Supplementary Material for: *Ditto: Building Digital Twins of Articulated Objects through Interaction*

Zhenyu Jiang Cheng-Chun Hsu Yuke Zhu
Department of Computer Science, The University of Texas at Austin

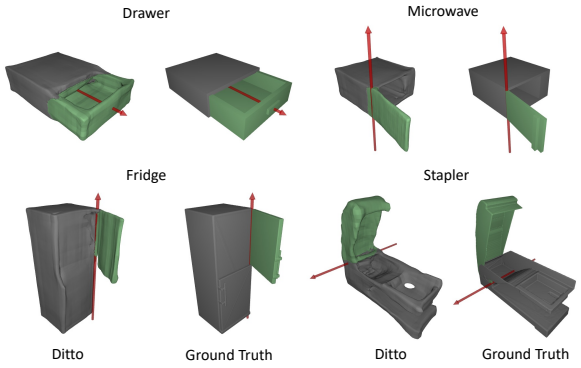

Figure 1. Qualitative results of generalizing to unseen categories.

1. Implementation Details

We use the Shape2Motion dataset [1] and the Synthetic dataset [2]. The Shape2Motion dataset is licensed under the GNU General Public License v3.0. For Shape2Motion dataset, we choose four categories with more than 30 instances. We choose 4 out of 6 categories for the synthetic dataset because the other two are very similar to the chosen ones. We sample 8,192 points for each input point cloud. In each iteration, we sample 2,048 pairs of $\mathbf{p}$ and corresponding occupancy. We also 512 pairs of $\mathbf{p}_{in}$ inside the object and corresponding segmentation and joint parameters as query points and ground truths. $\mathbf{p}$ and $\mathbf{p}_{in}$ are input query points for geometry decoder and articulation decoders separately.

We implement the models with Pytorch [3] and train the models with the Adam [4] optimizer and a learning rate of 10^{-4} and batch sizes of 8.

2. Ablation Study

Ditto uses 3D feature grid for geometry reconstruction and 2D feature plane for articulation estimation. To validate the advantage of this design, we evaluate another ablated version where two 3D feature grid are used for geometry and articulation respectively. As in Tab. 1, this ablated version has similar performance to Ditto. But it requires around 20% more memory usage and training time compared with Ditto.

Method	Whole CD ↓	Mobile CD
A-SDF (oracle code) [5]	0.66	-
Ditto (3D+3D feature)	0.27	0.12
Ditto	0.25	0.16

Table 1. Quantitative results of reconstruction on the Synthetic [2] dataset.

We show some qualitative results in Fig. 2. Our full Ditto model can recreate the articulated objects more accurately, especially the mobile part, benefiting from the attention-based fusion, separate decoders and features. In comparison, Concat Fusion and Share Feature are not able to reconstruct the smooth and complete surface. The ablated version with Share Feature uses 2D feature planes along with 3D feature grids for geometry reconstruction. This projection operation results in artifacts as in the faucet result in Fig. 2 (red circle in the second row, first column). The ablated version with Share Decoder has a problem segmenting the mobile and the static parts correctly. Overall, Ditto can achieve the best performance on the reconstruction of the articulated objects.

3. Comparison with A-SDF

To explore the possible reason behind the inferior performance of A-SDF, we try fixing the A-SDF's articulation code to the ground-truth one. The reconstruction result is improved and close to Ditto as in Tab. 1. It indicates that the inferior performance of A-SDF is caused by the in- terference between articulation and shape codes in test-time optimization. For example, the shape code degrades when the articulation code is in a local minimum far from the ground truth. In contrast, the articulation and geometry predictions do not interfere with each other in our model.

A-SDF [5] can control the joint state by changing the articulation code. On the other hand, Ditto explicitly re-

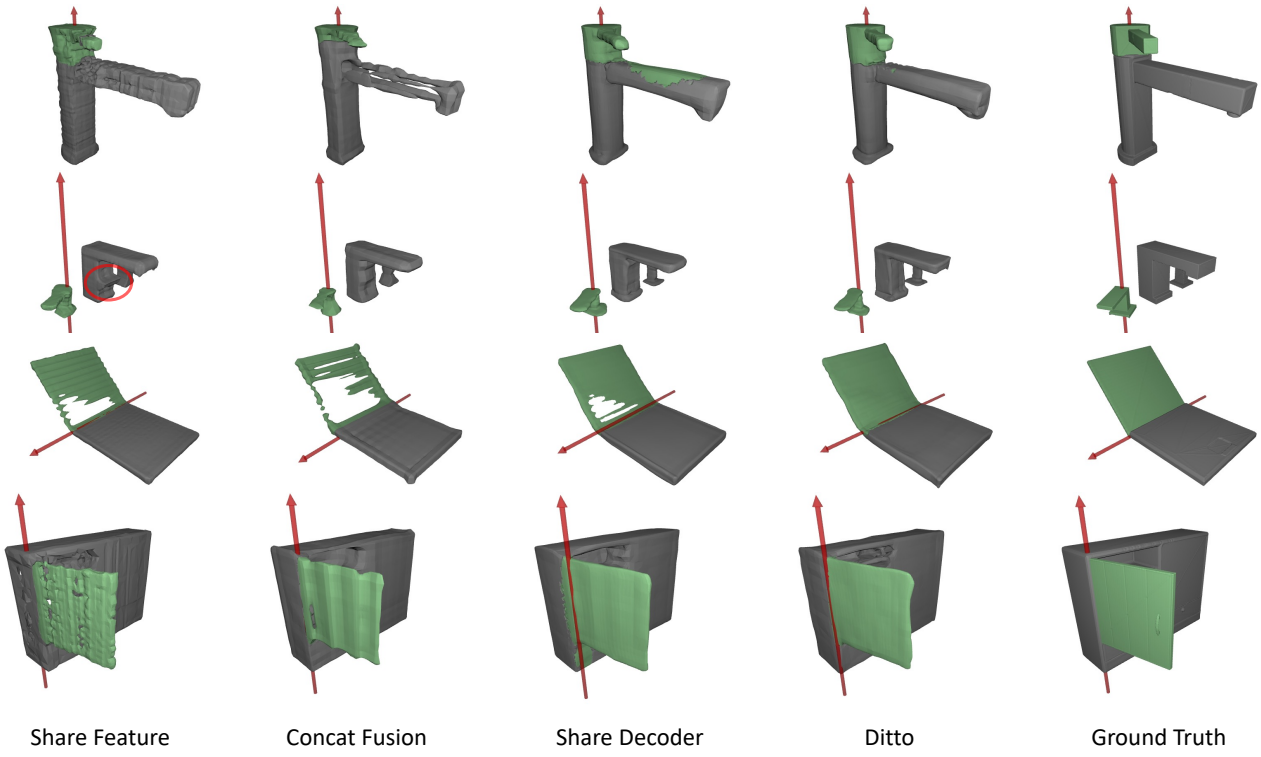

Figure 2. Reconstructed unseen articulated objects in the Shape2Motion [1] dataset of ablated versions. Static parts are colored grey while mobile parts are colored green. We also visualize the estimated joint with the red arrow.

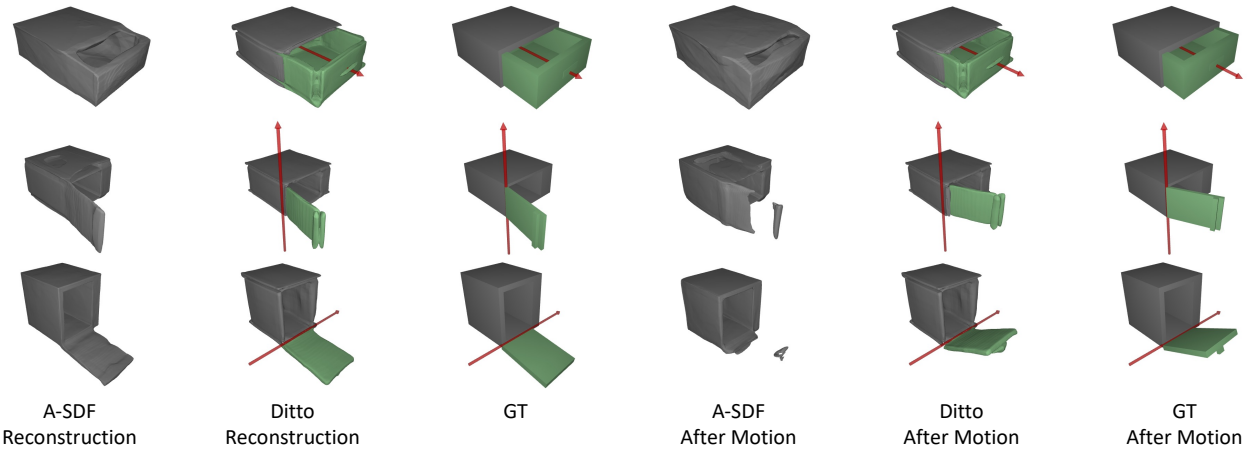

Figure 3. Objects after articulated motion on the Synthetic [2] dataset. Static parts are colored grey while mobile parts are colored green. We also visualize the estimated joint with the red arrow.

constructs the explicit part-level meshes and the articulation model, where the joint state can also be easily controlled. It is thus possible to compare the performance of articulated motion synthesis of A-SDF and Ditto. We first reconstruct the articulated object and then manipulate the articulated object to a new joint state. And we measure the whole Chamfer distance between manipulated results and the ground truth objects after such an articulated motion.

The quantitative results are in Tab. 2. Ditto achieves significantly better results compared with A-SDF. We also show some qualitative results in Fig. 3. Even though A-SDF can generally reconstruct the articulated object from

Method	Chamfer Distance ↓
A-SDF [5]	3.57
Ditto	**0.37**

Table 2. Quantitative results of articulated motion synthesis on the Synthetic [2] dataset.

Method	Joint type accuracy (%)
Global Joint [6]	88
Share Feature	100
Concat Fusion	96
Share Decoder	100
Ditto (Ours)	100

Table 3. Quantitative results of joint type prediction accuracy on the Shape2Motion [1] dataset.

the observation, the results after the articulated motion are not consistent with the initial state due to its latent representation of articulation. For example, the whole drawer body is widened after the motion. In contrast, Ditto explicitly extracts mobile part mesh and the corresponding joint parameters. Apart from the rigid transformation induced by articulated motion, there is no unexpected distortion after the motion.

4. Joint Type Prediction

Our model is also predicting the joint type. All methods give 100% joint type accuracy on the Synthetic dataset. We provide the results on the accuracy of joint type prediction on the Shape2Motion dataset in Tab. 3. Most methods also acquire 100% accuracy on this dataset except the global joint baseline and the ablated version with concat fusion.

5. Generalization to Unseen Categories

In our experiments, we trained our model for four categories altogether and evaluated it on the same four categories. To test generalization to novel categories, we run our model, trained on Shape2Motion, on four unseen categories (drawer, microwave, fridge, and stapler). Fig. 1 shows that our model generalizes robustly to geometrically similar categories (drawer, microwave, and fridge) but slightly worse on the new categories of more significant differences (stapler) as it learns shape priors from the training data.

References

[1] 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. 1, 2, 3

[2] 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. 1, 2, 3

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 1

[4] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[5] Jiteng Mu, Weichao Qiu, Adam Kortylewski, Alan Yuille, Nuno Vasconcelos, and Xiaolong Wang. A-sdf: Learning disentangled signed distance functions for articulated shape representation. *arXiv preprint arXiv:2104.07645*, 2021. 1, 3

[6] Christopher Choy, Jaesik Park, and Vladlen Koltun. Fully convolutional geometric features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8958–8966, 2019. 3