2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Grasp Manipulation Relationship Detection based on Graph Sample and Aggregation

Jiayuan Luo[1*], Yaxin Liu[1*], Han Wang[1], Mengyuan Ding[1] and Xuguang Lan[1†]

***Abstract*— In multi-object stacking scenarios, exploring the relationships among objects and determining the correct sequence of operations are crucial for robotic manipulation. However, previous algorithms inefficiently combine global and local information, often focusing solely on the local features of objects or the interactions of object features at a global level. This approach leads to imbalanced distribution of features and the generation of redundant or missing relationships in complex scenes, such as multi-object stacking and partial occlusion. To address this issue, we have developed a grasp manipulation relationship detection algorithm called Graph Sampling Aggregation Network for Visual Manipulation Relationship Detection (GSAGED). This algorithm assists robots in detecting targets in complex scenes and determining the appropriate grasping order. Firstly, the Positional Encoding Module in GSAGED enhances object feature information by considering global contexts. Secondly, the Graph Sampling Aggregation method effectively integrates global and local information, relieving imbalanced distribution of features. Finally, we applied the developed algorithm to a physical robot for grasping. Experimental results on the Visual Manipulation Relationship Dataset (VMRD) and the large-scale relational grasp dataset named REGRAD demonstrate that our method significantly improves the accuracy of relationship detection in complex scenes and exhibits robust generalization capabilities in real-world applications.**

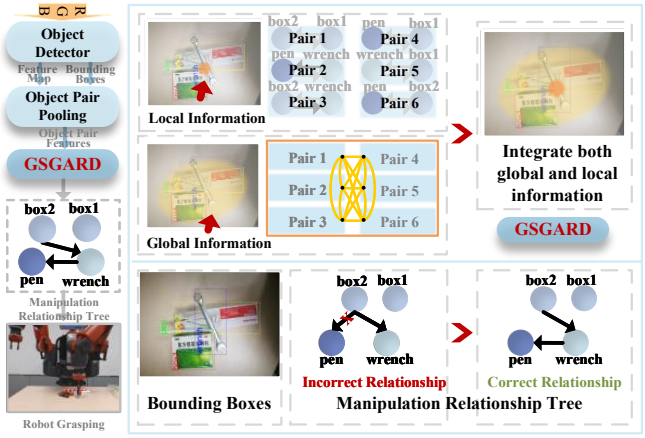

Fig. 1. **Model Architecture and Manipulation Relationship Significance.** Our model architecture takes RGB images as input, detects objects through the object detector. The paired features extracted from this process are then fed into GSAGED. The resulting output consists of manipulation relationship trees, which provide crucial guidance for the robotic grasping sequence. GSAGED integrates global and local information efficiently which reduces generating redundant or missing relations in complex scenes. The accuracy of these relationships is of paramount importance for ensuring the robustness and effectiveness of robot grasping operations.

I. INTRODUCTION

The rapid advancement of artificial intelligence has ushered in increasingly complex scenarios for robotics applications. Robot grasping represents the fundamental interaction between a robot and its environment. Robot grasping with improper grasping order in object stacked scenes can disrupt the integrity of the surroundings and even pose safety risks to operators. For instance, attempting to directly grasp a book beneath a ceramic cup will result in the cup breaking. The correct grasping order is to first grasp the cup and then the book. Therefore, choosing the correct grasping order, that is, detecting the grasp manipulation relationship, is a crucial problem to the robust grasping of the robot.

Recently there has been some work focused on this task. Some algorithms[1], [2], [3] primarily model the relationship between pairs of objects independently, ignoring the global context information of the whole scene. These approaches overlook the potential associations among features in complex scenes, which may generate redundant or missing relations in complex scenes. The others[4] solved this problem by judging the relationship between each pair of objects in combination with the global context information, represented by the yellow lines in the Fig. 1, to better understand the complex scene. However, placing excessive emphasis on associations among object pair features leads to imbalanced distribution of features, accompanied by computational inefficiencies, time-intensive processes, and subpar generalization capabilities.

To overcome the aforementioned problems, we propose a grasp manipulation relationship detection algorithm based on graph sampling aggregation network (GSAGED). GSAGED is specifically designed to detect grasp manipulation relationship, as depicted in Fig. 1. Our network takes object pair features as input and begins by employing a feature positional encoding module to enhance potential associations among relational features. Subsequently, the graph sampling aggregation method optimizes the potential relationships between object relationship features. Finally, the optimized features are supplied to a classifier to determine the manipulation relationship between objects. When provided with an image, the output comprises object categories, relationships, and the appropriate manipulation sequence. GSAGED can effectively integrate both global and local information extracted from

[1]J. Luo, Y. Liu, H. Wang, M. Ding, and X. Lan are with the National Key Laboratory of Human-Machine Hybrid Augmented Intelligence, National Engineering Research Center of Visual Information and Applications, Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China.
*J. Luo and Y. Liu are co-first authors.
†Corresponding author: xglan@mail.xjtu.edu.cn

object pair features to weaken the imbalanced distribution of features. It further provides accurate grasping sequences, a crucial element in robotic manipulation.

Our method is rigorously trained on the VMRD and REGRAD datasets, and subsequently deployed on real robots. Our robotic experiments in real-world scenarios meet stringent real-time requirements and achieve state-of-the-art performance. This practical validation underscores the effectiveness and applicability of our approach in real-world applications.

II. Related Work

A. Robotic Grasping

Intelligent robots face the complex task of identifying grasping targets, determining their poses, and locating suitable grasping points by sensing and comprehending their environments. Early algorithms primarily designed for single-target grasping scenarios struggle to meet the demands of grasping in cluttered real-world environments. Several researchers have focused on designing robots for grasping poses in complex scenes. Mahler et al. proposed the DEX network 2.0 [5], which swiftly predicts the probability of successful grasping from depth images. Fang et al.[6] proposed a task-oriented grasping network, optimizing task-oriented tool grasping and tool manipulation strategies simultaneously. In addition, some researchers[4], [7] have designed multi-objective grasping systems that focus on establishing the appropriate robot grasping sequence in complex scenes. However, these methods often suffer from low detection accuracy or require excessive processing time to achieve real-time robust grasping. The recent REGRAD dataset [8] has enriched research by providing object segmentation, pose, capture, and relationship information in scenes captured through both 2D and 3D images.

B. Grasp Manipulation Relationship Detection

To enable robust object grasping in complex scenarios, understanding object relationships is paramount. Visual manipulation relationship detection aims to precisely identify spatial relationships between pairs of objects within a scene. Zhang et al. [7] defined visual manipulation relationships as the correct order of robot grasping operations, including three types: parent, child, and no relations. Zhang et al. [1] directly detected manipulation relationships among stacked objects, constructing manipulation relationship trees. Yang et al. [2] introduced a fully connected conditional random field that improved grasping operation relationship detection accuracy by imposing global constraints on object stacking scenarios. Unfortunately, the conditional random field network cannot handle the misclassification of more than three relations. Zuo et al. [3] collected contextual relationships between objects by designing a graph convolution network, enhancing the efficiency of relational reasoning. The above works only concentrate on the relationships between two objects, ignoring the global features. Ding et al.[4] proposed a relationship detection model based on a graph neural network to optimize object features in a scene by combining global contextual information which focuses too much on the interaction of object features at the global level and wastes computing resources. However, The existing algorithms inefficiently combine the global context information and the local context information, generating redundant or missing relations in complex scenes. The low accuracy and high time consumption can't meet the requirements of robust real-time robot grasping. In response, we propose a model combining graph sampling and graph aggregation to better balance the optimization of global and local information.

C. Graph Neural Network

As a non-Euclidean data structure, graphs are used in many scenarios for high-performance processing of tasks such as node classification, link prediction, and clustering. Graph neural networks (GNNs)[9], [10] enable the learning process to be built directly on graph data. In recent years, several convolutional neural network architectures for learning over graphs have been proposed [9], [12], [13], [14], [15], [16]. GraphSAGE[21], a general inductive framework that leverages node feature information to efficiently generate node embeddings for previously unseen data, can be viewed as an extension of the GCN framework to the inductive setting. Our use of the graph neural network in deep learning parallels prior work solving network embedding problems[10], relational problems[17], [18], sequential problems[19] and classification problems[20]. So far, graph convolutional networks have matured in the detection of grasping operational relations. Most of them use graph structures to model a set of objects and their relations. While previous works have advanced the detection of grasping operational relations using graph convolutional networks, our approach stands apart by introducing a graph sampling aggregation network that effectively combines global and local information. Our graph sampling aggregation algorithm aligns with the strategy employed in the earlier GraphSAGE algorithm, as proposed by Hamilton et al. [21]. In contrast to the original algorithm, our approach primarily concentrates on reconstructing the graph's adjacency matrix while diverting its focus away from capturing information between feature contexts during the sampling phase. Consequently, our algorithm is most suitable for scenarios involving the reconstruction and induction of adjacency matrices with a limited number of nodes and relatively lower information content.

III. Proposal Approach

A. Problem Definition

The relationship between two objects can be divided into three categories: object 1 should be grasped before object 2, object 1 should be grasped after object 2, and object 1 and object 2 have no relationship. In the context of a manipulation relationship tree, objects are represented as nodes, and parent-child relationships between these nodes dictate the grasping order. Specifically, parent nodes should be grasped after child nodes. For instance, in Fig. 1, we observe that box 2 serves as the parent node for the wrench,

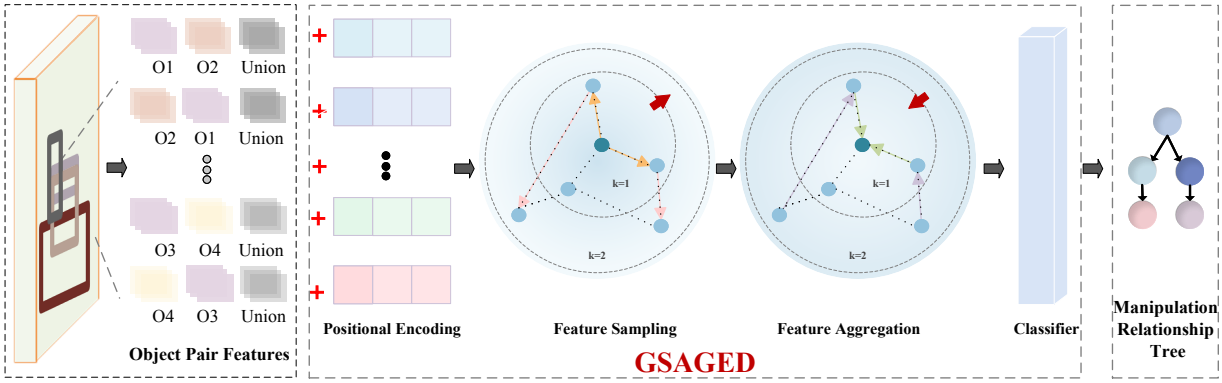

Fig. 2. **The architecture of the proposed algorithm.** And the object pair features embedded in positional encoding are sent to GSAGED. Finally, the manipulation graph is reasoned. We sample features from the inner to the outer layers. After the sampled features are acquired, they are aggregated from the outer to the inner layers to obtain central features.

while the pen is identified as the child node of the wrench. Notably, nodes indicated by the arrows signify objects that should be grasped first. Consequently, the correct sequence for grasping, in this case, should commence with the pen, followed by the wrench, and ending with box 2. Box 1 has no relationship with the other three objects and can be grasped at any time. Given an RGB image I with n objects, we define $O = o_1, o_2, ..., o_n$ to represent all objects in the scene and $R = r_1, r_2, ..., r_{n_{(n-1)}}$ to represent the visual manipulation relationship between objects. For the i-th and j-th objects in the scene, we define them and their relationship as a triple, expressed as (o_i, r_{ij}, o_j) which is distinguished by visual manipulation relationship detection.

B. Overall Architecture

The overall architecture of our model is illustrated in Fig. 1. The model comprises three main components: the feature extractor, the object detector, and the visual manipulation relationship predictor. The model takes RGB images as input, and outputs object detection results and manipulation relationship trees. through the feature exactor to get features. Firstly, input images are passed through the feature extractor to extract image features. Then object categories and bounding boxes in the scene are detected by the object detector. Finally, the object pair features generated by object pair pooling are delivered to the visual manipulation relationship predictor to detect the visual manipulation relationship, as shown in Fig. 2. The contents of each module will be described in detail in the following sections.

C. Object Detector and Object Pair Pooling

Inspired by previous work in visual manipulation relationship detection[2], our model utilizes the Faster R-CNN[22] as the object detector to locate and detect objects. We employ VGG-16 and ResNet-101 as backbones to extract features separately. To generate object pair features, we use object pair pooling[1] before the visual manipulation relationship predictor. For all object pairs in the scene, we make the features a mini-batch including features of two objects and union bounding boxes through object pair pooling.

D. Visual Manipulation Relationship Detection

Firstly, positional encoding is used to strengthen the potential association between relational features and reduce the loss of important local information generated by the sampling function. Then an aggregation function is used to aggregate the neighboring features and optimize the reconstructed features that are sufficient to reflect the potential association between the relational features. Finally, optimized features are put into the classifier to classify the relations.

Graph Sampling Aggregation Network: The feature sampling process is conducted from the inner to the outer layers, employing uniformly sampled neighboring features. The list of sampled nodes in each layer is concatenated according to depth order K. The sampling function and related hyperparameters, including the number of samples, are determined based on the inverse order of k. The visual illustration of the GSAGED's sample and aggregate approach is shown in Fig. 2. The result of the sampled function is subsequently used in the aggregation process.

For each node, a specific number of neighboring nodes are selected as the sampling set. This number can either be fixed or dynamically adjusted based on the node's degree, with the maximum determined by a threshold value u. The feature vectors of the current node and its neighboring nodes are concatenated to form a local graph for each node in the sampled set. The size of this local graph depends on the number of nodes in the sampled set and the dimensionality of the node feature vectors. The embedding vector of the current node is obtained by performing graph convolution on the local graph. This process typically involves employing a multilayer graph convolution network to process the local graph and extract the embedding vector for each node. Once the sampled features are acquired, they are aggregated from the outer to the inner layers to obtain central features. The aggregated neighboring features are concatenated with the features from the upper layer of the central node. This combined feature set is then input into a single-layer MLP (Multi-Layer Perceptron) to obtain new feature vectors. Finally, the features are normalized.

To ensure the functionality of the aggregation function,

it must be differentiable and adaptive to the number of aggregation nodes. This adaptability implies that the resulting feature vector after aggregation should maintain a consistent dimensionality, regardless of variations in the number of nodes during aggregation. Additionally, it should exhibit alignment invariance, indicating that the order of feature vectors after aggregation is independent of the input order of features. This can be represented by the following equation.

$$Aggregate(v_1, v_2) = Aggregate(v_2, v_1) \quad (1)$$

Initially applying nonlinear transformations to the preceding layer, followed by maximizing pooling of the resultant output. The pooling aggregation function can be designed as shown in the following formula.

$$AGGREGATE_k^{pool} = max(\sigma(W_{pool} h_{u^i}^k + b), \forall u^i \in N(v)) \quad (2)$$

Relationship Reasoning: The relationship classifier consists of three linear layers, which determine the category of object pairs. It takes the reconstructed features as input and outputs the relational results. We employ the multi-class cross-entropy function as the loss function for manipulation relationship prediction. This function compares the predicted results with the actual ground truth, facilitating backpropagation for optimization during training, and completing the algorithm's detection process.

IV. Experiment

A. Training Details

The dataset VMRD contains 31 objects categories, 5185 images, 17688 steerable objects, and 51530 visual manipulation relationships. The maximum number of objects in VMRD is 5. The dataset REGRAD contains 38 objects categories, 9 view images, and 17610 models. The maximum number of objects in REGRAD is 19.

We train and test our model on dataset VMRD[1] and REGRAD[8] respectively. Our model uses PyTorch as the framework of the deep learning algorithm and utilizes the NVIDIA GeForce RTX 3090 with 22GB memory. The mini-batch size is set to 2 and 1 on the datasets VMRD and REGRAD. The optimizer utilizes the stochastic gradient descent (SGD) with the momentum parameter set to 0.9. For feature extraction, pre-trained VGG16 and ResNet101 models are used as the training backbone network. In the training process, 20 epoch iterations were used, the training image batch size was 1, the initial learning rate was 1e-3, and the learning decay rate was set to 1e-5. The testing process takes the same hardware configuration. In the real-world manipulation task, objects are set up in random cluttered scenes within 10 step-by-step tests, and the success rate is scaled by ten scene-clearing grabs, and complete success is used as the judgment criterion.

B. Performance Metrics

The performance of the experiments is judged by following metrics:

mAP: Mean average precision is used to compute the performance of object detection algorithms. mAP is the average of all categories of Average Precision (AP) which calculates the average precision for the value of recall increasing from 0 to 1 for each class. **OR:** Object triplet Recall is used to compute the recall on object pairs. The result is true when the category and the location of both objects, as well as the predicted manipulation relationship, are correctly detected. **OP:** Object triplet Precision. OP computes the average precision of manipulation relationships based on object triplets consisting of two objects and their manipulation relationship. **IA:** Image-Wise Triplet Accuracy. This metric calculates the accuracy based on the whole image. Only when all possible triplets in the scene are predicted correctly, the image is considered correct.

C. Experiment Results

The experimental results on the existing grasping manipulation relationship datasets VMRD and REGRAD, comparing the baseline algorithm and the best algorithm in the first stage, show the performance of our model. There is an experimental performance comparison between the algorithms, an ablation experiment, and an evaluation of the performance of the relationship detection in real-world manipulation tasks shown in the following tables.

Comparative Results on Dataset VMRD: The experimental performance of the algorithm in this chapter on VMRD is shown in Table 1. For a fair comparison, experiments were conducted on the same object detection network Faster-RCNN and backbone network VGG16 and ResNet101. The final experimental results show that the detection performance of our model is significantly improved compared with the baseline algorithm VMRN, and the detection time of our model is significantly reduced compared with the best detection algorithm GGNN, which meets the performance requirements of real-time detection.

Table 2 and Table 3 evaluate the image-wise triplet accuracy in different object number scenes with different backbone networks. In Table 2 with the backbone extraction network VGG16, compared with other algorithms for detecting 2-5 objects in the VMRD dataset, our model improves 7.60% in total performance compared with the baseline algorithm VMRN and 0.54% compared with the latest excellent algorithm GGNN-VMRN. Among the IA of 2-5 objects, the IA of detecting four objects in the scene is respectively improved compared with the baseline. The better overall performance reflects the robustness of our model for complex scenes. Table 3 shows the image-wise triplet accuracy with the backbone network ResNet101. The overall performance metrics of our model are improved significantly than that of other methods, which reflects that our model uses object relationship features efficiently and the detection performance is affected little by complex scenes.

Comparative Results on Dataset REGRAD: The results in Table 4 demonstrate that under the same conditions, our model can effectively extract and optimize the features in object relationships, ultimately improving the accuracy of

TABLE I

PERFORMANCE OF GRASP MANIPULATION RELATIONSHIP DETECTION BASED ON DATASET VMRD

Author	Algorithm	Backbone	mAP	OR	OP	IA	Time(ms)
Zhang et al.	VMRN[1](Baseline)	VGG16	95.20	86.30	88.80	68.40	14
		ResNet101	95.40	85.40	85.50	65.80	10
Yang et al.	EVMRN-V[2]	VGG16	95.70	88.56	88.21	73.56	146
		ResNet101	96.40	88.95	86.03	71.56	132
Zuo et al.	GVMRN-RF[3]	VGG16	95.40	88.70	89.50	70.20	-
		ResNet101	94.60	86.90	87.50	68.80	-
Ding et al.	GGNN-VMRN[4]	VGG16	96.30	89.64	88.00	75.56	137
		ResNet101	**96.40**	90.09	88.01	75.33	130
ours	GSAGED	VGG16	96.30	89.4	**90.2**	**76.1**	**124**
		ResNet101	**96.40**	**91.2**	89.3	**75.6**	**112**

TABLE II

IMAGE-WISE TRIPLET ACCURACY IN SCENES WITH DIFFERENT OBJECT NUMBERS (VGG16)

Algorithm	Object Number				
	Total	Two	Three	Four	Five
VMRN[1](Baseline)	68.40	86.15	61.72	62.26	64.29
EVMRN-V[2]	73.56	86.15	66.03	66.04	74.33
GVMRN-RF[3]	70.2	**92.90**	70.30	63.80	60.30
GGNN-VMRN[4]	75.56	86.15	73.21	69.81	82.86
GSAGED(Ours)	**76.1**	91.25	**74.31**	**72.10**	**84.37**

TABLE III

IMAGE-WISE TRIPLET ACCURACY IN SCENES WITH DIFFERENT OBJECT NUMBERS (RESNET101)

Algorithm	Object Number				
	Total	Two	Three	Four	Five
VMRN[1](Baseline)	65.8	80.00	58.37	47.17	54.29
EVMRN-V[2]	71.56	81.54	61.24	52.83	65.71
GVMRN-RF[3]	68.8	91.40	69.20	61.20	57.50
GGNN-VMRN[4]	75.33	92.31	75.12	66.98	72.86
GSAGED(Ours)	**75.6**	**93.62**	**76.84**	**71.61**	**74.3**

relationship detection. Compared with the baseline VMRN, our model improves the image accuracy by 6.7%. Compared with the GGNN-VMRN, the IA metric is 4.3% higher for grasping manipulation relationship detection. Moreover, the equivalent detection time is substantially less than the optimal algorithm at this stage, which makes it more certain that our model meets the real-time requirements of real-world scene tasks. To verify the IA of our model with different-number objects in the scenes, experimental data are provided as shown in Table 5. The data in Table 5 demonstrate that the IA in more complex scenes is significantly improved than the baseline and the optimal algorithm. The reason is that the sampling and aggregation of relational features can effectively reduce the redundancy of feature information and increase the proportion of valid information, which ultimately affects the classification results of the grasp manipulation relationship. Futhermore, compared with the class distrubution of VMRD (None:Parent:Child=2:1:1), the class distrubution of REGRAD (None:Parent:Child=27:1:1) is more imbalanced. The imbalanced distrubution of features in REGRAD is even more pronounced, leading to a more substantial improvement.

TABLE IV

PERFORMANCE OF GRASP MANIPULATION RELATIONSHIP DETECTION BASED ON DATASET REGRAD

Algorithm	mAP	OR	OP	IA	Time(ms)
VMRN[1]	93.3	93.53	93.53	16.8	74
GGNN-VMRN[4]	93.6	94.3	94.3	19.2	162
GSAGED(Ours)	94.8	95.4	95.4	**23.5**	**125**

TABLE V

IMAGE-WISE TRIPLET ACCURACY IN SCENES WITH DIFFERENT OBJECT NUMBERS BASED ON DATASET REGRAD

Algorithm	Object Number				
	Two	Three	Four	Five	Six
VMRN[1](Baseline)	100	98	46.4	45.7	42.1
GGNN-VMRN[4]	100	100	53.4	51.8	50.3
GSAGED(Ours)	100	100	**64.7**	**59.6**	**61.2**
Algorithm	Object Number				
	Seven	Eight	Nine	Ten	Eleven
VMRN[1](Baseline)	20.6	9.8	5.8	3.4	3.2
GGNN-VMRN[4]	32.5	12.3	8.5	6.7	4.1
GSAGED(Ours)	**37.2**	**20.2**	**13.4**	**11.6**	**7.6**
Algorithm	Object Number				
	Twelve	Thirteen	Fourteen	Fifteen	Sixteen
VMRN[1](Baseline)	2.4	2.1	0.1	0	0
GGNN-VMRN[4]	3.7	1.8	0.6	0	0
GSAGED(Ours)	**5.3**	**3.8**	**1.7**	**0.6**	0

To better analyze the strengths of our model, four visualization results of the grasp manipulation relationship tree are shown in Fig. 3. The labels of grasp manipulation relationship can be divided into the correct relationship, redundant relationship, and incorrect relationship. The redundant relationship is defined as two unrelated objects are detected concerning each other which is attributed to the close physical location, giving the neural network the illusion of feature association. Most of the incorrect relationships are due to the visual misalignment of the images, which causes incorrect relationship classification.

As shown in the Fig. 3, VMRN generates more error detection cases because it simply uses the feature relationship to build the grasping operation relationship tree. GGNN-VMRN enhances the object relationship features using contextual information, but its huge computation and incorrect

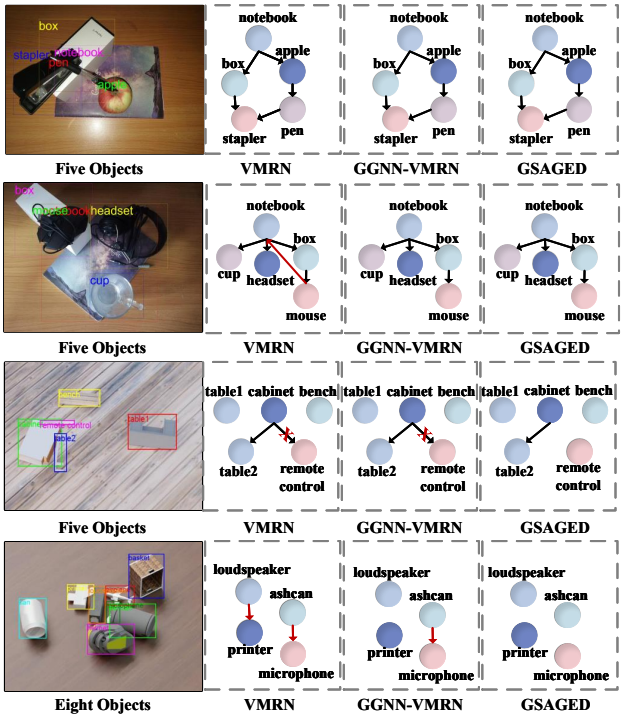

Fig. 3. **Visualization results of grasp manipulation relationship tree.** In the visualization results of the grasp manipulation relationship trees, it becomes evident that VMRN generates a higher number of redundant and inaccurate relationships. Meanwhile, GGNN-VMRN struggles to identify incorrect relationships effectively. In contrast, our model consistently produces a greater number of correct and precise detection results.

TABLE VI

SUCCESS RATE OF ROBOT SAFE GRASPING ON COMPLEX SCENES

Algorithm	Dataset	Two	Three	Four	Five
VMRN[1]	VGG16	10/10	6/10	6/10	6/10
	ResNet101	8/10	6/10	6/10	5/10
EVMRN-N[2]	VGG16	10/10	9/10	8/10	8/10
	ResNet101	9/10	7/10	8/10	7/10
GGNN-VMRN[4]	VGG16	10/10	10/10	10/10	9/10
	ResNet101	10/10	10/10	9/10	8/10
GSAGED(Ours)	VGG16	**10/10**	**10/10**	**10/10**	**10/10**
	ResNet101	**10/10**	**10/10**	**10/10**	9/10
Algorithm	Dataset	Six	Seven	Eight	Nine
VMRN[1]	VGG16	5/10	5/10	4/10	3/10
	ResNet101	4/10	4/10	3/10	2/10
EVMRN-N[2]	VGG16	7/10	6/10	6/10	4/10
	ResNet101	6/10	5/10	5/10	3/10
GGNN-VMRN[4]	VGG16	8/10	7/10	7/10	5/10
	ResNet101	7/10	6/10	6/10	4/10
GSAGED(Ours)	VGG16	**9/10**	**7/10**	**7/10**	**6/10**
	ResNet101	8/10	7/10	6/10	4/10

identification of the visual misalignment are still problems. Our model is able to optimize the focus on some important features by sampling key features after location coding, so as to obtain more correct detection results.

Comparative Results on Robot Grasping System: The robot grasping task comparison experiment is deployed according to the robot's real-world grasping process. One experiment is considered successful unless all objects in the scene are captured in the correct order. Table 6 shows the cumulative number of successes for ten grabs when the number of objects in the real-world scene is increased from 2 to 9 in sequence, which demonstrates that our model shows better performance in detecting the relationship of multi-objective grasping manipulation when extending to real-world scenes. Fig. 4 shows the robot environment and a stacking scene as well as an example of the robot grasping process instructed by the manipulation relationship tree.

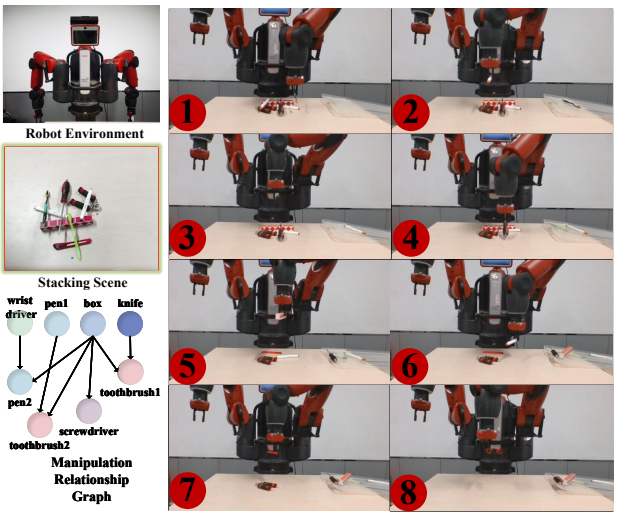

Fig. 4. **Example of robot sequential grasping.** This picture shows the step-by-step grasping process of the robot. The manipulation relationship graphs are generated in the stacking scene of 3 and 9 objects. It is worth noting that the robot first grasps the objects represented by the child nodes in the picture. In the image, the robot sequentially grasped the following objects: toothbrush 2, pen 2, toothbrush 1, screwdriver, box, pen 1, utility knife, and waist driver.

V. CONCLUSION

In this paper, we introduce a grasp manipulation relationship detection algorithm based on GraphSAGE. This algorithm aids robots in identifying targets in complex scenes and determining the optimal grasping order. We have trained our model on the VMRD and REGRAD datasets. By incorporating positional encoding and sampling aggregation functions, our model efficiently integrates global and local information, relieving imbalanced distribution of features in the context of complex scenes. Furthermore, our real-world grasping experiments demonstrate that our model outperforms existing approaches in practical tasks. In future research, we aim to explore methods for enabling autonomous scene exploration by the robot, which can assist in selecting the most suitable detection viewpoint or incorporating multi-view images as input.

ACKNOWLEDGMENT

This work was supported in part by National Key R&D Program of China under grant No. 2021ZD0112700, NSFC under grant No.62125305, No. U23A20339, No.62088102, No. 62203348.

REFERENCES

[1] H. Zhang, X. Lan, X. Zhou, Z. Tian, Y. Zhang, and N. Zheng, "Visual manipulation relationship recognition in object-stacking scenes," *Pattern Recognition Letters*, vol. 140, pp. 34–42, 2020.

[2] C. Yang, X. Lan, H. Zhang, X. Zhou, and N. Zheng, "Visual manipulation relationship detection with fully connected crfs for autonomous robotic grasp," in *2018 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2018, pp. 393–400.

[3] G. Zuo, J. Tong, H. Liu, W. Chen, and J. Li, "Graph-based visual manipulation relationship reasoning network for robotic grasping," *Frontiers in Neurorobotics*, vol. 15, p. 719731, 2021.

[4] M. Ding, Y. Liu, C. Yang, and X. Lan, "Visual manipulation relationship detection based on gated graph neural network for robotic grasping," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1404–1410.

[5] J. Mahler, J. Liang, S. Niyaz, M. Laskey, R. Doan, X. Liu, J. A. Ojea, and K. Goldberg, "Dex-net 2.0: Deep learning to plan robust grasps with synthetic point clouds and analytic grasp metrics," *arXiv preprint arXiv:1703.09312*, 2017.

[6] K. Fang, Y. Zhu, A. Garg, A. Kurenkov, V. Mehta, L. Fei-Fei, and S. Savarese, "Learning task-oriented grasping for tool manipulation from simulated self-supervision," *The International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 202–216, 2020.

[7] H. Zhang, X. Lan, X. Zhou, Z. Tian, Y. Zhang, and N. Zheng, "Visual manipulation relationship network for autonomous robotics," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*. IEEE, 2018, pp. 118–125.

[8] H. Zhang, D. Yang, H. Wang, B. Zhao, X. Lan, J. Ding, and N. Zheng, "Regrad: A large-scale relational grasp dataset for safe and object-specific robotic grasping in clutter," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2929–2936, 2022.

[9] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 729–734.

[10] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: going beyond euclidean data," *IEEE Signal Processing Magazine*, vol. 34, no. 4, pp. 18–42, 2017.

[11] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun, "Spectral networks and locally connected networks on graphs," 12 2013.

[12] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[13] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, Eds., vol. 28. Curran Associates, Inc., 2015.

[14] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *CoRR*, vol. abs/1609.02907, 2016.

[15] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2014–2023.

[16] W. L. Hamilton, R. Ying, and J. Leskovec, "Representation learning on graphs: Methods and applications," *arXiv preprint arXiv:1709.05584*, 2017.

[17] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[18] Y. Chen, M. Rohrbach, Z. Yan, Y. Shuicheng, J. Feng, and Y. Kalantidis, "Graph-based global reasoning networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 433–442.

[19] J. B. Lee, R. A. Rossi, S. Kim, N. K. Ahmed, and E. Koh, "Attention models in graphs: A survey," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 6, pp. 1–25, 2019.

[20] T. N. Kipf and M. Welling, "Variational graph auto-encoders," *arXiv preprint arXiv:1611.07308*, 2016.

[21] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2017.

[22] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.