Learning Part Motion of Articulated Objects Using Spatially Continuous Neural Implicit Representations

Yushi Du*
1900012147@pku.edu.cn
Ruihai Wu*
wuruihai@pku.edu.cn
Yan Shen
yan790@pku.edu.cn
Hao Dong†
hao.dong@pku.edu.cn

Center on Frontiers of Computing Studies
School of Computer Science
Peking University

* denotes equal contribution
† denotes corresponding author

Abstract

Articulated objects (*e.g.*, doors and drawers) exist everywhere in our life. Different from rigid objects, articulated objects have higher degrees of freedom and are rich in geometries, semantics, and part functions. Modeling different kinds of parts and articulations with nerual networks plays an essential role in articulated object understanding and manipulation, and will further benefit 3D vision and robotics communities. To model articulated objects, most previous works directly encode articulated objects into feature representations, without specific designs for parts, articulations and part motions. In this paper, we introduce a novel framework that explicitly disentangles the part motion of articulated objects by predicting the transformation matrix of points on the part surface, using spatially continuous neural implicit representations to model the part motion smoothly in the space. More importantly, while many methods could only model a certain kind of joint motion (such as the revolution in the clockwise order), our proposed framework is generic to different kinds of joint motions in that transformation matrix can model diverse kinds of joint motions in the space. Quantitative and qualitative results of experiments over diverse categories of articulated objects demonstrate the effectiveness of our proposed framework.

1 Introduction

There are a plethora of 3D objects around us in the real world. Compared to those rigid objects with only 6 degrees of freedom (DoF), articulated objects (*e.g.*, doors and drawers) additionally contain semantically and functionally important articulated parts (*e.g.*, the screen of laptops), resulting in their higher DoFs in state space, and more complicated geometries and functions. Therefore, understanding and representing articulated objects with diverse geometries and functions is an essential but challenging task for 3D computer vision.

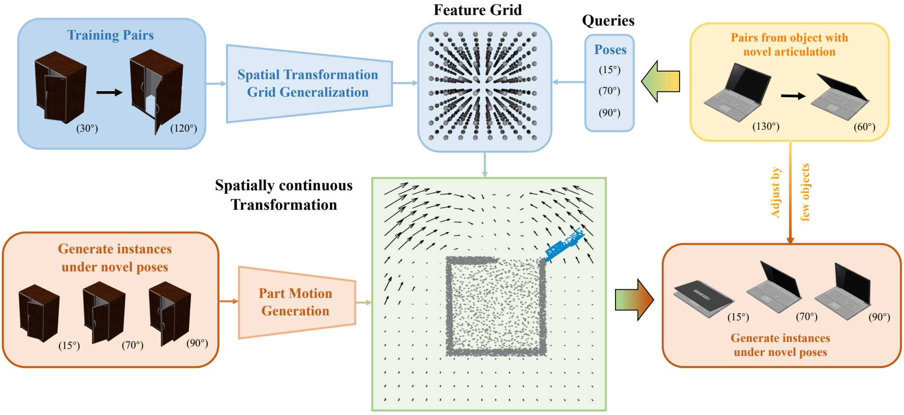
Training Pairs
(30°)
(120°)
Spatial Transformation Grid Generalization
Feature Grid
Queries
Poses
(15°)
(70°)
(90°)
Pairs from object with novel articulation
(130°)
(60°)
Adjust by few objects
Spatially continuous Transformation
Generate instances under novel poses
(15°)
(70°)
(90°)
Part Motion Generation
(15°)
(70°)
(90°)
Generate instances under novel poses

Figure 1: We propose **spatially continuous neural implicit grid** that receives two point clouds of the same object under different part poses. The point clouds are provided with their corresponding articulated part poses and the grid could encode two frames of point clouds into a spatially continuous implicit feature grid with both geometric and pose information. By taking different new part poses as queries, we decode per-point transformations representing articulated part motions from the feature grid. Then we move the object using the transformation to generate objects under novel poses. This representation could be easily adjusted to articulated objects with novel shapes and joint motions (*e.g.*, from door to laptop) tuned on only a few new objects.

Many studies have been investigating the perception of 3D articulated objects, including discovering articulated parts [7, 10, 28], inferring kinematic models [1, 16], estimating joint configurations [5, 12, 18, 19], predicting part poses [36, 41], building digital twins [14] and manipulating parts [40]. One recent work, A-SDF [27], studies the representations of articulated objects by encoding shape and articulation into latent space. But instead of considering modeling articulation objects as linking parts under motion constraints, they directly decode the whole object point cloud into the latent space. Another work, Ditto [14], successfully generates objects under novel poses over diverse joint motions (*e.g.*, rotation and displacement over different axis) using a single network. However, this method relies on specific articulation annotations such as joint type, orientation, and displacement which limits their ability to generalise across diverse articulations (*e.g.*, different joint motion and type).

In this paper, we introduce a novel framework for learning a spatial continuous representation of the part motion of articulated objects, and enable the few-shot generalisation across different novel object categories with different joint motion. To be specific, we model articulation as a constraint that can map a scalar value representing the part poses to a transformation describing the movements of the articulated parts.

To further study the representations of articulated objects, with a focus on the objects' parts, we introduce our novel framework for learning the part motions of articulated objects. To be specific, we model the movement of parts as a mapping between a scalar representing the part pose and a transformation matrix. For a reason that part motion is a core and generic property shared by all articulated objects, our proposed framework is generic to various articulated objects with diverse kinds of part motions, without any need to have specific designs for each kind of object.

Considering the limited number of DoF of joints on articulated objects, the motions of points on the articulated part should make up a continuous and smooth distribution with respect to points' positions on parts. In other words, close points on the part surface have similar motions, while far away points have varied motions. Therefore, we further propose to use spatially continuous neural implicit representations for the representations of point motions on the articulated part. Inspired by ConvONet [31], we build a fine-grained and spatially continuous implicit grid for learning the representations of point-level transformations from one pose to another.

We conduct experiments over large-scale PartNet-Mobility dataset [2, 25, 39], covering 3D articulated objects with diverse geometries over 7 object categories. Quantitative and qualitative results demonstrate that using the spatially continuous grid, our method accurately and smoothly models part motion and generates articulated objects with novel part poses reserving detailed geometries, showing our superiority over baseline methods.

2 Related Work

2.1 Representing Articulated Objects

How to understand and to model articulated objects has been a long-lasting research topic, including segmenting articulated parts [11, 15, 16, 20], tracking feature trajectories [5, 9, 16], estimating joint configurations [9, 15, 17, 20, 21], and modelling kinematic structures [20, 21, 23, 35]. Recently, many works [1, 10, 12, 18, 19, 36, 41, 42] further utilise the deep learning methods to study diverse articulated objects, leading to better performance and stronger generalisation. A recent work A-SDF [27] studies the problem of generic articulated object synthesis and leverages implicit functions to decode articulated objects into latent codes. However, most of these works represent articulated objects by abstracting standardised kinematic structure, estimating joint parameters, and predicting part pose, which may not provide explicit information on articulated shapes for downstream tasks like robotics manipulation [6, 7, 26, 37, 38]. Different from those works, we utilise neural implicit functions for explicit articulated object representation and generation.

2.2 Neural Implicit Representation

A vast and impressive literature has investigated neural implicit representations [3, 8, 13, 22, 24, 29, 30, 31, 34], which utilises deep neural networks to implicitly encode 3D shapes into continuous and differential signals in high resolution. While most of the previous works study the representation of 3D rigid objects, two recent works, A-SDF [27] and Ditto [14], focus on the representation of 3D articulated objects. A-SDF [27] represents the articulated objects by separately encoding shape and articulation into latent space. Ditto [14] builds digital twins of articulated objects by reconstructing the part-level geometry and estimating the articulations explicitly. However, both of the works represent articulated objects without considering the integrity of the articulated parts, which is a generic property shared by all articulated objects. In this work, we utilise this property and leverage spatially continuous neural implicit representation to model the motion of the monolithic articulated parts.

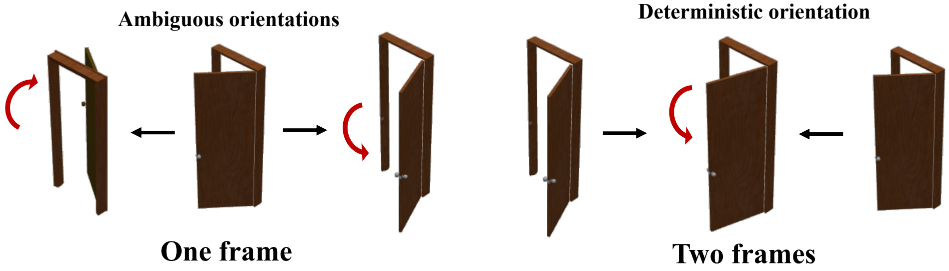

Figure 2: **Two point cloud frames are required** for learning articulated part motions, as one frame may indicate ambiguous motions (*e.g.,* clockwise and anti-clockwise orientations).

3 Problem Formulation

Learning the motion of the articulated part on an object requires at least two frames of that object under different poses (*e.g.,* different door opening degree). That is because using one frame as the observation would have an ambiguity problem, take Figure 2 as an example, given an observation of a door, we cannot distinguish whether the revolute direction is clockwise or anti-clockwise.

In this study, each object in training set provides two point cloud $I_1, I_2 \in \mathbb{R}^{N\times 3}$ under different part poses. The model maps part motion to corresponding part pose scalar values $\phi_1, \phi_2 \in \mathbb{R}$ representing the degree of articulation, and can 1) generate new point cloud I_3 given a new part pose scalar $\phi_3 \in \mathbb{R}$. 2) can few-shot generalise to novel object categories.

4 Method

As shown in Figure 3, our proposed framework is mainly composed of two procedures, **Spatial Transformation Grid Generation** (Left) and **Part Motion Generation** (Right).

Spatial Transformation Grid Generation: As is described in 1. The distribution of the movements of points on the articulated part possesses spatial continuity over the 3D space. In this section, our framework receives a pair of articulated object point clouds with their corresponding part poses $((I_1,\ \phi_1),(I_2,\ \phi_2))$, as well as a new part pose ϕ_3 as input. Then output a Spatial Transformation Feature Grid G to extract such spatial continuous features representing the part motions of articulated objects.

Part Motion Generation: In order to generate the object under pose ϕ_3, we decode the transformation matrices from ϕ_1 to ϕ_3 of each point from the Spatial Transformation Feature Grid G. Firstly, with respect to a novel part pose ϕ_3, our framework retrieves each point p in I_1's transformation representation $\psi_{\phi_3} \in \mathbb{R}^{N\times d_\psi}$ in Spatial Transformation Grid G using trilinear interpolation. Then we decode each point's transformation representation into a transformation matrix t_p, and thus all the points' transformation matrix t_p compose the whole transformation matrix T_{ϕ_3} for the whole point cloud I_1. Finally, we apply the transformation matrices T_{ϕ_3} to I_1 and get the point cloud $\hat{I}_3$ under the articulated part pose ϕ_3. In the following sections, we show details of our proposed framework.

4.1 Spatial Transformation Grid Generation

In this procedure, we generate the Spatial Transformation Feature Grid G to extract the spatial distribution of joint motion features.

As mentioned in Section 1, the point motions on the articulated part surface make up a continuous and smooth distribution with respect to point positions. Therefore, spatially continuous neural implicit representations are suitable for the representations of point motions.

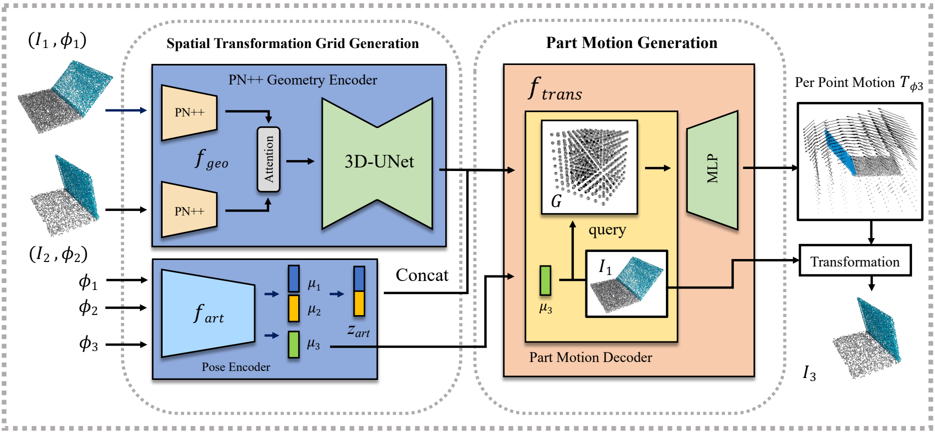

Figure 3: **Our proposed framework** receives two point clouds I_1 and I_2 from the same articulated object under two different part poses ϕ_1 and ϕ_2. Then generate the object point cloud I_3 with a new part pose ϕ_3. It aggregates the geometric information of I_1 and I_2, and the pose information of ϕ_1 and ϕ_2 into a spatially continuous Transformation Grid. During inferencing, conditioned on the new part pose ϕ_3, it decodes the transformation of each point by querying each point in the Grid to generate the input object with the novel pose.

We build such a 3D grid with $K \times K \times K$ points uniformly distributed in space ($K = 32$), each point having implicit features representing both the object geometries and part motions.

To empower the learned Grid with both object geometries and part motions, the **Spatial Transformation Grid Generation** procedure consists of two submodules: 1) *Geometry Encoder* f_{geo} that takes two point clouds under different part poses (which is I_1 and I_2) as input and outputs an implicit feature grid G_{geo}; 2) *Pose Encoder* f_{art} that takes part poses ϕ_1, ϕ_2, ϕ_3 respectively as input and outputs their respective features $\mu_1, \mu_2, \mu_3 \in \mathbb{R}^{d_\mu}$, and then concatenates μ_1, μ_2 into z_{art} while passing down μ_3 for further use. Finally, we concatenate z_{art} to each grid feature of G_{geo} to form Transformation Feature Grid G:

$$G_{geo} = f_{geo}(I_1,\ I_2), z_{art} = f_{art}(\phi_1,\ \phi_2), G = [G_{geo},\ z_{art}]$$

4.1.1 Geometry Encoder

Inspired by Ditto [14], to extract geometric information of the two input point clouds, we first use PointNet++ [32] encoders to encode I_1 and I_2, and extract sub-sampled point cloud features $h_1, h_2 \in \mathbb{R}^{N' \times d}$, where N' denotes the point number after the sub-sampling procedure of PointNet++, and we use $N' = 128$ in our work.

To aggregate the features of the two input point clouds, we employ an attention module between sub-sampled point features h_1, h_2 into an aggregated feature h:

$$h = [h_1, softmax(\frac{h_1 h_2^T}{\sqrt{d}})h_2]$$

Then, we feed the aggregated feature h into a 3D-UNets [4]and generate 3D geometric implicit feature grid G_{geo} representing geometric information of the two input point clouds with $K \times K \times K$ uniformly distributed points.

4.1.2 Pose Encoder

We use Multi-Layer Perceptrons (MLP) to separately encode part poses ϕ_1, ϕ_2, ϕ_3 into articulation features μ_1, μ_2, μ_3 $\in \mathbb{R}^{N \times d_{art}}$, and concatenate μ_1 and μ_2 to form $z_{art} = [\mu_1, \mu_2]$.

We again concatenate z_{art} with each point feature in G_{geo} to form Spatial Transformation Feature Grid G, containing spatially continuous implicit features about both the geometric information and the pose information of the target object in the space.

4.2 Part Motion Generation

During the above **Spatial Transformation Grid Generation** procedure, we have generated the Spatial Transformation Feature Grid G. In this **Part Motion Generation** procedure, we use G to generate spatially continuously distributed point motions from I_1 to the target I_3.

Firstly, from G which is composed of $K \times K \times K$ points uniformly distributed in the space with their corresponding features, we query the feature f_p under μ_3 of each point p on the articulated part using trilinear interpolation.

Then, we employ a motion decoder f_{trans} (composed of an MLP network) to decode the transformation matrix t_p from ϕ_1 to ϕ_3 of each point p on the articulated part. Taking pose feature μ_3 as conditions, our decoder obtain the corresponding t_p and conduct elemental-wise production to generate the point cloud prediction $\hat{I_3}$ under the part pose ϕ_3.

$$\psi_{\phi_3} = Query(I_1, G), T_{\phi_3} = f_{trans}(\psi_{\phi_3}), \hat{I_3} = T_{\phi_3} \cdot I_1$$

In this way, for those points on the articulated parts, their motions could be generated smoothly from the spatially continuous distribution, keeping the part as a whole after the motion, while maintaining the geometric details of them.

4.3 Training and Loss

Data collection. To generate diverse data for training, we randomly sample articulated part poses ϕ_1, ϕ_2 and ϕ_3 and then generate point cloud observation I_1, I_2 and I_3 corresponding to each part poses. Ascribing to the ability to get point could in simulator with arbitrary part poses, we can generate diverse $((I_1, \phi_1), (I_2, \phi_2), \phi_3)$ for training.

Loss function. We use Earth Mover's Distance (EMD) [33] as the loss function. EMD is utilised to estimate the distance between two distributions. We can calculate the EMD between two point clouds by calculating the minimum amount of point movements needed to change the generated object point cloud into the target. In our work, with the input data $((I_1, \phi_1), (I_2, \phi_2), \phi_3)$, the EMD is computed between the ground truth point cloud I_3 of the articulated object with the part pose ϕ_3, and our prediction $\hat{I_3}$.

We set up a loss optimising whole point cloud and increase the weight of loss on movable part to facilitate neat part formulation with smooth surfaces and fewer outliers.

$$Loss = EMD(I_3, \hat{I_3})$$

5 Experiments

We conduct our experiments using the large-scale PartNet-Mobility [1, 25, 39] dataset of 3D articulated objects, covering over 7 object categories. We evaluate the performance of our method in several tasks including: 1) the articulated object generation for unseen objects in training categories, 2) few-shot articulated object generation for novel object categories, and 3) the interpolation and extrapolation of the spatial continuous NIR. Quantitative and qualitative results compared to several baselines and an ablated version demonstrate our method's superiority over other methods.

5.1 Baselines and Metrics

We evaluate and compare our approach with the following two baselines and one ablation:

A-SDF [27] represents objects with a shape code and an articulation code. Given an object, it first infers the shape and articulation codes and then generates the shape at unseen angles by keeping the shape code unchanged and changing the articulation code.

Ditto [14] also takes two point clouds as input to learn the structure of an articulated object. It directly predicts the occupancy, the segmentation, and the joint configuration to build a digital twin. The original paper demonstrates the point cloud reconstruct ability, we modify it to take a new part pose as input and then generate the corresponding object.

Ours w/o NIR is an ablated version of our method that directly predicts the transformation matrix for each point to generate the new point cloud without applying spatially continuous NIR as a middle step. We conduct this ablation version to demonstrate the effectiveness of our design using Spatial Transformation Feature Grid G.

To evaluate the generated objects and their similarity with the ground-truth objects, we apply the Earth Mover's Distance (EMD) [33] as the evaluation metric.

5.2 Evaluation on Unseen Objects in Training Categories

Method	Laptop	Stapler	Door	Scissors	Oven	Refrigerator	Microwave	Table
A-SDF	1.6923	3.9335	3.2459	1.9307	1.3983	2.2532	3.7570	1.8467
Ditto	1.6195	3.1161	2.9811	2.1619	1.3401	1.9863	4.8210	1.4010
Ours w/o NIR	1.6080	3.3369	2.5863	2.0628	1.1294	2.1539	1.9281	1.5189
Ours	**1.4420**	**3.0850**	**2.2808**	**1.8025**	**1.1134**	**1.6431**	**1.8088**	**1.3315**

Table 1: **Earth Mover's Distance (EMD) on object generation in training categories.**

In this task, given an articulated object in the training category with two point clouds and the corresponding part poses, we generate its point cloud with novel part poses.

The quantitative results in Table 1 demonstrate that our proposed framework outperforms all other methods in all categories with lower EMD, which means that our generated articulated objects are the closest to the ground-truth shapes. The qualitative results in Figure 4 also show that our generated objects reserve the most detailed geometry. In comparison, the performance of both Ditto and A-SDF is worse, for example, they both fail to predict the door frame straightly, and fail to predict the microwave door surface smoothly.

The main reason for the difference is, A-SDF and Ditto directly decode the whole point cloud into latent space, while ours takes the integrity of parts into consideration by querying the motion of each point in the original point cloud. This one-to-one mapping from the original shape to the generated shape best preserves geometric features of the original shape.

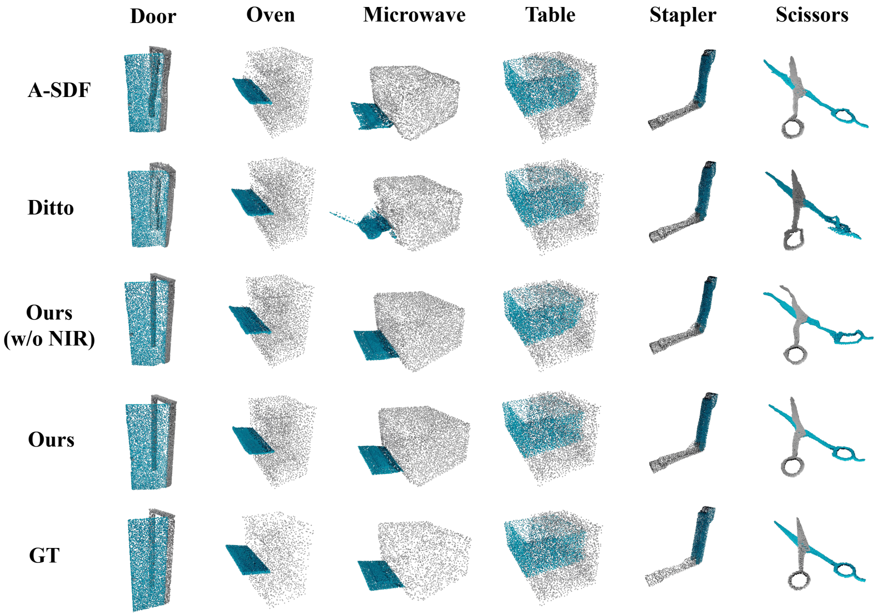

Figure 4: **Visualisation of generated objects in training categories** shows our method reserves the most detailed geometries of both articulated parts and object bases. For example, our model predicts the straightest door frame and the smoothest microwave door surface.

5.3 Evaluation on Novel Categories

In this task, we use the pretrained model in one category and finetune the model in a novel category using only a few objects for a few epochs. Specifically, we use 8 objects in the novel category, and the finetuning time is one-twentieth of the training time from scratch. It is worth mentioning that the directions of the articulated part axes in the training set and finetuning set are different in these experiments (*i.e.,* we train on the up-down opening ovens and finetune on the left-right opening refrigerators.) This task aims to demonstrate that learning the part motions of articulated objects makes the model easier to adjust to a novel kind of articulated object, as it is the shared property of all articulated objects.

The quantitative results in Table 2 show that our method achieves significantly better results with lower EMD compared to all the baselines, especially in the Oven-Refrigerator block. The visualisation results of Figure 5 also show that our method present the most accurate part poses and the most precise part geometry after a short-period finetuning.

Failures of A-SDF possibly come from that, the representations learned by A-SDF are limited to the trained articulated object category and are hard to adjust to novel shapes and articulations.

Method	Oven-Refri	Refri-Oven	Door-Laptop
A-SDF	37.7618	2.0164	2.2532
Ditto	3.9443	2.1832	2.3997
Ours w/o NIR	15.5203	1.6832	2.3547
Ours	**2.5794**	**1.5440**	**2.0873**

Table 2: **Earth Mover's Distance (EMD) on object generation in novel categories.**

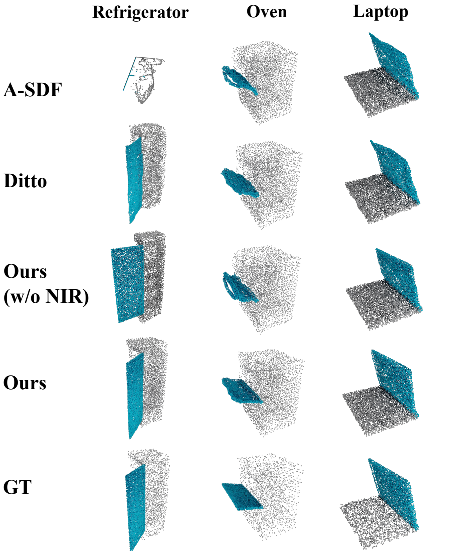

Figure 5: **Visualisation of generated objects in novel categories** shows our method maintains geometric consistency.

We have also conducted experiments using the widely-used metric Pose Angle Error (PAE) and Chamfer Distance (CD), with results shown in Table 3.

Our superior performance in novel categories against Ditto mainly comes from the use of transformation matrix to represent part motion. Intuitively, a transformation matrix could represent any kind of motion in 3D space and is spatial continuous for points on the motion part. As a result, it has the potential to few-shot generalise to any kind of part motion no matter its displacement.

5.4 Ablation Studies and Analysis

We compare our method with the ablated version without Neural Implicit Representations (**Ours w/o NIR**). Results in Table 1 and Table 2 show that NIR helps the generated point cloud to be closer to the ground-truth target, representing by the lower EMD between the generated objects and the ground-truth objects. From the visualisation in Figure 4 and Figure 5, we can observe that the point clouds generated with NIR have more accurate part pose and smoother part surface. Those results demonstrate that by using Spatially Continuous Neural Implicit Representation to model the part motion, our framework gets a better distribution for motion representations in the 3D space.

5.5 Analysis of Transformation on Grid Points

Figure 6 visualises the transformation grid of a refrigerator instance (in the first row) and an oven instance (in the second row). The figures on the left are displayed in 3D while the right ones are displayed in 2D. Note that for better visualisation and understanding, on the right, we represent the refrigerator in the top-down view, and represent the oven in the side view. The arrows forging circles centreing the ground-truth joint show that our model successfully projects the part motion to euclidean space.

Metric	A-SDF	Ditto	Ours
CD ↓	2.213	2.019	**1.782**
PAE (degree) ↓	6.457	6.212	**4.767**

Table 3: **Evaluations on CD and PAE.**

5.6 Interpolation and Extrapolation

Interpolation and extrapolation between shapes is a key ability for 3D object representations which reveals the distribution of articulation part poses.

In this task, given two shapes of the same object, we generate the object with novel articulation degrees in between or beyond. In Table 4, quantitative results show that our method outperforms A-SDF and Ditto in both interpolation and extrapolation tasks. In Figure 6,

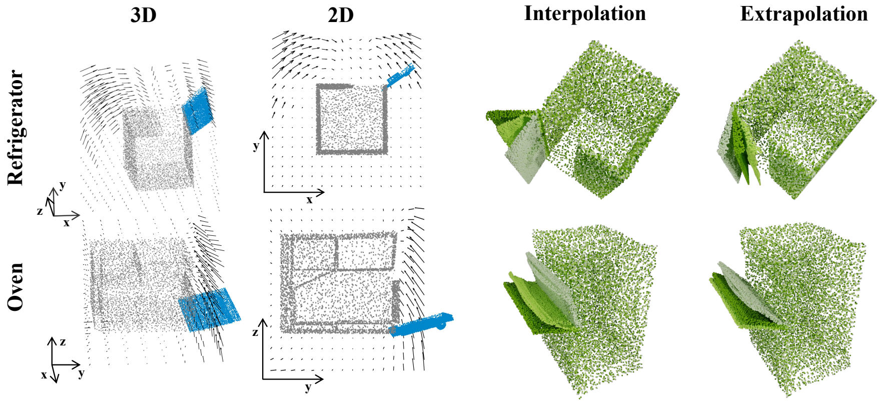

Figure 6: **Visualisation of the transformation on grid points (left), and results of interpolation and extrapolation (right).**

Method	Fridge	Oven	Door	Table	Fridge	Oven	Door	Table
A-SDF	4.1248	1.6185	2.8883	8.3870	4.2671	2.5514	5.3937	8.0931
Ditto	3.2421	1.2861	2.5974	9.8180	3.1738	1.9405	4.4676	8.5025
Ours	**2.1256**	**1.2364**	**2.1330**	**8.1688**	**2.8376**	**1.7669**	**3.5298**	**7.0804**

Table 4: **EMD on interpolation (Left) and extrapolation (Right) results.**

we represent the input parts with dark and light green, and the generated part with medium green. The results demonstrate our representation of part motion is continuous and dense.

5.7 Multi-part Generation

Our method can easily extend to objects with multiple parts by changing the input part angle to a vector of part angles, shown in Figure 7.

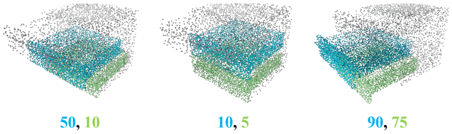

Figure 7: **Multi-part object generation.**

6 Conclusion

In this paper, we propose a novel framework for modelling and generating articulated objects. To model the continuous articulations and motions smoothly, we adopt neural implicit representations (NIR) to predict the transformations of moving part points of the object. Experiments on different representative tasks demonstrate that our proposed framework outperforms other methods both quantitatively and qualitatively.

7 Acknowledgment

This work was supported by National Natural Science Foundation of China (No. 62136001).

References

[1] Ben Abbatematteo, Stefanie Tellex, and George Konidaris. Learning to generalize kinematic models to novel objects. In *Proceedings of the 3rd Conference on Robot Learning*, 2019.

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

[3] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019.

[4] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *International conference on medical image computing and computer-assisted intervention*, pages 424–432. Springer, 2016.

[5] Karthik Desingh, Shiyang Lu, Anthony Opipari, and Odest Chadwicke 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.

[6] Ben Eisner, Harry Zhang, and David Held. Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. *arXiv preprint arXiv:2205.04382*, 2022.

[7] Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Act the part: Learning interaction strategies for articulated object part discovery. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15752–15761, 2021.

[8] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020.

[9] Karol Hausman, Scott Niekum, Sarah Osentoski, and Gaurav S Sukhatme. Active articulation model estimation through interactive perception. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3305–3312. IEEE, 2015.

[10] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi-Min Hu, and Leonidas J Guibas. Multibodysync: Multi-body segmentation and motion estimation via 3d scan synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7108–7118, 2021.

[11] Xiaoxia Huang, Ian Walker, and Stan Birchfield. Occlusion-aware reconstruction and manipulation of 3d articulated objects. In *2012 IEEE International Conference on Robotics and Automation*, pages 1365–1371. IEEE, 2012.

[12] Ajinkya Jain, Rudolf Lioutikov, Caleb Chuck, and Scott Niekum. Screwnet: Category-independent articulation model estimation from depth images using screw theory. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13670–13677. IEEE, 2021.

[13] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020.

[14] Zhenyu Jiang, Cheng-Chun Hsu, and Yuke Zhu. Ditto: Building digital twins of articulated objects from interaction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5616–5626, 2022.

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

[16] Dov Katz, Moslem Kazemi, J Andrew Bagnell, and Anthony Stentz. Interactive segmentation, tracking, and kinematic modeling of unknown 3d articulated objects. In *ICRA*, pages 5003–5010, 2013.

[17] Dov Katz, Andreas Orthey, and Oliver Brock. Interactive perception of articulated objects. In *Experimental Robotics*, pages 301–315. Springer, 2014.

[18] Xiaolong Li, He Wang, Li Yi, Leonidas J Guibas, A Lynn Abbott, and Shuran Song. Category-level articulated object pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3706–3715, 2020.

[19] Qihao Liu, Weichao Qiu, Weiyao Wang, Gregory D Hager, and Alan L Yuille. Nothing but geometric constraints: A model-free method for articulated object pose estimation. *arXiv preprint arXiv:2012.00088*, 2020.

[20] Roberto Martin Martin and Oliver 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.

[21] Roberto Martín-Martín, Sebastian Höfer, and Oliver 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.

[22] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4460–4470, 2019.

[23] Frank Michel, Alexander Krull, Eric Brachmann, Michael Ying Yang, Stefan Gumhold, and Carsten Rother. Pose estimation of kinematic chain instances via object coordinate regression. In *BMVC*, pages 181–1, 2015.

[24] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021.

[25] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao 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.

[26] Kaichun Mo, Leonidas J Guibas, Mustafa Mukadam, Abhinav Gupta, and Shubham Tulsiani. Where2act: From pixels to actions for articulated 3d objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6813–6823, 2021.

[27] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-sdf: Learning disentangled signed distance functions for articulated shape representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13001–13011, 2021.

[28] Neil Nie, Samir Yitzhak Gadre, Kiana Ehsani, and Shuran Song. Structure from action: Learning interactions for articulated object 3d structure discovery. *arXiv preprint arXiv:2207.08997*, 2022.

[29] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3504–3515, 2020.

[30] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[31] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020.

[32] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017.

[33] Yossi Rubner, Carlo Tomasi, and Leonidas J Guibas. The earth mover's distance as a metric for image retrieval. *International journal of computer vision*, 40(2):99–121, 2000.

[34] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3d-structure-aware neural scene representations. *Advances in Neural Information Processing Systems*, 32, 2019.

[35] Jürgen Sturm, Cyrill Stachniss, and Wolfram Burgard. A probabilistic framework for learning kinematic models of articulated objects. *Journal of Artificial Intelligence Research*, 41:477–526, 2011.

[36] Xiaogang Wang, Bin Zhou, Yahao Shi, Xiaowu Chen, Qinping Zhao, and Kai Xu. Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8876–8884, 2019.

[37] Yian Wang, Ruihai Wu, Kaichun Mo, Jiaqi Ke, Qingnan Fan, Leonidas Guibas, and Hao Dong. AdaAfford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions. 2022.

[38] Ruihai Wu, Yan Zhao, Kaichun Mo, Zizheng Guo, Yian Wang, Tianhao Wu, Qingnan Fan, Xuelin Chen, Leonidas Guibas, and Hao Dong. VAT-mart: Learning visual action trajectory proposals for manipulating 3d ARTiculated objects. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=iEx3PiooLy.

[39] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, Li Yi, Angel X. Chang, Leonidas J. Guibas, and Hao Su. SAPIEN: A simulated part-based interactive environment. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[40] Zhenjia Xu, He Zhanpeng, and Shuran Song. Umpnet: Universal manipulation policy network for articulated objects. *IEEE Robotics and Automation Letters*, 2022.

[41] Zihao Yan, Ruizhen Hu, Xingguang Yan, Luanmin Chen, Oliver Van Kaick, Hao Zhang, and Hui Huang. Rpm-net: recurrent prediction of motion and parts from point cloud. *arXiv preprint arXiv:2006.14865*, 2020.

[42] Vicky Zeng, Tabitha Edith Lee, Jacky Liang, and Oliver Kroemer. Visual identification of articulated object parts. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2443–2450. IEEE, 2021.