Locate n' Rotate: Two-stage Openable Part Detection with Foundation Model Priors

Siqi Li[1,2], Xiaoxue Chen[3,4], Haoyu Cheng[1,2], Guyue Zhou[3], Hao Zhao[3], and Guanzhong Tian[1,2](✉)

[1] Ningbo innovation Center, Zhejiang University.
[2] College of Control Science and Engineering, Zhejiang University
[3] Institute for AI Industry Research, Tsinghua University
[4] Department of Computer Science and Technology Tsinghua University

Abstract. Detecting the openable parts of articulated objects is crucial for downstream applications in intelligent robotics, such as pulling a drawer. This task poses a multitasking challenge due to the necessity of understanding object categories and motion. Most existing methods are either category-specific or trained on specific datasets, lacking generalization to unseen environments and objects. In this paper, we propose a Transformer-based Openable Part Detection (OPD) framework named Multi-feature Openable Part Detection (MOPD) that incorporates perceptual grouping and geometric priors, outperforming previous methods in performance. In the first stage of the framework, we introduce a perceptual grouping feature model that provides perceptual grouping feature priors for openable part detection, enhancing detection results through a cross-attention mechanism. In the second stage, a geometric understanding feature model offers geometric feature priors for predicting motion parameters. Compared to existing methods, our proposed approach shows better performance in both detection and motion parameter prediction. Codes and models are publicly available at https://github.com/lisiqi-zju/MOPD

Keywords: Openable Part Detection · Geometric Feature · Transformer

1 Introduction

For articulated objects, "openable" refers to an affordance attribute that indicates the parts of objects capable of being opened. For example, a door can be opened through revolute motion, while a drawer can be opened via prismatic motion. Detecting the openable parts within real-world objects is a crucial task in computer vision, with numerous applications in intelligent robotics and manipulation [7, 11, 18, 24]. In this paper, we aim to address the task of Openable Part Detection (OPD), where the input is a single-view image, and the outputs

✉ Corresponding authors

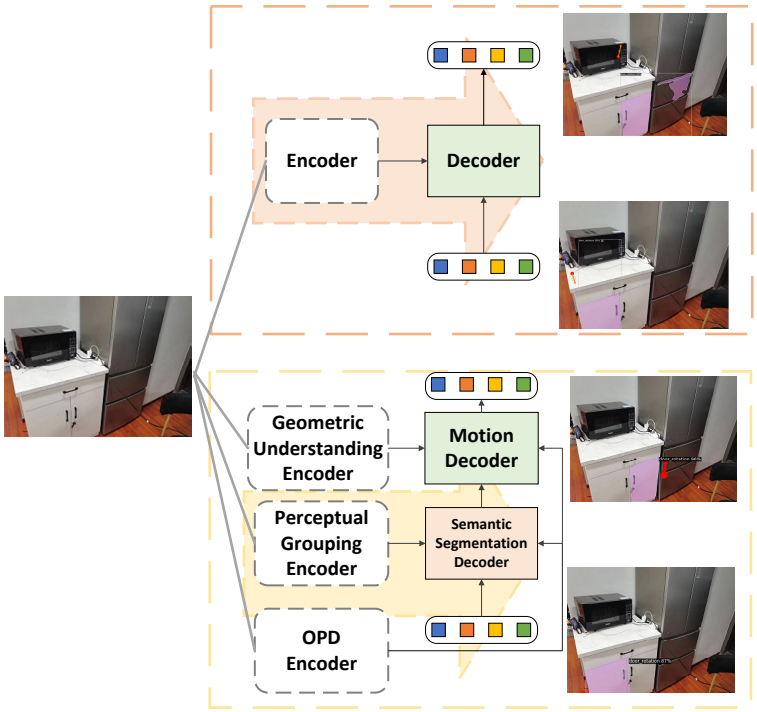

Fig. 1: Comparison of network architecture between our framework and MultiOPD. The outputs are the results on an in-the-wild image. Our model achieves superior performance and showcases generation capabilities to unseen scenarios.

include detected openable parts along with their corresponding motion parameters.

Identifying openable parts within a multi-object scene is a challenging problem, as it can easily be confused by complex indoor furniture. This implies a comprehension of the articulated object's category and function. In addition, the motion parameters of openable parts typically encompass two components: the motion type (prismatic or revolute) and the motion vector consisting of the motion origin and axis. To analyze the motion parameters of openable parts, it's important to establish a geometric understanding of the articulated object's surface.

With the advent of Embodied AI, a range of methods [8, 10, 23, 26] have emerged to analyze the structure of articulated objects and predict their motion parameters. However, these methods often rely on 3D information inputs such as depth images or point clouds, which offer geometric priors for articulated objects. Moreover, many of these approaches are category-specific, limiting their practicality in the context of intelligent robots.

Opd [9] first introduced a category-agnostic method that predicts openable parts along with their corresponding motion parameters for a single openable object from a single-view image. However, this method only considers scenarios with a single articulated object, which may not be practical in real-world applications. Recently, OPDMulti [21] extended OPD to handle multi-object situations. Both methods treat the OPD task as an instance segmentation task and utilize an end-to-end network supervised by ground-truth segmentation masks, without considering the perceptual-level knowledge and geometric priors of the object. This leads to inaccurate predictions of openable parts and their motion parameters. Moreover, the capability of these methods is limited by the training samples and may perform poorly on in-the-wild images, which is illustrated in Fig. 1 (the upper one).

To address the aforementioned challenges in openable part detection, we introduce a two-stage Transformer-based framework named MOPD that incorporates the OPD task with perceptual grouping and geometric prior, resulting in superior performance compared to previous methods. Specifically, in the first stage, we introduce a perceptual grouping feature extracted from a perceptual grouping encoder and fuse it with the input image feature using a cross-attention mechanism. This incorporation provides perceptual grouping priors for OPD and enhances detection results. Additionally, in the second stage, we extract a geometric feature from a geometric understanding encoder and fuse it with the first-stage feature, thereby providing geometric priors for motion prediction. It's worth noting that the perceptual grouping and geometric understanding encoder can be replaced with a similar one, making our model a general framework. Fig. 1 provides an architecture comparison with the previous method. And we also introduced a motion cost in the matching step, leveraging our two-stage forecasting approach and combining it with the Optimal Transport model for training. These enhancements played a crucial role in significantly improving the performance of the model.

2 Related works

Our study focuses specifically on the segmentation task of detecting the openable parts of objects, which is named Openable Part Detection (OPD). By integrating instance and geometric features, we attain superior performance compared to prior OPD methods [9,21].Understanding articulated objects is crucial for the development of intelligent robots. Considerable work has been done in this area, including studies such as [3, 6, 31]. With the rise of Embodied AI, many publicly accessible datasets [1,15,17] and physical simulators [5,12,19,20] have been introduced to establish a solid research foundation for articulated objects. To achieve a comprehensive understanding of articulated objects, many works analyze them from various perspectives, such as 3D shape reconstruction [27, 28] and 6D pose estimation [13, 22]. With the holistic understanding, some works [7,11,18,24] focus on predict manipulation in a robotic system. Furthermore, a series of works [9,10,21,23,26] have been proposed to analyze parts

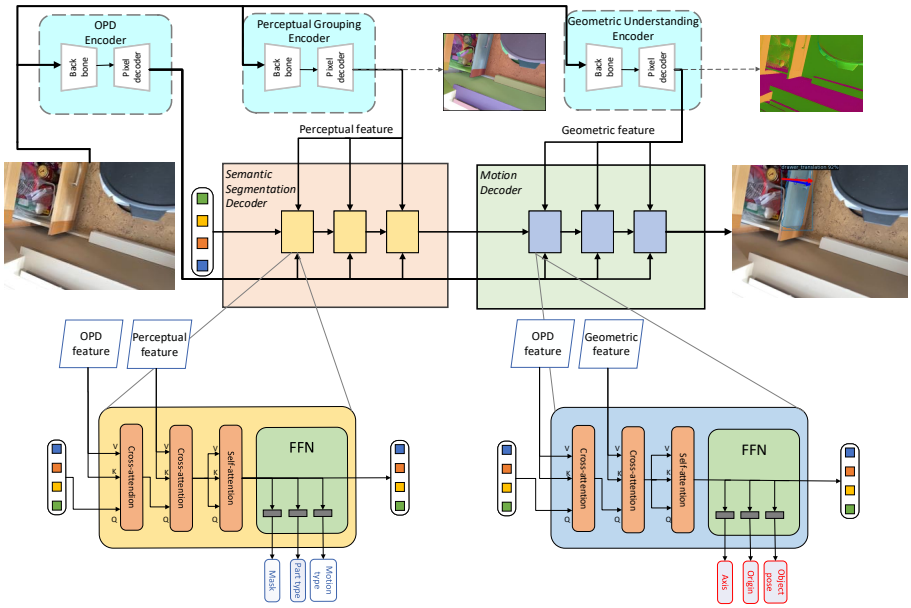

Fig. 2: The overall architecture for MOPD. The top side shows the overall network while the bottom shows the decoder in detail. The model employs three encoders to extract the features from the images. The pixel-level embeddings from the encoder are passed to the transformer decoder with learnable part queries to learn embeddings that are used to predict the openable part. The OPD feature and perceptual grouping feature are successively crossed in the segmentation decoder to obtain a high-resolution mask. In the same way, the OPD feature and geometric feature are used in the motion decoder. The motion type, part type, and mask are predicted in all FFN layers of the semantic segmentation decoder, while the object poses, origin, and axis are predicted in all FFN layers of the motion decoder. The image on the far right shows the output. The GT axis is in blue and the predicted axis is in red.

of articulated objects using RGB images or point clouds, which is crucial for understanding their structures and supporting part-level manipulation. Single-view geometric understanding has received significant attention due to its lower hardware requirements and wider application scope, as evidenced by studies such as [16, 29, 30, 32].For instance, Ditto [10] reconstructs part-level geometry of articulated objects from visual observations of interactions, while OPDMulti [21] segments the openable parts of articulated objects from single-view images using a Transformer-based architecture.

3 Methods

In this paper, our aim is to detect the openable parts of articulated objects based on RGB images. In this section, we begin by formally defining the Open-

able Part Detection (OPD) task in Section 3.1. Following that, we provide an overview of the overall framework in Section 3.2. Additionally, we introduce the details of multi-feature for perceptual grouping and geometric understanding in Section 3.3. We then delve into the network architectures in Section 3.4 and 3.5. Finally, we introduce the our matching strategy with optimal transport assignment.

3.1 Preliminary for OPD Task

The objective of openable part detection is to identify all openable components from single vision. Unlike traditional instance segmentation tasks, which focus on delineating individual objects, openable part detection requires a deeper understanding of the concept of "openable". While this concept is intuitive for humans, it is challenging to precisely define. Therefore, we simplify the notion of openable by decomposing it into distinct constituent parts within the dataset following [21]. Specifically, an openable part p_i is characterized by a 2D bounding box b_i or a segmentation mask m_i, along with a motion axis direction $a_i \in R^3$ and motion origin $o_i \in R^3$ (for revolute motion type only). Each openable part is categorized into one of three semantic types $l_i \in \{drawer, door, lid\}$and one of two motion types $c_i \in \{prismatic, revolute\}$. Since there is usually more than one articulated object in a real-world indoor scene, an output from an image will consist of a set of openable parts $P = \{p_i, ..., p_k\}$.

3.2 Overall Network Architecture

To address the Openable Part Detection (OPD) task, we introduce a novel methodology called MOPD (Multi-feature Openable Part Detection), which is based on Mask2former [4]. In the upcoming paragraphs, we will delve into the differences between our approach and Mask2former to highlight the contributions of our OPD model.

The network structure of Mask2former mainly includes three stages: multi-level feature extraction based on Backbone, a pixel-level decoder based on a multi-level deformable self-attention mechanism, and a multi-head cross-attention mechanism decoder based on the mask. In MOPD, we propose two key network architecture contributions to enhance the performance of the OPD task:

1. In OPD task the motion parameters are predicted with detection together, by the same query through the different model heads. It made the two kinds of prediction disturb each other. So we predict them in two decoders which incorporate different features. To incorporate perceptual grouping and geometric priors, two types of additional features are introduced through specialized feature encoders. These feature encoders comprise a backbone and a pixel-level decoder. The input image is processed to obtain these features, which subsequently serve as the key and value vectors for input into the transformer decoder. The details will be depicted in the next section.

2. In order to address the OPD task, we propose a two-stage task-specific framework, each corresponding to perceptual grouping and geometric features respectively. In the first stage, we emphasize the semantic information of the objects and predict bounding boxes (or masks), part types, and motion types. In the second stage, we incorporate geometric features and output object poses, origins, and axes. Through different FNN layers, the two categories of predictions are generated in different modules successively.

The overall model architecture is depicted in Fig. 2. The model to maintain real-time utilizes a ResNet-50 backbone as the Openable Part Detection (OPD) encoder to extract features from the input image. Additionally, it employs a perceptual grouping feature encoder sourced from EfficientSAM [25], along with a geometric understanding encoder from DSINE [2]. Both the perceptual grouping encoder and geometric understanding encoder are pre-trained on their respective tasks and then fine-tuned with the entire model for the openable part detection task. The features extracted from the perceptual grouping encoder and geometric understanding encoder are fused with the image features using a cross-attention layer within the transformer.

3.3 Multi-Feature of Perceptual grouping and Geometric

Given the perceptual grouping and geometric nature of the OPD task, our approach focuses on enhancing model performance and generalization by incorporating perceptual grouping and geometric features learned from other computer vision tasks. These features are extracted from models pre-trained in their respective tasks, and we fine-tune their encoders specifically for the OPD task. Specifically, we introduce two types of encoders to aid in detecting openable parts: a perceptual grouping encoder to help the model understand the categories of articulated objects and a geometric understanding encoder to produce spatial features to assist in predicting the origin, axis, and object pose. These encoders are combined with the backbone and pixel decoder. The image passes through the backbone to obtain the embedding, and then through the pixel decoder to extract the feature. These features are utilized as a key and value vector in the cross-attention layer to fuse with the query vector obtained by the OPD encoder. Perceptual grouping encoders and geometric encoders are respectively be pretrained by EfficientSAM and DSINE.

EfficientSAM: A lightweight SAM model that exhibits decent performance with largely reduced complexity. It takes SAM pre-trained lightweight image encoders and mask decoder to build EfficientSAMs and finetune the models on SA-1B for segment anything task [25].

DSINE: It utilizes the per-pixel ray direction and encodes the relationship between neighboring surface normals by learning their relative rotation. It shows a stronger generalization ability, despite being trained on an orders of magnitude smaller dataset [2].

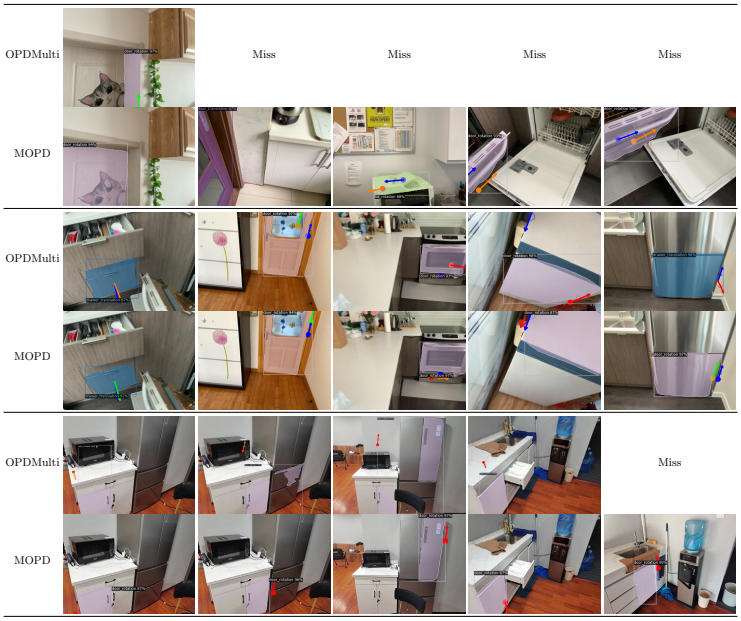

Fig. 3: Qualitative results on the OPDMulti and MOPD val split. The first two rows are a comparison of MOPD variants with OPDMulti in valid dataset. The last rows are a comparison in the wild. The GT axis is in blue and the predicted axis is in green if it is within 5° of the GT,orange if between 5°and 10° and red if the angle difference is greater than 10°.

3.4 Transformer Decoder

To obtain high-resolution masks, we employ a two-stage strategy utilizing multi-scale visual feature maps at increasing resolutions, each of which is respectively inputted into the perceptual grouping and geometric understanding decoder. The transformer comprises two decoders: one for semantic segmentation stacked for L_1 layers, and the other for motion prediction stacked for L_2 layers. Each layer in the transformer stack consists of two cross-attention layers, one self-attention layer, and a feedforward neural network (FFN) layer. The mask and type prediction are separated from the motion prediction. By employing different combinations of prediction heads, the prediction position can be flexibly adjusted at any position in the decoder layers. A comprehensive overview of the architecture is depicted in Fig. 2.

3.5 FFN layer and Training Losses

For the segmentation and motion losses, we add the auxiliary loss after each transformer decoder. And we predict them successively in two different transformer decoders with different features.

1. **Segmentation losses**: The mask segmentation loss for the first stage comprises the following components: binary cross-entropy loss L_{ce}, the dice loss L_{dice}, cross-entropy loss L_{cls} and the motion type cross-entropy loss L_c. The overall formulation is: $L_{seg} = \lambda_{cd}L_{ce} + \lambda_{dice}L_{dice} + \lambda_{cls} + L_{cls} + \lambda_c L_c$, where $\lambda_.$ is the loss weight.
2. **Motion losses**: The motion losses for the second stage consist of the following components: smooth L1 losses for the motion axis L_a, motion origin losses L_o and object pose losses L_o. $L_{mot} = \lambda_a L_a + \lambda_o L_o + \lambda_{pose} L_{pose}$, where λ_a, λ_o, and λ_{pose} are the weighting coefficients for each respective loss component.

We sum the segmentation loss and the motion loss to obtain the overall loss used during training: $L = L_{seg} + L_{mot}$.

3.6 Optimal Transport

Traditional object detectors perform detection by predicting classification labels and regression offsets for a set of proposals. To train the detector, matching targets for each proposal is a necessary process. Most strategies may result in suboptimal proposal assignments for each ground truth individually without context, as assigning ambiguous proposals to any ground truth might bring harmful gradients to other ground truths. To achieve a globally optimal assignment result in a one-to-many situation, optimal transport formulates label assignment as an Optimal Transport (OT) problem. Specifically, The cost between the ground truth and a proposal is defined solely by their pairwise classification cost. After formalizing this, finding the optimal assignment scheme is transformed into solving the optimal transport plan, which can be efficiently and quickly solved using the ready-made Sinkhorn-Knopp iteration. We name this assignment strategy Optimal Transport Assignment (OTA).

Previous works keep working on the object matching, lacking the attention of matching motion itself. To address the influence of motion matching, we propose the match cost of motion. Our proposed match cost includes the following two aspects:

1. **Origin Match cost**: We set the origin match cost as the normalized cross product of the predicted origin drift and the ground truth axis:

$$C_{origin} = \frac{(\mathbf{O}_{pred} - \mathbf{O}_{gt}) \times \mathbf{I}_{gt}}{L_{diag}}$$

   where L_{diag} is the diagonal length of the object, characterizing the size.
2. **Axis Match cost**: The axis match cost is the angular difference between the predicted axis and the ground truth axis:

$$C_{axis} = arccos(\frac{I_{pred} \cdot I_{gt}}{|I_{pred}||I_{gt}|})$$

We sum the origin match cost and the axis match cost to obtain the overall matching cost matrix used during training: $C = C_{obj} + C_{origin} + C_{axis}$.$C_{obj}$ represents the matching cost matrix in a traditional detection task.

4 Experiments

In this section, we conduct experiments to verify the effectiveness of our model and compare MOPD and several varieties with the previous baselines. We also show the efficiency of our algorithm with different modules through ablation studies.

4.1 Evaluation Metrics

We follow the evaluation metrics for part detection and motion prediction used in OPDmulti [21]. The metrics extend the traditional mAP metric. To evaluate the detection of openable parts the metrics include several metrics. First is AP@IoU=0.5 for the predicted part label and 2D bounding box (PDet) or mask. On the basis of PDet to evaluate the motion parameters, For each metric, the detection is further constrained by whether: the motion type is matched (+M), motion type and motion axis are matched (+MA), and whether the motion type, axis, and origin are all matched (+MAO), within predefined error thresholds [21].

Table 1: Quantitative results. The MOPD model utilizes EfficienceSAM and DSINE as the perceptual grouping encoder and geometric understanding encoder.

Model	Part-averaged mAP % ↑			
	PDet	+M	+MA	+MAO
OPDRCNN-C	27.3	25.7	8.8	7.8
OPDRCNN-O	20.0	18.3	3.9	0.5
OPDRCNN-P	20.9	19.0	7.2	5.7
OPDFORMER-C	30.3	28.9	13.1	12.1
OPDFORMER-O	30.1	28.5	5.2	1.6
OPDFORMER-P(OPDMulti)	32.9	31.6	19.4	16.0
MOPD	37.3	36.1	**20.7**	16.6
MOPD(Optimal Transport)	**37.8**	**37.7**	20.1	**17.2**

4.2 Implementation details

Our model is implemented based on Mask2Former [4]. We employ a ResNet-50 backbone pre-trained on the COCO dataset [14], with a learning rate of 0.0001. Experiments are conducted on the OPDMulti dataset [21] using an A100-SXM4-80GB GPU. Models evaluated on the OPDMulti dataset are initially trained on the OPDReal dataset [9], followed by fine-tuning on the OPDMulti dataset. Training is performed end-to-end for 60000 steps, and the best checkpoint is selected based on validation set performance (using the +MAO metric). A confidence threshold of 0.8 is applied to determine the validity of predicted parts.

4.3 Quantitative and Qualitative Results

As shown in Table 1, our MOPD framework demonstrates notable superiority over the OPDMulti baselines across various metrics, improving part detection

Fig. 4: At the top, there is a comparison of the results obtained from the w/o perceptual grouping encoder in MOPD. At the bottom, there is the output from EfficientSAM, which we utilized to pre-train the encoder. The figure demonstrate that our model indeed utilizes the pre-trained encoder. Since the DETR is a query-based model, it can occasionally detect two distinct objects as a single entity. However, by leveraging the segmentation capabilities inherent in the EfficientSAM model, we are able to effectively mitigate such errors and improve the overall accuracy of detections. The quantitative result are shown in Table 3.

Table 2: In the second row, the perceptual grouping encoder is replaced with SAM. In the third row, the geometric understanding encoder is replaced with Omnidata. In the third row. We frozen EfficientSAM and DSINE.

Model	PDet	+M	+MA	+MAO
OPDMulti	32.9	31.6	19.4	16.0
MOPD (with SAM)	34.3	33.1	19.6	15.9
MOPD (with Omnidata)	33.5	32.3	18.9	16.5
MOPD (frozen)	35.4	32.7	19.6	16.1
MOPD	**37.3**	**36.1**	**20.7**	**16.6**

mAP by 4.9% and motion parameter accuracy by 1.2%, respectively. The remarkable performance of our approach can be attributed to several key factors, with one of the primary contributors being the integration of the perceptual grouping encoder and geometric understanding encoder. These encoders collaborate to extract essential features that significantly improve part detection accuracy. The perceptual grouping encoder identifies and segments different parts of the object, while the geometric understanding encoder captures geometric details and surface normal, providing additional context for enhanced detection. This integration improves part detection and enhances the accuracy of motion parameters. The extracted features enable a more precise understanding of the object's shape, see Fig.4. Resulting in more accurate motion parameter estimations see Fig. 5.

Qualitative results are illustrated in Fig. 3. It demonstrates the detection of a wide range of openable parts, a task also performed by OPDMulti. However,

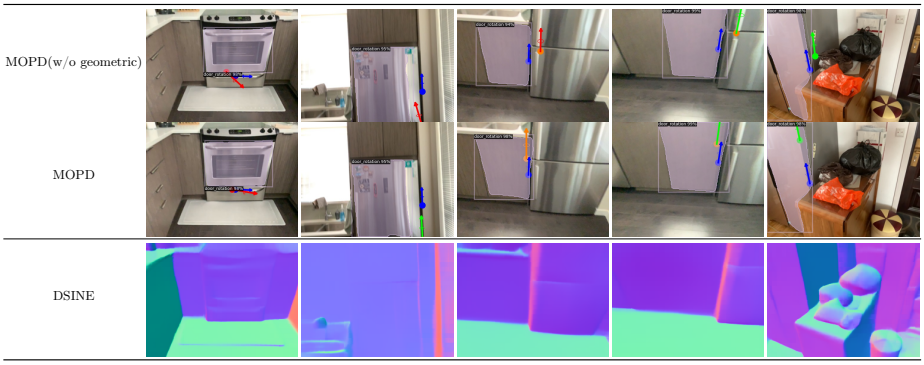

Fig. 5: At the top, there is a comparison of the results obtained from the w/o geometric encoder in MOPD. At the bottom, there is the output from DSINE, which we utilized to pre-train the geometric encoder. Through the plugging in of geometric features, the model corrects the axis direction to make it closer to the surface normal evaluation. It indicates that the decoder indeed takes advantage of geometric features. When comparing the two models, we can observe that our model has better precision with origin prediction, especially when the two models have a similar axis prediction. This is because the RGB picture lacks information regarding the degree of the two crossing surfaces, which makes the model unable to provide an accurate prediction of Three-dimensional coordinates when the axis is near the edge of the door and lid. The introduction of normal features can alleviate this issue by pushing the origin away from the incorrect surface.

MOPD not only detects these parts but also accurately estimates their motion parameters. This capability is crucial in applications such as robotics and automation, where a precise understanding of object motion is vital for effective interaction and manipulation.

PR-curves of MOPD and OPDMulti are compared in Fig. 6. It consistently demonstrates that MOPD exhibits high precision at low recall rates. This implies that there are fewer false positives and false negatives at high thresholds. This indicates that our model performs better in detecting objects that more closely align with the definition of an openable part.

Table 3: Ablation study: It indicates that geometric features can not only enhance the prediction of motion parameters but also improve the ability of part detection. This is because pixels with the same surface normal are more likely to belong to a same part.

Model	PDet	+M	+MA	+MAO	Model Size	Test Memory	Training Memory	Computation time per image
OPDMulti	32.9	31.6	19.4	16.0	188MB	4942MB	175868 MB	0.190s
MOPD (w/o geometric)	34.3	33.1	19.6	15.9	224MB	5433MB	187393 MB	0.207s
MOPD (w/o perceptual)	36.7	35.4	20.2	**16.7**	297MB	5768MB	190352 MB	0.212s
MOPD	**37.3**	**36.1**	**20.7**	16.6	371MB	5980MB	196308 MB	0.228s

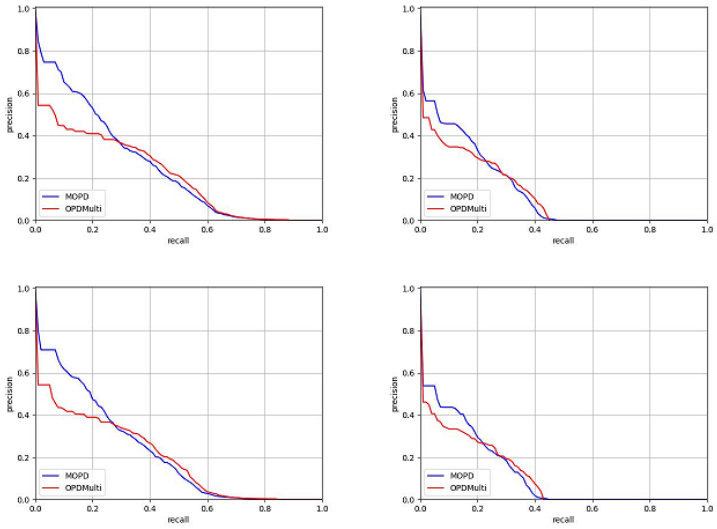

Fig. 6: PR-curve: Top left: mAP@IoU=0.5 for the detection of openable parts using bounding boxes. Top right: mAP@IoU=0.5 for the prediction of motion parameters using bounding boxes. Bottom left: mAP@IoU=0.5 for the detection of openable parts with masks. Bottom right: mAP@IoU=0.5 for the prediction of motion parameters with masks.

Alternative pretrained models We first conducted experiments using SAM and Omnidata. Then, we found that EfficientSAM and DSINE are better. The result are shown in Tabel 2.

4.4 Optimal Transport

As shown in Table. 1 and Figure. 7, our proposed motion match cost effectively improves the accuracy of motion axis and origin matching. The improvement brought by our method is due to C_{origin} providing supervision for origin matching, thus limiting the drift of the motion origin. More importantly, C_{axis} imposes constraints on the inclination of the motion axis, and more accurate motion trajectories lead to more accurate motion matching.

4.5 Efficiency

The primary objective of openable part detection is to tackle the intricate challenge of enabling robotics to interact seamlessly with openable objects. Given the complexity involved in such interactions, efficiency becomes a paramount metric for evaluating the performance of any algorithm designed to achieve this goal. After all, an efficient algorithm can significantly enhance the robot's ability to perform tasks quickly and accurately.

(a) Ground Truth **(b)** MOPD(w/o OT) **(c)** MOPD

Fig. 7: Qualitative Results of Ablation Study on Optimal Transport. It is very intuitive that Optimal Transport's improvement in accurately perceiving the motion axis and origin for MOPD is very significant.

To assess the efficiency of our approach, we conducted a series of experiments and compiled the results in Table 3. This table provides a comprehensive overview of how our model, equipped with DSINE and EfficientSAM, fares in terms of prediction accuracy and overall efficiency. As the table clearly illustrates, our model exhibits superior performance in both these aspects.

Notably, our approach also demonstrates that a smaller encoder can effectively address the OPD problem while maintaining a lightweight design. This is a significant advantage as it reduces the computational burden on the robot, enabling it to operate more efficiently and with less power consumption. In turn, this can lead to longer operational durations and fewer maintenance requirements, making our approach highly practical and appealing for real-world robotics applications.

5 Conclusion

Openable part detection plays a crucial role in applications involving interaction with articulated objects. This paper presents a two-stage OPD Transformer-based framework that integrates perceptual grouping and geometric features. In the initial stage, we introduce a perceptual grouping encoder to provide perceptual grouping feature priors for openable part detection, thereby improving detection outcomes through a cross-attention mechanism. Subsequently, in the second stage, a geometric understanding encoder offers geometric feature priors for detecting motion parameters. Finally, we introduced a motion cost in matching step, combined with the Optimal Transport model for training, which significantly improved the performance of the model. Extensive experiments demonstrate that our method outperforms previous approaches in both generalization and performance. Furthermore, ablation studies validate the effectiveness of the proposed models. We believe our method will bring benefits to downstream robotics applications.

Acknowledgments: This work is supported in part by the National Natural Science Foundation of China under Grant 62303405, in part by Ningbo Natural Science Foundation Project under Grant 2023J400, and in part by Ningbo Key Research and Development Plan under Grant 2023Z116.

References

1. Abbatematteo, B., Tellex, S., Konidaris, G.: Learning to generalize kinematic models to novel objects. In: Proceedings of the 3rd Conference on Robot Learning (2019)
2. Bae, G., Davison, A.J.: Rethinking inductive biases for surface normal estimation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
3. Chen, X., Liu, T., Zhao, H., Zhou, G., Zhang, Y.Q.: Cerberus transformer: Joint semantic, affordance and attribute parsing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19649–19658 (2022)
4. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-attention mask transformer for universal image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1290–1299 (2022)
5. Deitke, M., Han, W., Herrasti, A., Kembhavi, A., Kolve, E., Mottaghi, R., Salvador, J., Schwenk, D., VanderBilt, E., Wallingford, M., et al.: Robothor: An open simulation-to-real embodied ai platform. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3164–3174 (2020)
6. Ding, K., Chen, B., Wu, R., Li, Y., Zhang, Z., Gao, H.a., Li, S., Zhu, Y., Zhou, G., Dong, H., et al.: Preafford: Universal affordance-based pre-grasping for diverse objects and environments. arXiv preprint arXiv:2404.03634 (2024)
7. Eisner, B., Zhang, H., Held, D.: Flowbot3d: Learning 3d articulation flow to manipulate articulated objects. arXiv preprint arXiv:2205.04382 (2022)
8. Hu, R., Savva, M., van Kaick, O.: Functionality representations and applications for shape analysis. In: Computer Graphics Forum. vol. 37, pp. 603–624. Wiley Online Library (2018)
9. Jiang, H., Mao, Y., Savva, M., Chang, A.X.: Opd: Single-view 3d openable part detection. In: European Conference on Computer Vision. pp. 410–426. Springer (2022)
10. Jiang, Z., Hsu, C.C., Zhu, Y.: Ditto: Building digital twins of articulated objects from interaction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5616–5626 (2022)
11. Katz, D., Brock, O.: Manipulating articulated objects with interactive perception. In: 2008 IEEE International Conference on Robotics and Automation. pp. 272–277. IEEE (2008)
12. Li, C., Xia, F., Martín-Martín, R., Lingelbach, M., Srivastava, S., Shen, B., Vainio, K., Gokmen, C., Dharan, G., Jain, T., et al.: igibson 2.0: Object-centric simulation for robot learning of everyday household tasks. arXiv preprint arXiv:2108.03272 (2021)
13. Li, X., Wang, H., Yi, L., Guibas, L.J., Abbott, A.L., Song, S.: Category-level articulated object pose estimation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3706–3715 (2020)
14. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13. pp. 740–755. Springer (2014)
15. Liu, L., Xu, W., Fu, H., Qian, S., Yu, Q., Han, Y., Lu, C.: Akb-48: A real-world articulated object knowledge base. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14809–14818 (2022)

16. Long, X., Zheng, Y., Zheng, Y., Tian, B., Lin, C., Liu, L., Zhao, H., Zhou, G., Wang, W.: Adaptive surface normal constraint for geometric estimation from monocular images. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024)
17. Martín-Martín, R., Eppner, C., Brock, O.: The rbo dataset of articulated objects and interactions. The International Journal of Robotics Research **38**(9), 1013–1019 (2019)
18. Mo, K., Guibas, L.J., Mukadam, M., Gupta, A., Tulsiani, S.: Where2act: From pixels to actions for articulated 3d objects. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6813–6823 (2021)
19. Savva, M., Kadian, A., Maksymets, O., Zhao, Y., Wijmans, E., Jain, B., Straub, J., Liu, J., Koltun, V., Malik, J., et al.: Habitat: A platform for embodied ai research. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9339–9347 (2019)
20. Shen, B., Xia, F., Li, C., Martín-Martín, R., Fan, L., Wang, G., Pérez-D'Arpino, C., Buch, S., Srivastava, S., Tchapmi, L., et al.: igibson 1.0: a simulation environment for interactive tasks in large realistic scenes. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 7520–7527. IEEE (2021)
21. Sun, X., Jiang, H., Savva, M., Chang, A.X.: Opdmulti: Openable part detection for multiple objects. arXiv preprint arXiv:2303.14087 (2023)
22. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2642–2651 (2019)
23. Wang, X., Zhou, B., Shi, Y., Chen, X., Zhao, Q., Xu, K.: Shape2motion: Joint analysis of motion parts and attributes from 3d shapes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8876–8884 (2019)
24. Wang, Y., Wu, R., Mo, K., Ke, J., Fan, Q., Guibas, L.J., Dong, H.: Adaafford: Learning to adapt manipulation affordance for 3d articulated objects via few-shot interactions. In: European conference on computer vision. pp. 90–107. Springer (2022)
25. Xiong, Y., Varadarajan, B., Wu, L., Xiang, X., Xiao, F., Zhu, C., Dai, X., Wang, D., Sun, F., Iandola, F., et al.: Efficientsam: Leveraged masked image pretraining for efficient segment anything. arXiv preprint arXiv:2312.00863 (2023)
26. Yan, Z., Hu, R., Yan, X., Chen, L., Van Kaick, O., Zhang, H., Huang, H.: Rpm-net: recurrent prediction of motion and parts from point cloud. arXiv preprint arXiv:2006.14865 (2020)
27. Yang, G., Sun, D., Jampani, V., Vlasic, D., Cole, F., Chang, H., Ramanan, D., Freeman, W.T., Liu, C.: Lasr: Learning articulated shape reconstruction from a monocular video. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15980–15989 (2021)
28. Yang, G., Vo, M., Neverova, N., Ramanan, D., Vedaldi, A., Joo, H.: Banmo: Building animatable 3d neural models from many casual videos. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2863–2873 (2022)
29. Zhao, H., Lu, M., Yao, A., Guo, Y., Chen, Y., Zhang, L.: Physics inspired optimization on semantic transfer features: An alternative method for room layout estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 10–18 (2017)

30. Zheng, Y., Li, X., Li, P., Zheng, Y., Jin, B., Zhong, C., Long, X., Zhao, H., Zhang, Q.: Monoocc: Digging into monocular semantic occupancy prediction. arXiv preprint arXiv:2403.08766 (2024)
31. Zhong, C., Zheng, Y., Zheng, Y., Zhao, H., Yi, L., Mu, X., Wang, L., Li, P., Zhou, G., Yang, C., et al.: 3d implicit transporter for temporally consistent keypoint discovery. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3869–3880 (2023)
32. Zhong, L., Zhang, Y., Zhao, H., Chang, A., Xiang, W., Zhang, S., Zhang, L.: Seeing through the occluders: Robust monocular 6-dof object pose tracking via model-guided video object segmentation. IEEE Robotics and Automation Letters **5**(4), 5159–5166 (2020)