objects directly from depth images and can generalize to novel objects within and across articulation model categories

To evaluate ScrewNet, we conduct a series of experiments on two benchmarking datasets: the simulated articulated objects dataset provided by Abbatematteo et al. [12], and the PartNet-Mobility dataset [14–16]. We test the performance of ScrewNet in estimating the articulation model parameters for unseen objects from depth images belonging to known and novel object classes, within and across the articulation model categories. We compare the performance of ScrewNet with a current state-of-the-art method proposed by Abbatematteo et al. [12] and three ablated versions of ScrewNet and show that it outperforms all baselines with a significant margin.

2 Related Work

Articulation model estimation from visual observations: Sturm et al. [6] proposed a probabilistic framework to learn the articulation relationships between different parts of an articulated object from the time-series observations of 6D poses of object parts [6, 17]. Pillai et al. [9] extended the framework to estimate the articulation model for textured objects directly from raw RGB images by extracting SURF features from the images and tracking them robustly. Niekum et al. [7] and Jain and Niekum [11] have explored modeling articulated objects that exhibit configuration-dependent changes in the articulation model, rather than having a single model throughout their motion. Recently, Abbatematteo et al. [12] posed the problem of articulation model parameter estimation as a regression task given a known articulation model category and proposed a mixture density network-based approach to predict model parameters using a single depth image of the scene. However, in a realistic setting, an object's articulation model category might not be available a priori to the robot.

Interactive perception (IP): IP approaches leverage the robot's interaction with the objects for generating a rich perceptual signal for robust articulation model estimation [18–20]. Katz and Brock [21] first studied IP to learn articulated motion models for planar objects [21], and later extended it to learn 3D kinematics of articulated objects [22]. In more recent works, Martín-Martín et al. [23] and Martín-Martín and Brock [10] have further extended the approach and used hierarchical recursive Bayesian filters to develop online algorithms from articulation model estimation from RGB images. However, current IP approaches still require textured objects for estimating the object articulation model from raw images, whereas, ScrewNet imposes no such requirement on the objects.

Articulated object pose estimation: For known articulated objects, the problem of articulation model parameter estimation can also be treated as an articulated object pose estimation problem. Different approaches leveraging object CAD model information [24, 25] and the knowledge of ar-

ticulation model category [12, 13, 26] have been proposed to estimate the 6D pose of the articulated object in the scene. These approaches can be combined with an object detection method, such as YOLOv4 [27], to develop a pipeline for estimating the articulation model parameters for objects from raw images. On the other hand, ScrewNet can directly estimate the articulation model for an object from depth images without requiring any prior knowledge about it.

Other approaches: Pérez-D'Arpino and Shah [28] and Liu et al. [8] have proposed methods to learn articulation models as geometric constraints encountered in a manipulation task from non-expert human demonstrations. Leveraging natural language descriptions during demonstrations, Daniele et al. [29] have proposed a multimodal learning framework that incorporates both vision and natural language information for articulation model estimation. However, all these approaches use fiducial markers to track the movement of the object, unlike ScrewNet, that works on raw images.

3 Background

Screw displacements: Chasles' theorem states that *"Any displacement of a body in space can be accomplished by means of a rotation of the body about a unique line in space accompanied by a translation of the body parallel to that line"* [30]. This line is called the screw axis of displacement, S [31, 32]. In this work, we use Plücker coordinates to represent this line. The Plücker coordinates of the line l having direction $\mathbf{l}$ and passing through the point $\mathbf{p}$, are defined as $(\mathbf{l}, \mathbf{m})$, where $\mathbf{m} = \mathbf{p} \times \mathbf{l}$ refers to the moment vector of the line [31, 32]. Two additional constraints $\|\mathbf{l}\| = 1$, and the Plücker constraint, i.e. $\mathbf{l} \cdot \mathbf{m} = 0$ ensure that the degrees of freedom of the line in space are restricted to four. The complete rigid body displacement in $SE(3)$ can then be defined as $\sigma = (\mathbf{l}, \mathbf{m}, \theta, d)$, where the linear displacement d along the axis is related to the rotation θ through the pitch h of the screw axis as $d = h\theta$. The distance between two lines $l_1 := (\mathbf{l}_1, \mathbf{m}_1)$ and $l_2 := (\mathbf{l}_2, \mathbf{m}_2)$ is calculated as:

$$\mathsf{d}((\mathbf{l}_1, \mathbf{m}1), (\mathbf{l}_2, \mathbf{m}_2)) = \begin{cases} 0, & \text{if } l_1 \text{ and } l_2 \text{ intersect} \\ \|\mathbf{l}_1 \times (\mathbf{m}_1 - \mathbf{m}_2)\|, & \text{else if } l_1 \text{ and } l_2 \text{ are parallel, i.e. } \|\mathbf{l}_1 \times \mathbf{l}_2\| = 0 \\ \dfrac{|\mathbf{l}_1 \cdot \mathbf{m}_2 + \mathbf{l}_2 \cdot \mathbf{m}_1|}{\|\mathbf{l}_1 \times \mathbf{l}_2\|}, & \text{else, } l_1 \text{ and } l_2 \text{ are skew lines} \end{cases} \tag{1}$$

Frame transformations on Plücker lines: Given a rotation matrix R and a translation vector $\mathbf{t}$ between two frames $\mathcal{F}_A$ and $\mathcal{F}_B$, a 3D line displacement matrix $\tilde{D}$ can be defined between the two frames for transforming a line $l := (\mathbf{l}, \mathbf{m})$ from frame $\mathcal{F}_A$ to frame $\mathcal{F}_B$ as:

$$\begin{bmatrix} {}^B\mathbf{l} \\ {}^B\mathbf{m} \end{bmatrix} = {}^B\tilde{D}_A \begin{bmatrix} {}^A\mathbf{l} \\ {}^A\mathbf{m} \end{bmatrix}, \text{ where, } {}^B\tilde{D}_A = \begin{bmatrix} R & \mathbf{0} \\ [\mathbf{t}]_\times R & R \end{bmatrix}, \ [\mathbf{t}]_\times = \begin{bmatrix} 0 & -t_3 & t_2 \\ t_3 & 0 & -t_1 \\ -t_2 & t_1 & 0 \end{bmatrix} \tag{2}$$

where $[\mathbf{t}]_\times$ denotes the skew-symmetric matrix corresponding to the translation vector $\mathbf{t}$, and $({}^A\mathbf{l}, {}^A\mathbf{m})$ and $({}^B\mathbf{l}, {}^B\mathbf{m})$ represents the line l in frames $\mathcal{F}_A$ and $\mathcal{F}_B$, respectively [33].

4 Approach

4.1 Problem Formulation

Given a sequence of n depth images $\mathcal{I}_{1:n}$ of relative motion between two parts of an articulated object, we wish to estimate the articulation model $\mathcal{M}$ and its parameters ϕ governing the motion between the two parts without knowing the articulation model category a priori. Additionally, we wish to estimate the configurations $q_{1:n}$ that uniquely identify different relative spatial displacements between the two parts in the given sequence of images $\mathcal{I}_{1:n}$ under model $\mathcal{M}$ with parameters ϕ. We consider articulation models with at most one degree-of-freedom (DoF), i.e. $\mathcal{M} \in \{\mathcal{M}_{\text{rigid}}, \mathcal{M}_{\text{revolute}}, \mathcal{M}_{\text{prismatic}}, \mathcal{M}_{\text{helical}}\}$. Model parameters ϕ are defined as the parameters of the screw axis of motion, i.e. $\mathsf{S} = (\mathbf{l}, \mathbf{m})$, where both $\mathbf{l}$ and $\mathbf{m}$ are three-dimensional real vectors. Each configuration q_i corresponds to a tuple of two scalars, $q_i = (\theta_i, d_i)$, defining a rotation around and a displacement along the screw axis S. We assume that the relative motion between the two object parts is governed only by a single articulation model.

4.2 ScrewNet

We propose ScrewNet, a novel architecture that given a sequence of segmented depth images $\mathcal{I}_{1:n}$ of the relative motion between two rigid objects can estimate the articulation model $\mathcal{M}$ between the objects, its parameters ϕ, and the corresponding configurations $q_{1:n}$ observed during the motion. In contrast to the comparable state-of-the-art approaches, ScrewNet does not require a priori knowledge of the articulation model category for the objects to estimate their models. ScrewNet achieves category independent articulation model estimation by representing different articulation models through a unified representation based on the screw theory [30]. We propose to use the Chasles' theorem for modeling the articulation relationship between the objects. Specifically, we propose to represent the articulation relationships between rigid objects of at most one degree-of-freedom (rigid, revolute, prismatic, and helical) as a sequence of screw displacements along a common screw axis. Under this representation, a rigid model is defined as a sequence of identity transformations (i.e., as a sequence of screw displacements having both $\theta = 0$ and $d = 0$), a revolute model as a sequence of pure rotations around a common axis (i.e., as a sequence of screw displacements with $\theta \neq 0$ and $d = 0$), a prismatic model as a sequence of pure displacements along the same axis(i.e., as a sequence of screw displacements having $\theta = 0$ and $d \neq 0$), and, a helical model as a sequence of correlated rotations and displacements along a shared axis (i.e., as a sequence of screw displacement with both $\theta \neq 0$ and $d \neq 0$).

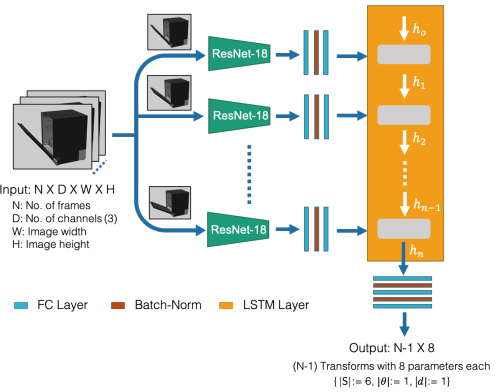


Figure 2: Taking a sequence of depth images as input, ScrewNet first extracts features from the depth images using ResNet, passes them through an LSTM layer to encode their sequential information, and then uses MLP to predict a sequence of screw displacements having a shared screw axis

Under the proposed unified representation, all articulation models with at most one DoF can be represented using the same number of parameters, i.e. 6 parameters for the common screw axis S and $2n = |\{(\theta_i, d_i) \forall i \in \{1...n\}\}|$ parameters for configurations, which enables ScrewNet to perform category independent articulation model estimation. While ScrewNet does not need to know the model category $\mathcal{M}$ to estimate the articulation model parameters, it may be beneficial to estimate the model category as well, as its knowledge can potentially reduce the number of control parameters required for manipulating the object [11]. A unified representation also allows ScrewNet to use a single network to estimate the articulation motion models across categories, unlike prior approaches that required separate networks, one for each articulation model category [12, 13]. Having a single network grants ScrewNet two major benefits: first, it needs to train fewer total parameters, and second, it allows for a greater sharing of training data across articulation categories, resulting in a significant increase in the number of training examples that the network can use. Additionally, in theory, ScrewNet can also estimate an additional articulation model category, the helical model, which was not addressed in earlier work [6, 10, 12].

Architecture: ScrewNet comprises of a ResNet-18 CNN [34], an LSTM with one hidden layer, and a 3-Layer deep MLP, connected sequentially. ResNet-18 extracts features from the depth images that are fed into the LSTM layer to encode the sequential information from the extracted features into a latent representation. Then, the MLP is used to predict a sequence of screw displacements having a common screw axis using the latent representation. The complete network is trained in an end-to-end fashion. We use a ReLU activation function for the fully-connected layers. The detailed network architecture is shown in Fig. 2. The articulation model category $\mathcal{M}$ is later deduced from the predicted screw displacements using a decision-tree based on the aforementioned properties of the screw displacements belonging to a particular model class.

Loss function: Screw displacements are composed of two major components: the screw axis S, and the corresponding configurations q_i about it. Reflecting this, we propose the following loss function

to train ScrewNet:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\mathsf{S}_{\text{ori}}} + \lambda_2 \mathcal{L}_{\mathsf{S}_{\text{dist}}} + \lambda_3 \mathcal{L}_{\mathsf{S}_{\text{cons}}} + \lambda_4 \mathcal{L}_{\text{q}} \tag{3}$$

where $\mathcal{L}_{\mathsf{S}_{\text{ori}}}$ penalizes the screw axis orientation mismatch and is calculated as the angular difference between the target and prediction screw axis orientations, $\mathcal{L}_{\mathsf{S}_{\text{dist}}}$ penalizes the spatial distance between the target and predicted screw axes and is calculated as defined in the Eqn. 1, $\mathcal{L}_{\mathsf{S}_{\text{cons}}}$ enforces the Plücker constraint ($\mathbf{l} \cdot \mathbf{m} = 0$) and the unit norm constraint on $\mathbf{l}$, $\mathcal{L}_{\text{q}}$ penalizes errors in the configurations, and λ_i correspond to scalar weights. The configurational error $\mathcal{L}_{\text{q}}$ has two parts: the rotational error $\mathcal{L}_\theta$, and the translational error $\mathcal{L}_d$, which are calculated as following:

$$\mathcal{L}_\theta = \mathbf{I}_{3,3} - R(\theta_{\text{tar}}; \mathbf{l}_{\text{tar}})\, R(\theta_{\text{pred}}; \mathbf{l}_{\text{pred}})^T, \qquad \mathcal{L}_d = \|d_{\text{tar}} \cdot \mathbf{l}_{\text{tar}} - d_{\text{pred}} \cdot \mathbf{l}_{\text{pred}}\| \tag{4}$$

where $R(\theta; \mathbf{l})$ denotes the rotation matrix corresponding to a rotation of angle θ about the axis $\mathbf{l}$. We choose this particular form of the loss function for $\mathcal{L}_{\text{q}}$, rather than a standard loss function such as an $L2$ loss, as it ensures that the network predictions are grounded in their physical meaning. By imposing a loss based on the orthonormal property of the $3D$ rotations, the proposed loss function ensures that the learned angle-axis pair $(\mathbf{l}_{\text{pred}}, \theta_{\text{pred}})$ corresponds to a rotation $R(\theta_{\text{tar}}; \mathbf{l}_{\text{tar}}) \in SO(3)$. Similarly, the loss function $\mathcal{L}_d$ calculates the difference between the two displacements along two different axes $\mathbf{l}_{\text{tar}}$ and $\mathbf{l}_{\text{pred}}$, rather than calculating the difference between the two configurations, d_{tar} and d_{pred}, which assumes that they represent displacements along the same axis. Hence, this choice of loss function ensures that the network predictions conform to the definition of a screw displacement. We empirically choose weights to be $\lambda_1 = 1, \lambda_2 = 2, \lambda_3 = 1$, and $\lambda_4 = 1$.

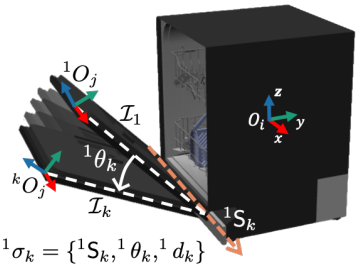


Figure 3: For generating the training labels, we first calculate the screw displacements between the temporally displaced poses of the object o_j, and later, express them in a frame of reference attached to the base object o_i

Training data generation: Training data for ScrewNet consists of sequences of depth images of objects moving relative to each other and the corresponding screw displacements. We use Mujoco [35] to render the objects in simulation and record depth images. We use the cabinet, drawer, microwave, the toaster-oven object classes from the simulated articulated object dataset provided by Abbatematteo et al. [12]. The cabinet, microwave, and toaster object classes contain a revolute joint each, while the drawer class contains a prismatic joint. We consider both left-opening cabinets and right-opening cabinets. From the PartNet-Mobility dataset [14–16], we consider the dishwasher, oven, and microwave object classes for the revolute articulation model category, and the storage furniture object class consisting of either a single column of drawers or multiple columns of drawers, for the prismatic articulation model category. Further details are presented in the appendix.

To generate the labels for screw displacements, we consider one of the objects, o_i, as the base object, and calculate the screw displacements between temporally displaced poses of the second object o_j with respect to it. Specifically, given a sequence of n images $\mathcal{I}_{1:n}$, we first calculate a sequence of $n-1$ screw displacements ${}^1\boldsymbol{\sigma}_{o_j} = \{{}^1\sigma_2, ...{}^1\sigma_n\}$, where each ${}^1\sigma_k$ corresponds to the relative spatial displacement between the pose of the object o_j in the first image $\mathcal{I}_1$ and the images $\mathcal{I}_{k,\ k\in\{2...n\}}$. Note ${}^1\boldsymbol{\sigma}_{o_j}$ is defined in the frame $\mathcal{F}_{o_j^1}$ attached to the pose of the object o_j in the first image $\mathcal{I}_1$. We can transform ${}^1\boldsymbol{\sigma}_{o_j}$ to a frame attached to the base object $\mathcal{F}_{o_i}$ by defining the 3D line motion matrix $\tilde{D}$ (Eqn. 2) between the frames $\mathcal{F}_{o_j^1}$ and $\mathcal{F}_{o_i}$ [33], and transforming the common screw axis ${}^1\mathsf{S}$ to the target frame $\mathcal{F}_{o_i}$. The configurations ${}^1q_k$ remain the same during frame transformations.

5 Experiments

We evaluate ScrewNet's performance in estimating the articulation models for objects by conducting three sets of experiments on two benchmarking datasets: the simulated articulated objects dataset provided by Abbatematteo et al. [12] and the recently proposed PartNet-Mobility dataset [14–16]. The first set of experiments evaluates ScrewNet's performance in estimating the articulation models for unseen object instances that belong to the object classes used for training the network. Next,

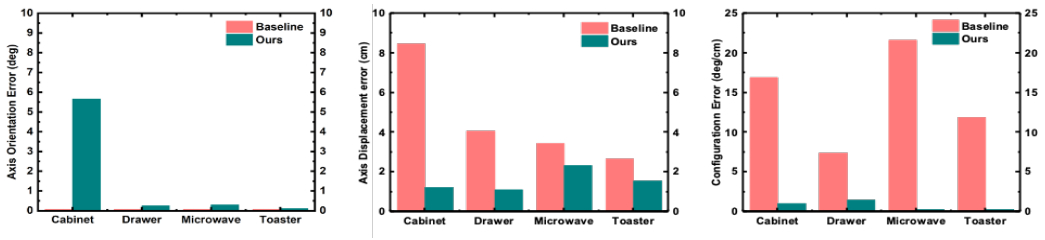


Figure 4: **[Same object class]** Mean error values for the joint axis orientations, positions, and joint configurations for 1000 test objects for each object class from the simulated articulated objects dataset [12]. Configuration errors reported in degrees for cabinet, microwave, and toaster, and in cm for drawer

we test ScrewNet's performance in estimating the model parameters for novel articulated objects that belong to the same articulation model category as seen during training. In the final set of experiments, we train a single ScrewNet on object instances belonging to different object classes and articulation model categories and evaluate its performance in cross-category articulation model estimation. We compare ScrewNet with a state-of-the-art articulation model estimation method proposed by Abbatematteo et al. [12].

In all the experiments, we assume that the input depth images are semantically segmented and contain non-zero pixels corresponding only to the two objects between which we wish to estimate the articulation model. Given this input, ScrewNet estimates the articulation model parameters for the pair of objects in an object-centric coordinate frame defined at the center of the bounding box of the object. Later for manipulating the object, an off-the-shelf object detection algorithm such as YOLOv4 [27] can be used to transform these parameters to the camera frame of the robot by defining a corresponding 3D line motion matrix ${}^{cam}\tilde{D}_{obj}$ (Eqn.2). Note while the approach proposed by Abbatematteo et al. [12] can be used to estimate the articulation model parameters directly in the camera frame, for a fair comparison to our approach, we modified the baseline to predict the model parameters in the object-centric reference frame as well.

5.1 Same object class

In the first set of experiments, we investigate whether our proposed approach can generalize to unseen object instances belonging to the object classes seen during the training. For this set of experiments, we train a separate ScrewNet and a baseline network [12] for each of the object classes and test how ScrewNet fares in comparison to the baseline under similar experimental conditions. We generate 10,000 training examples for each object class in both datasets and perform evaluations on 1,000 withheld object instances. From Fig. 4, it is evident that ScrewNet outperforms the baseline in estimating the joint axis position and the observed joint configurations by a significant margin for the first dataset. However, for the joint axis orientation estimation, the baseline method reports lower errors than the ScrewNet. Similar trends in the performance of the two methods were observed on the PartNet-Mobility dataset (see Fig. 5). ScrewNet significantly outperformed the baseline method in estimating the joint axis displacement and observed joint configurations, while the baseline reported lower errors than ScrewNet in estimating the joint axis orientations. However, for both the datasets, the errors reported by ScrewNet in screw axis orientation estimation are reasonably low ($< 5^{\circ}$), and the model parameters predicted by ScrewNet may be used directly for manipulating the object. These experiments demonstrate that under similar experimental conditions, ScrewNet can estimate the joint axis positions and joint configurations for objects better than the baseline method, while reporting reasonably low but higher errors in joint axis orientations.

5.2 Same articulation model category

Next, we investigate if our proposed approach can generalize to unseen object classes belonging to the same articulation model category. We conduct this set of experiments only on the PartNet-Mobility dataset as the simulated articulated objects dataset does not contain enough variety of object classes belonging to the same articulation model category (only 3 for revolute and 1 for prismatic). For the revolute category, we train ScrewNet and the baseline on the object instances generated

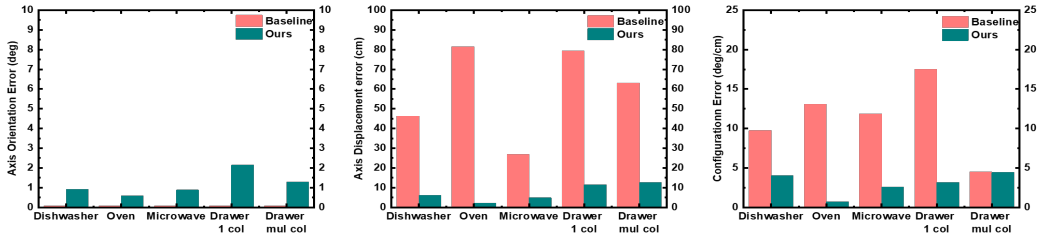


Figure 5: **[Same object class]** Mean error values for the joint axis orientations, positions, and joint configurations for 1000 test objects for each object class from the PartNet-Mobility Dataset. Configuration errors reported in degrees for dishwasher, oven, and microwave, and in cm for drawer

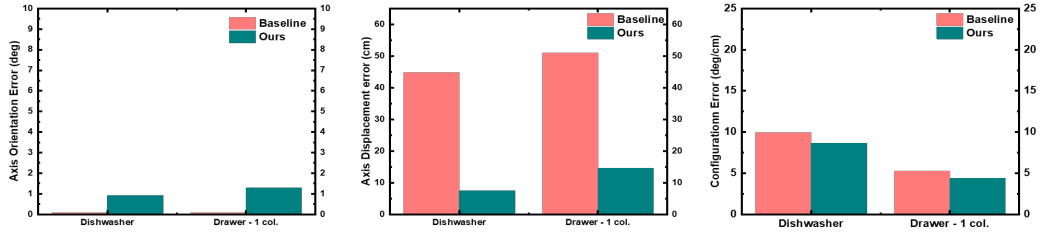


Figure 6: **[Same articulation model category]** Mean errors for the joint axis orientations, positions, and joint configurations for 1000 test objects for each object class from the PartNet-Mobility Dataset.

from the oven and the microwave object classes and test it on the objects from the dishwasher class. For the prismatic category, we train them on the objects from the storage furniture class containing multiples columns of drawers and test it on the storage furniture objects containing a single column of drawers. We train a single instance of ScrewNet and the baseline for each articulation model category and use them to predict the articulation model parameters for the test object classes. We use the same training datasets as used in the previous set of experiments. Results are reported in Fig. 6. It is evident from Fig. 7 that ScrewNet can generalize to novel object classes belonging to the same articulation model category, while the baseline fails to do so. Both methods report low errors in the joint axis orientation and the observed configurations. However, for the joint axis position, the baseline method reports mean errors of an order of magintude higher than the ScrewNet for both the articulation model categories.

5.3 Across articulation model category

Finally, we study whether ScrewNet can estimate articulation model parameters for unseen objects across the articulation model category. For this set of experiments, we train a single ScrewNet on a mixed dataset consisting of object instances belonging to all object model classes for the dataset. To test whether sharing training data across articulation categories can help in reducing the number of examples required for training, we use only half of the dataset available for each object class (5000 examples each, instead of available $10,000$ examples) while preparing the mixed dataset. We compare its performance with a baseline network that is trained specifically on the particular

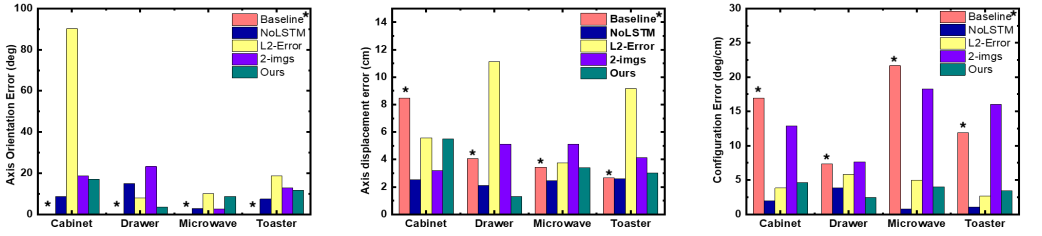


Figure 7: **[Across articulation model category]** Mean error values for the joint axis orientations, positions, and joint configurations for 1000 test objects for each object class from the simulated articulated objects dataset. Asterisk $(*)$ denote that the baseline has a significant advantage over other methods as it uses a separate network for each object class

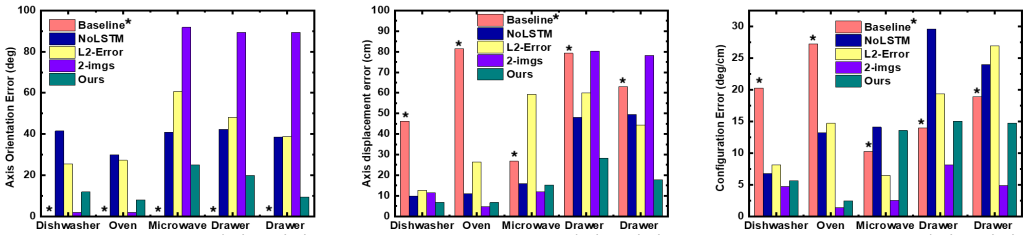


Figure 8: **[Across articulation model category]** Mean error values for the joint axis orientations, positions, and joint configurations for 1000 test objects for each object class from the simulated articulated objects dataset

object class. Additionally, we also conduct ablation studies to test the effectiveness of the various components of the proposed method, detailed descriptions of which are presented in the appendix.

Fig. 7 summarizes the results for the first dataset. Even though we use a single network to estimate the articulation model for objects belonging to different articulation model categories, the ScrewNet can perform at par or better than the baseline method for all the object model categories. ScrewNet outperforms the baseline while estimating the observed joint configurations for all object classes, even though the baseline was trained separately for each object class. For joint axis position estimation, ScrewNet reports significantly lower errors than the baseline for the cabinet and the drawer classes, and comparable errors for the microwave and the toaster classes. In estimating the joint axis orientations, both methods report comparable errors for the cabinet, drawer, and the toaster classes. However, for the cabinet object class, ScrewNet reports higher error than the baseline method, which may stem from the fact that the cabinet object class includes both left-opening and right-opening configurations that have a difference 180° in their axis orientations. On the PartNet-Mobility dataset (see Fig. 8), the performances of the methods follow similar trends, with ScrewNet outperforming the baseline method with a significant margin in estimating the joint axis positions and the observed joint configurations while reporting higher errors than the baseline in estimating the joint axis orientations. The results show that by using a unified representation, ScrewNet can perform cross-category articulation model estimation with better on average performance than the current state-of-the-art method while using only half the training examples.

In comparison to its ablated versions, ScrewNet outperforms the L2-error and the two-images versions by a significant margin for both the datasets and performs comparably to the NoLSTM version. For the first dataset, the NoLSTM version reports lower errors than ScrewNet in estimating the joint axis orientations, their positions, and the observed joint configurations for the microwave, cabinet, and the toaster classes. However, the NoLSTM version fails to generalize across articulation model categories and reports higher errors than the ScrewNet for the drawer class, sometimes even reporting NaNs as predictions. On the second dataset, ScrewNet reports much lower errors than the NoLSTM ablated version for all object model categories. These results demonstrate that for reliably estimating articulation model parameters across categories, both the sequential information available in the input and a loss function that grounds predictions in their physical meaning are crucial.

6 Conclusion

Articulated objects are common in human environments such as an oven, a dishwasher, and a cabinet, and service robots will be interacting with them frequently while assisting humans. For manipulating such objects safely, a robot will need to learn the articulation properties of such objects through raw sensory data such as RGB-D images. Current methods for estimating the articulation model of objects from visual observations either require textured objects or need to know the articulation model category a priori for estimating the articulation model parameters from the depth images. Addressing this, we propose ScrewNet that uses screw theory to unify the representation of different articulation models and performs category-independent articulation model estimation from depth images. We evaluate the performance of ScrewNet on two benchmarking datasets and compare it with a state-of-the-art method. Results demonstrate that ScrewNet can estimate articulation models and their parameters for objects across object classes and articulation model categories successfully with better on average performance than the baseline while using half the training data and without requiring to know the object articulation model category a priori.

While ScrewNet can successfully perform cross-category articulation model estimation, at present, it can only predict 1-DOF articulation models between objects. For objects with multiple DOFs, such as a cabinet with two doors, an additional image segmentation step is required to mask out all other object parts except the relevant pair of parts before feeding the data into the network. This procedure can be repeated iteratively pairwise on all object parts to estimate local relative models between object parts, that later can be combined appropriately to build a complete model for the object. A future extension can be to learn a segmentation network along with the ScrewNet so that a complete articulation model for objects with multi-DOFs can be estimated directly. Another possible future work could be to predict the articulation model parameters directly in the robot's camera frame rather than in an object-centric frame. Having direct predictions in the camera frame can help the robot to learn the articulation model for the object in an active learning fashion.

References

[1] A. Jain and C. C. Kemp. Pulling open novel doors and drawers with equilibrium point control. In *Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on*, pages 498–505. IEEE, 2009.

[2] M. Baum, M. Bernstein, R. Martin-Martin, S. Höfer, J. Kulick, M. Toussaint, A. Kacelnik, and O. Brock. Opening a lockbox through physical exploration. In *2017 IEEE-RAS 17th International Conference on Humanoid Robotics (Humanoids)*, pages 461–467. IEEE, 2017.

[3] A. Gupta, V. Kumar, C. Lynch, S. Levine, and K. Hausman. Relay policy learning: Solving long-horizon tasks via imitation and reinforcement learning. *arXiv preprint arXiv:1910.11956*, 2019.

[4] O. Kroemer, S. Niekum, and G. Konidaris. A review of robot learning for manipulation: Challenges, representations, and algorithms. *arXiv preprint arXiv:1907.03146*, 2019.

[5] A. Jain and S. Niekum. Efficient hierarchical robot motion planning under uncertainty and hybrid dynamics. In *Conference on Robot Learning*, pages 757–766, 2018.

[6] J. Sturm, C. Stachniss, and W. Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011.

[7] S. Niekum, S. Osentoski, C. G. Atkeson, and A. G. Barto. Online bayesian changepoint detection for articulated motion models. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1468–1475. IEEE, 2015.

[8] Y. Liu, F. Zha, L. Sun, J. Li, M. Li, and X. Wang. Learning articulated constraints from a one-shot demonstration for robot manipulation planning. *IEEE Access*, 7:172584–172596, 2019.

[9] S. Pillai, M. R. Walter, and S. Teller. Learning articulated motions from visual demonstration. *arXiv preprint arXiv:1502.01659*, 2015.

[10] R. Martín-Martín and O. Brock. Coupled recursive estimation for online interactive perception of articulated objects. *The International Journal of Robotics Research*, page 0278364919848850, 2019.

[11] A. Jain and S. Niekum. Learning hybrid object kinematics for efficient hierarchical planning under uncertainty. *arXiv preprint arXiv:1907.09014*, 2019.

[12] B. Abbatematteo, S. Tellex, and G. Konidaris. Learning to Generalize Kinematic Models to Novel Objects. In *Proceedings of the Third Conference on Robot Learning*, 2019.

[13] X. Li, H. Wang, L. Yi, L. J. Guibas, A. L. Abbott, and S. Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3706–3715, 2020.

[14] F. Xiang, Y. Qin, K. Mo, Y. Xia, H. Zhu, F. Liu, M. Liu, H. Jiang, Y. Yuan, H. Wang, L. Yi, A. X. Chang, L. J. Guibas, and H. Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[15] K. Mo, S. Zhu, A. X. Chang, L. Yi, S. Tripathi, L. J. Guibas, and H. Su. PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[16] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[17] J. Sturm, V. Pradeep, C. Stachniss, C. Plagemann, K. Konolige, and W. Burgard. Learning kinematic models for articulated objects. In *IJCAI*, pages 1851–1856, 2009.

[18] R. M. Martin and O. Brock. Online interactive perception of articulated objects with multi-level recursive estimation based on task-specific priors. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2494–2501. IEEE, 2014.

[19] K. Hausman, S. Niekum, S. Osentoski, and G. S. Sukhatme. Active artic lation model estimation through interactive perception. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 3305–3312. IEEE, 2015.

[20] J. Bohg, K. Hausman, B. Sankaran, O. Brock, D. Kragic, S. Schaal, and G. S. Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE Transactions on Robotics*, 33(6):1273–1291, 2017.

[21] D. Katz and O. Brock. Manipulating articulated objects with interactive perception. In *2008 IEEE International Conference on Robotics and Automation*, pages 272–277. IEEE, 2008.

[22] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *2013 IEEE International Conference on Robotics and Automation*, pages 5003–5010. IEEE, 2013.

[23] R. Martín-Martín, S. Höfer, and O. Brock. An integrated approach to visual perception of articulated objects. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5091–5097. IEEE, 2016.

[24] F. Michel, A. Krull, E. Brachmann, M. Y. Yang, S. Gumhold, and C. Rother. Pose estimation of kinematic chain instances via object coordinate regression. In *BMVC*, pages 181–1, 2015.

[25] K. Desingh, S. Lu, A. Opipari, and O. C. Jenkins. Factored pose estimation of articulated objects using efficient nonparametric belief propagation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7221–7227. IEEE, 2019.

[26] L. Yi, H. Huang, D. Liu, E. Kalogerakis, H. Su, and L. Guibas. Deep part induction from articulated object pairs. In *SIGGRAPH Asia 2018 Tech. Pap. SIGGRAPH Asia 2018*, volume 37, 2018. doi:10.1145/3272127.3275027. URL https://dl.acm.org/doi/10.1145/3272127.3275027.

[27] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[28] C. Pérez-D'Arpino and J. A. Shah. C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 4058–4065. IEEE, 2017.

[29] A. F. Daniele, T. M. Howard, and M. R. Walter. A multiview approach to learning articulated motion models. In *Robotics Research*, pages 371–386. Springer, 2020.

[30] B. Siciliano and O. Khatib. *Springer handbook of robotics*. Springer, 2016.

[31] M. T. Mason. *Mechanics of robotic manipulation*. MIT press, 2001.

[32] Y.-b. Jia. Plücker Coordinates for Lines in the Space [Lecture Notes]. pages 1–11, August 2019.

[33] A. Bartoli and P. Sturm. The 3d line motion matrix and alignment of line reconstructions. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[34] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

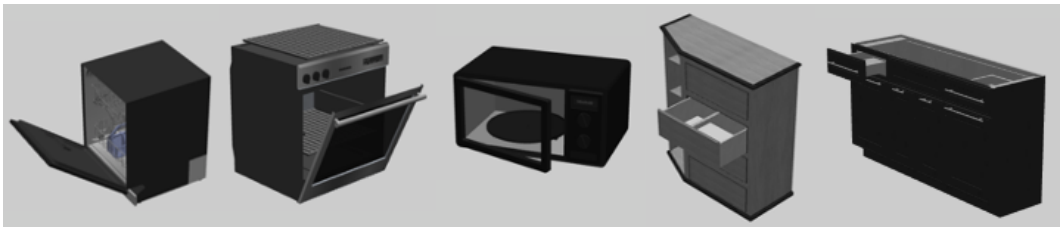
[35] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

A Experimental details

A.1 Dataset

Objects used in the experiments from each of the dataset are shown in the Figures 9a and 9b. We sampled a new object geometry and a joint location for each training example in the simulated articulated object dataset, as proposed by [12]. For the PartNet-Mobility dataset, we considered 11 microwave (8 train, 3 test), 36 dishwasher (27 train, 9 test), 9 oven (6 train, 3 test), 26 single column drawer (20 train, 6 test), and 14 multi-column drawer (10 train, 4 test) object models. For both datasets, we sampled object positions and orientations uniformly in the view frustum of the camera up to a maximum depth dependent upon the object size.

(a) Object classes used from the simulated articulated object dataset [12]. Object classes: cabinet, drawer, microwave, and toaster (left to right)



(b) Object classes used from the PartNet-Mobility dataset [14–16]. Object classes: dishwasher, oven, microwave, drawer- 1 column, and drawer- multiple columns (left to right)

A.2 Experiment 1: Same object class

Numerical error values for the first set of experiments for the simulated articulated objects dataset are presented in the Table 1, and for the PartNet-Mobility dataset are shown in the Table 2.

	Axis Orientation (deg)	Axis displacement (cm)	Configuration
Cabinet - Baseline	$\mathbf{0.082 \pm 0.000}$	8.472 ± 6.277	16.921 ± 8.212 deg
Cabinet - Ours	5.667 ± 13.888	$\mathbf{1.236 \pm 0.66}$	$\mathbf{1.083 \pm 0.707}$ deg
Drawer - Baseline	$\mathbf{0.082 \pm 0.000}$	4.067 ± 1.483	7.389 ± 2.439 cm
Drawer - Ours	0.252 ± 0.000	$\mathbf{1.114 \pm 0.035}$	$\mathbf{1.517 \pm 0.016}$ cm
Microwave - Baseline	$\mathbf{0.083 \pm 0.000}$	3.441 ± 1.218	21.670 ± 7.097 deg
Microwave - Ours	0.304 ± 0.000	$\mathbf{2.322 \pm 1.323}$	$\mathbf{0.329 \pm 0.005}$ deg
Toaster - Baseline	$\mathbf{0.082 \pm 0.000}$	2.669 ± 1.338	11.902 ± 4.242 deg
Toaster - Ours	0.114 ± 0.000	$\mathbf{1.566 \pm 0.018}$	$\mathbf{0.314 \pm 0.018}$ deg

Table 1: Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test object instances for each object class from the simulated articulated objects dataset [12]. Lowest error values for a particular test object set are reported in bold.

	Axis Orientation (deg)	Axis displacement (cm)	Configuration
Dishwasher - Baseline	$\mathbf{0.082 \pm 0.000}$	46.267 ± 20.247	9.735 ± 4.6 deg
Dishwasher - Ours	0.918 ± 0.000	$\mathbf{6.136 \pm 5.455}$	$\mathbf{4.037 \pm 1.613}$ deg
Oven - Baseline	$\mathbf{0.082 \pm 0.000}$	81.444 ± 27.083	13.087 ± 4.649 deg
Oven - Ours	0.583 ± 0.223	$\mathbf{2.111 \pm 1.910}$	$\mathbf{0.720 \pm 0.140}$ deg
Microwave - Baseline	$\mathbf{0.082 \pm 0.000}$	26.781 ± 10.273	11.856 ± 2.456
Microwave - Ours	0.879 ± 0.063	$\mathbf{4.893 \pm 4.252}$	$\mathbf{2.549 \pm 0.939}$ deg
Drawer- 1 column - Baseline	$\mathbf{0.082 \pm 0.000}$	79.228 ± 13.944	17.524 ± 3.700 cm
Drawer- 1 column - Ours	2.140 ± 0.000	$\mathbf{11.567 \pm 9.748}$	$\mathbf{3.181 \pm 0.793}$ cm
Drawer- Multi. cols. - Baseline	$\mathbf{0.082 \pm 0.000}$	63.064 ± 18.913	4.483 ± 6.403 cm
Drawer- Multi. cols. - Ours	1.287 ± 0.000	$\mathbf{12.557 \pm 8.317}$	$\mathbf{4.419 \pm 2.891}$ cm

Table 2: Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test cases for each object class from the PartNet-Mobility Dataset

A.3 Experiment 2: Same articulation model category

Numerical results for the second set of experiments are shown in the Table 3.

	Axis Orientation (deg)	Axis displacement (cm)	Configuration
Oven - Baseline	$\mathbf{0.082 \pm 0.000}$	44.699 ± 12.259	9.915 ± 3.934 deg
Oven - Ours	0.918 ± 0.000	$\mathbf{7.486 \pm 1.273}$	$\mathbf{8.650 \pm 0.207}$ deg
Drawer- 1 column - Baseline	$\mathbf{0.082 \pm 0.000}$	50.990 ± 25.984	5.283 ± 8.862 cm
Drawer- 1 column - Ours	1.287 ± 0.000	$\mathbf{14.548 \pm 5.823}$	$\mathbf{4.399 \pm 0.654}$ cm

Table 3: Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test objects belonging to each object classes from the PartNet-Mobility Dataset

A.4 Ablation studies

We consider three ablated versions of ScrewNet. First, to test the effectiveness of the proposed loss function, we consider an ablated version of ScrewNet which is trained using a raw L2-loss between the labels and the network predictions (named as L2-Error version while reporting results). As the second ablation study, we test whether using an LSTM layer in the network helps with the performance or not (named as NoLSTM version while reporting results). We replace the LSTM layer of the ScrewNet with a fully connected layer such that the two networks, ScrewNet and its ablated version, have a comparable number of parameters. Lastly, to check if a sequence of images is helpful in the model estimation or not, we consider an ablated version of ScrewNet that estimates the articulation model using just a pair of images (named as 2_imgs version while reporting results). Note that ScrewNet and all its ablated versions use a single network each. Numerical results for

the simulated articulated objects dataset are presented in the Table 4, and for the PartNet-Mobility dataset are shown in the Table 5.

	Axis Orientation (deg)	**Axis displacement (cm)**	**Configuration**
Cabinet - Baseline*	$\mathbf{0.082 \pm 0.000}$	8.472 ± 6.277	16.921 ± 8.212 deg
Cabinet - NoLSTM	8.688 ± 20.504	$\mathbf{2.521 \pm 4.341}$	$\mathbf{1.984 \pm 5.172}$ deg
Cabinet - L2-Error	90.186 ± 12.244	5.580 ± 5.138	3.847 ± 5.377 deg
Cabinet - 2_imgs	18.716 ± 40.197	3.188 ± 5.795	12.898 ± 8.846 deg
Cabinet - Ours	16.988 ± 14.971	5.479 ± 4.363	4.65 ± 5.904 deg
Drawer - Baseline*	$\mathbf{0.082 \pm 0.000}$	4.067 ± 1.483	7.389 ± 2.439 cm
Drawer - NoLSTM	14.957 ± 25.526	2.116 ± 2.287	3.878 ± 2.883 cm
Drawer - L2-Error	7.931 ± 13.745	11.141 ± 3.159	5.847 ± 1.468 cm
Drawer - 2_imgs	23.310 ± 27.888	5.118 ± 2.829	7.664 ± 4.883 cm
Drawer - Ours	3.473 ± 8.839	$\mathbf{1.302 \pm 0.999}$	$\mathbf{2.448 \pm 1.092}$ cm
Microwave - Baseline*	$\mathbf{0.082 \pm 0.000}$	3.441 ± 1.218	21.67 ± 7.097 deg
Microwave - NoLSTM	2.725 ± 8.813	$\mathbf{2.439 \pm 1.708}$	$\mathbf{0.803 \pm 2.519}$ deg
Microwave - L2-Error	10.125 ± 10.953	3.76 ± 3.021	4.957 ± 4.489 deg
Microwave - 2_imgs	2.547 ± 3.480	5.115 ± 5.076	18.269 ± 12.658 deg
Microwave - Ours	8.770 ± 13.363	3.398 ± 2.675	4.033 ± 5.998 deg
Toaster - Baseline*	$\mathbf{0.082 \pm 0.000}$	2.669 ± 1.338	11.902 ± 4.242 deg
Toaster - NoLSTM	7.410 ± 17.645	$\mathbf{2.597 \pm 1.86}$	$\mathbf{1.030 \pm 2.230}$ deg
Toaster - L2-Error	18.750 ± 17.243	9.173 ± 4.229	2.661 ± 2.823 deg
Toaster - 2_imgs	12.833 ± 22.596	4.123 ± 3.196	16.016 ± 10.703 deg
Toaster - Ours	11.583 ± 14.798	3.003 ± 1.75	3.471 ± 2.876 deg

Table 4: Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test objects belonging to each object classes from the simulated articulated objects dataset. Asterisk $(*)$ denote that the baseline has a significant advantage over other methods as it uses a separate network for each object class, while all ScrewNet and its ablations use a single network

	Axis Orientation (deg)	Axis displacement (cm)	Configuration
Dishwasher - Baseline*	$\mathbf{0.082 \pm 0.000}$	46.267 ± 20.247	9.735 ± 4.600 deg
Dishwasher - NoLSTM	41.485 ± 41.184	9.815 ± 6.782	$\mathbf{5.415 \pm 4.097}$ deg
Dishwasher - L2 Error	25.405 ± 15.119	12.653 ± 8.119	7.828 ± 1.913 deg
Dishwasher - 2_imgs	1.935 ± 0.021	11.544 ± 4.729	5.706 ± 4.152 deg
Dishwasher - Ours	11.850 ± 15.267	$\mathbf{6.789 \pm 5.630}$	6.081 ± 3.043 deg
Oven - Baseline*	$\mathbf{0.082 \pm 0.000}$	81.429 ± 27.244	13.026 ± 4.670 deg
Oven - NoLSTM	29.968 ± 39.034	11.014 ± 13.235	10.574 ± 6.332 deg
Oven - L2 Error	27.197 ± 13.103	26.452 ± 14.704	11.823 ± 1.067 deg
Oven - 2_imgs	1.939 ± 0.018	$\mathbf{4.791 \pm 1.370}$	10.498 ± 7.481 deg
Oven - Ours	7.881 ± 7.763	6.786 ± 2.443	$\mathbf{5.010 \pm 1.233}$ deg
Microwave - Baseline*	$\mathbf{0.082 \pm 0.000}$	26.781 ± 10.273	11.856 ± 2.456 deg
Microwave - NoLSTM	40.911 ± 32.830	15.993 ± 14.080	3.865 ± 2.350 deg
Microwave - L2 Error	60.566 ± 7.705	59.286 ± 6.485	7.463 ± 1.612 deg
Microwave - 2_imgs	91.826 ± 0.012	$\mathbf{11.994 \pm 2.549}$	5.212 ± 3.606 deg
Microwave - Ours	24.959 ± 24.847	15.271 ± 13.561	$\mathbf{3.507 \pm 1.987}$ deg
Drawer- 1 col. - Baseline*	$\mathbf{0.082 \pm 0.000}$	79.228 ± 13.944	17.524 ± 3.700 cm
Drawer- 1 col. - NoLSTM	42.318 ± 35.604	47.991 ± 29.586	10.923 ± 6.449 cm
Drawer- 1 col. - L2 Error	48.136 ± 9.533	60.046 ± 19.375	14.202 ± 2.153 cm
Drawer - 1 col. - 2_imgs	89.372 ± 0.047	80.356 ± 8.087	25.753 ± 18.374 cm
Drawer- 1 col. - Ours	19.876 ± 21.684	$\mathbf{28.329 \pm 15.005}$	$\mathbf{5.729 \pm 4.259}$ cm
Drawer- Multi. cols. - Baseline*	$\mathbf{0.082 \pm 0.000}$	63.064 ± 18.913	4.483 ± 6.403 cm
Drawer- Multi. cols.- NoLSTM	38.393 ± 33.113	49.419 ± 23.998	6.181 ± 5.228 cm
Drawer- Multi. cols.- L2 Error	38.866 ± 5.243	44.422 ± 26.927	6.422 ± 0.766 cm
Drawer- Multi. cols. - 2_imgs	89.361 ± 0.053	78.131 ± 4.888	12.229 ± 3.961 cm
Drawer- Multi. cols. - Ours	9.292 ± 15.295	$\mathbf{17.813 \pm 14.719}$	$\mathbf{0.915 \pm 1.772}$ cm

Table 5: [Experiment: Across articulation model category] Mean error values for joint axis orientation, joint axis position, and configurations for 1000 test objects belonging to each object classes from the PartNet-Mobility Dataset. Asterisk $(*)$ denote that the baseline has a significant advantage over other methods as it uses a separate network for each object class