2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

KOSMOS-E : Learning to Follow Instruction for Robotic Grasping

Zhi Wang[1,2*], Xun Wu[1,2*], Shaohan Huang[1†], Li Dong[1], Wenhui Wang[1], Shuming Ma[1], Furu Wei[1]

***Abstract*— Tuning on instruction-following data has been shown to enhance the capabilities and controllability of language models, but the idea is less explored in the robotic field. In this work, we introduce KOSMOS-E, a Multimodal Large Language Model (MLLM) that leverages instruction-following robotic grasping data to enhance capabilities for precise and intricate robotic grasping maneuvers. To achieve this, we craft a large-scale instruction-following robotic grasping dataset, termed INSTRUCT-GRASP, primarily comprising two aspects: (i) grasp a single object following varying levels of granularity descriptions, e.g., different angles and aspects, and (ii) grasp a specific object within a multi-object environment following specific attributes, e.g., color and shape. Extensive experiments show the effectiveness of KOSMOS-E on robotic grasping tasks across a variety of environments.**

I. INTRODUCTION

The field of large pretrained models, e.g., Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) has witnessed remarkable progress on various tasks [1], [2], [3], [4], e.g., reasoning [5], [6], planning [7], [8], code generation [9], [10] and open-vocabulary visual interactions [11], [12], [13]. Such capabilities would be tremendously useful for generalist robots that must perform a variety of tasks in real-world environments. Recently, large pretrained models are introduced to enhance performance across various robotic tasks, e.g., robotic manipulation [14], [15], [16], reasoning [17], [18] and grasping [19].

Concurrently, as instruction-tuning has been validated as an effective technique to enhance the capabilities and controllability of large pretrained models [20], [21], attempts have been made to leverage the instruction-following ability of large pre-trained models for robotic tasks and showed promising results [14], [17]. Focusing on the robotic grasping task, the prior work, RT-Grasp [22] training MLLMs on instruction data to predicting adaptable numerical grasping informed by reasoned strategies, thus efficiently utilize the reasoning capability of MLLMs and show a better grasping performance than non-instruction MLLMs.

Despite the progress, there remains a question about how to use instruction data to enable embodied agent to physically grasp in the complex real world, and the capability of embodied vision-language models to devise executable programs is still largely uncharted territory. To empower robotic grasping models with instruction-following capabilities, we first leveraged GPT-4 Vision to construct a large-scale robotic grasping instruction dataset, termed as INSTRUCT-GRASP. This dataset encompasses a diverse array of visual scenes paired with complex, fine-grained instructions, meticulously tailored to train and assess the reasoning and grasping capabilities of robotic systems in cluttered environments. Specifically, as shown in Figure 2, we design different fine-grained instruction to handle different scenes.

Based on INSTRUCT-GRASP, we introduce KOSMOS-E, a MLLM training on INSTRUCT-GRASP that shows a better grasping ability in complex environment. Figure 1 illustrates how KOSMOS-E integrates an agent's visual perspective to devise precise robot action (grasping point and rotation angle) and yield accurate grasping in complex environment by following textual input instructions. To validate the efficacy of KOSMOS-E, we construct expensive experiments on the grasp benchmarks and show that KOSMOS-E not only achieves better performance compared to state-of-the-art (SOTA) robotic grasping models, but also enable significant improvements to generalization over objects, scenes, and exhibit a breadth of emergent instruction-following abilities.

In sum, our key contributions include:

- We construct INSTRUCT-GRASP, a large-scale fine-grained robotic grasping instruction dataset, which contains multiple kinds of instruction for both single and multi object scenes.
- We propose KOSMOS-E, a multimodal large language model training on INSTRUCT-GRASP, demonstrating a strong instruction-following ability acting as generalizable and semantically aware robotic grasping policies.
- We conduct extensive experiments on multiple benchmarks and show that our proposed KOSMOS-E trained with INSTRUCT-GRASP achieves competitive or superior performance compared to SOTA robotic grasping models.

II. CONSTRUCTION OF INSTRUCT-GRASP

The community has witnessed a surge in the amount of public grasping data, such as Cornell grasping dataset [23] and Jacquard grasping dataset [24]. However, the available volume of instruction-following grasping data is limited, partially because the process for creating such data is time-consuming and not well-defined. To address this issue, we introduce INSTRUCT-GRASP, a large-scale robotic grasping instruction dataset, which is created based on Cornell grasping dataset [23]. Specifically, inspired by the success of recent GPT models in text-annotation tasks, we utilize GPT-4 Vision for grasping instruction-following data collection. The construction pipeline is shown in Figure 2.

[1] Microsoft Research Asia, Beijing, China.
[2] Tsinghua Univeristy, Beijing, China.
* The first two authors contributed equally to this work. † Corresponding author.

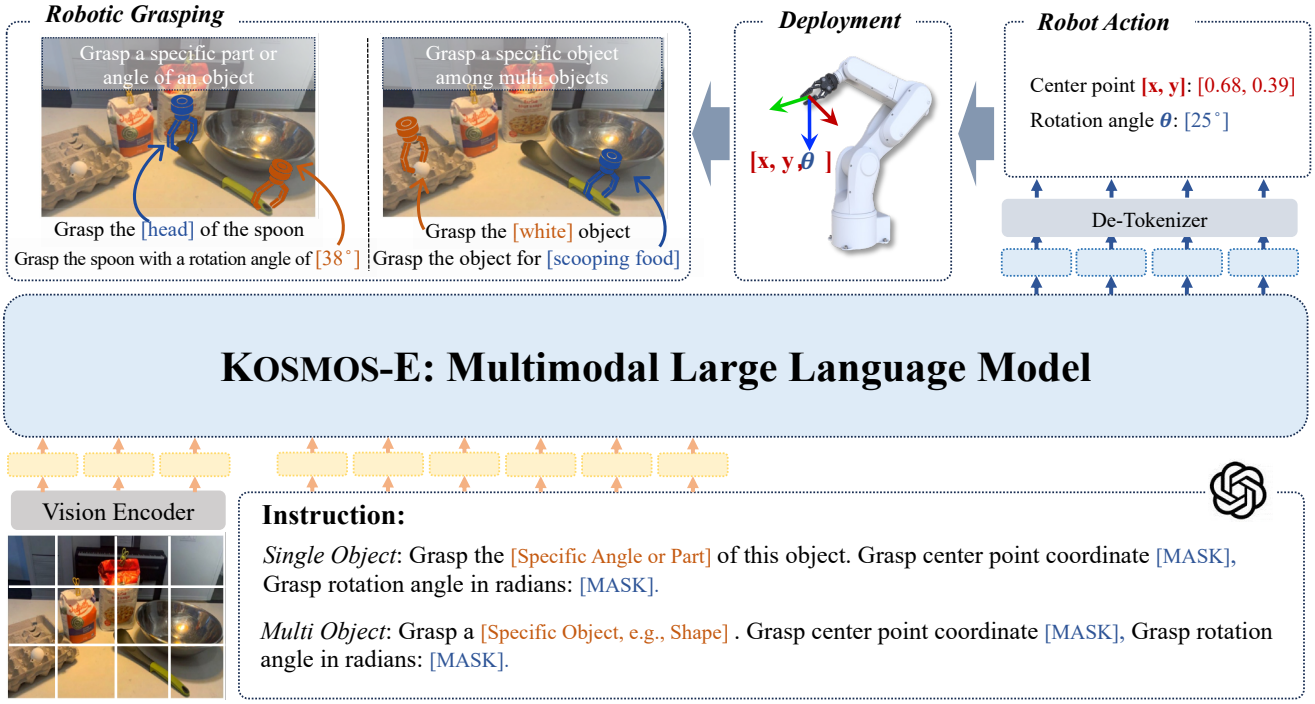


Fig. 1: KOSMOS-E is a multimodal large language model that has new capabilities of robotic grasping, which can understand multimodal input and follow different instructions to generate a numerical grasp pose prediction (grasp center point $[x, y]$ and rotation angle θ), guiding the robot to accurately grasp in both single object and multi-object scenes.

A. Data Augmentation

The Cornell grasping dataset [23] consists of 885 RGB-D images (640×480 px) of 240 distinct objects, with 5,110 human-labelled positive and 2,909 negative grasps. Given the relatively limited number of data, we performed extensive data augmentation. Firstly, the images are center cropped to obtain a 351×351 region. Secondly, the cropped image is randomly rotated between 0 to 360 degree. Thirdly, the rotated image is randomly translated in x and y direction by up to 40 pixels. Finally, the translated image is cropped to 224×224 in size. Consequently, we have INSTRUCT-GRASP-NON dataset, with 250k grasp examples. And only positively labeled grasps from the dataset were used.

B. Grasp Instruction Generation

After data augmentation, we generate detailed instruction for each objection and grasp. We categorize the instructions into three types: (1) *Object-wise*, (2) *Image-wise*, and (3) *Grasp-wise*.

Object-wise Instruction. To begin with, we manually categorized the objects into 75 distinct categories based on their inherent characteristics. This categorization allows us to establish a systematic framework for generating instructions tailored to each category. We employ the instruction "Grasp the [object name]" for each object to obtain the INSTRUCTION-NAME component.

To generate more object-wise instructions, we employ the advanced language model GPT-4 Vision, leveraging its powerful natural language processing capabilities. By utilizing GPT-4 Vision, we are able to generate two crucial components for each object category: INSTRUCTION-SHAPE and INSTRUCTION-PURPOSE.

- The INSTRUCTION-SHAPE component describes the geometric attributes of the objects, encompassing shape-adjectives such as "cylindrical", "rectangular", "round", and "flat". This information provides the robotic system with a precise understanding of the object's physical properties.
- The INSTRUCTION-PURPOSE component elucidates the real-world significance and function of the objects. By conveying the meaning behind the objects, the instructions facilitate a higher level of comprehension for the robot, enabling it to better interpret human intentions and adapt its actions accordingly.

Drawing inspiration from the work of RT-Grasp [22], we further develop the INSTRUCTION-STRATEGY component, which outlines effective grasping techniques specific to each object category, based on human knowledge and expertise. These strategies serve as valuable guidance for the robotic system, ensuring efficient and successful grasping operations.

Image-wise Instruction. For each image in our dataset, we employ GPT-4 Vision to generate the INSTRUCTION-COLOR component. The INSTRUCTION-COLOR component serves to illustrate the color of the object depicted in the image. We utilize color-adjectives such as "blue", "orange", and "beige" to accurately describe the object's color. In cases where the object exhibits multiple colors, we strive to provide comprehensive descriptions by using multiple color-adjectives or phrases such as 'with [color] details'. This approach

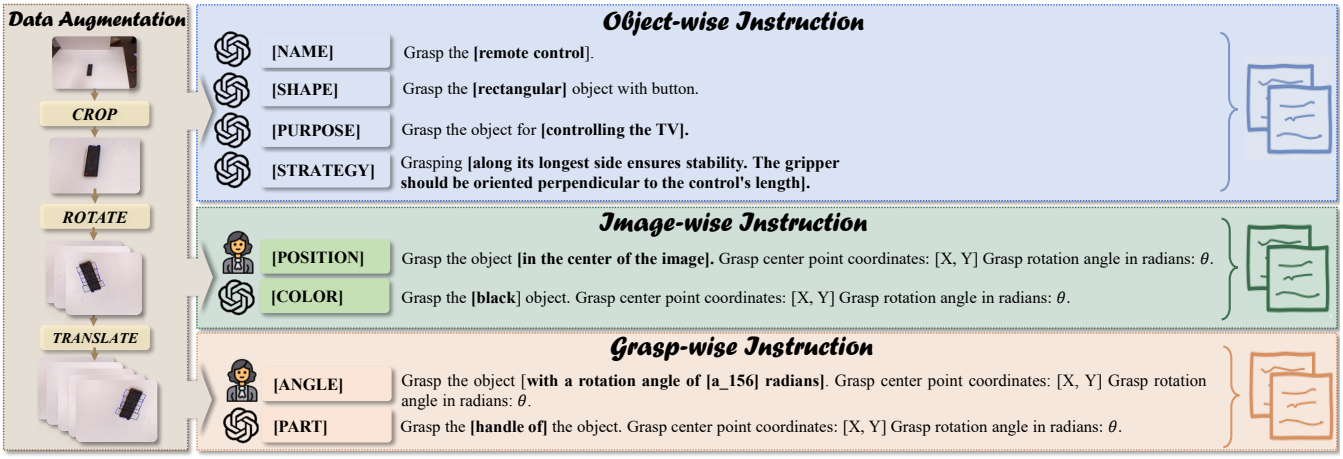


Fig. 2: The pipeline of constructing INSTRUCT-GRASP. Here we designate (👩) as human while the (🤖) as the GPT-4 Vision.

Instruction for Single Object

Angle: "Grasp the object with a rotation angle of [A ENCODED] radians. Grasp center point coordinates: [X, Y]."

Part (GPT-4 Vision): "Grasp the handle of the brush. Grasp center point coordinates: [X, Y] Grasp rotation angle in radians: [A ENCODED]."

Fig. 3: The input instruction for single object grasping scene.

ensures that the instructions encompass all relevant color information, enabling the robotic system to perceive and differentiate objects based on their color attributes.

Additionally, we manually generate the INSTRUCTION-POSITION component based on the spatial location of the object within the image. By dividing the image into nine sections—using terms such as "middle", "top", "bottom", "left", and "right", we establish a framework for generating instructions that convey the object's precise location.

Grasp-wise Instruction. For each grasp in every image, we still harness GPT-4 Vision to generate the Instruction-Part component. The INSTRUCTION-PART component illustrates which specific part of the object should be grasped. By generating these detailed instructions, we enable the robotic system not only to grasp the object as a whole but also to target and manipulate specific parts of the object with accuracy and dexterity. This capability enhances the system's overall grasping versatility, allowing it to perform complex tasks that require interaction with specific object components.

Additionally, we create the INSTRUCTION-ANGLE component, which is formatted as "Grasp the object with a rotation angle of xxx radians." These instructions enable us to incorporate the functionality of specifying a precise angle before initiating the grasping action. By providing explicit guidance on the desired rotation angle, we empower the robotic system to perform grasping maneuvers with enhanced precision and control. The corresponding detailed prompt templates for both INSTRUCTION-PART and INSTRUCTION-ANGLE construction can be found in Figure 3.

Multi-object Instruction. After completing the aforementioned steps, we obtain the INSTRUCT-GRASP-SINGLE dataset. To expand the scope and complexity of the dataset, we employ a random selection process. For each image in the dataset, we randomly select three additional images. We then combine these four images, positioning them at the four corners of a larger composite image, which has dimensions of 448 $\times$ 448 pixels. Subsequently, we resize the combined image to a standardized size of 224 $\times$ 224 pixels, while keeping the instructions for each individual image unchanged. Notably, we adapt the INSTRUCTION-POSITION component to align with the revised layout of the composite image. By following this process, we obtain the INSTRUCT-GRASP-MULTI dataset. This expanded dataset presents a more challenging and realistic scenario for the robotic system, as it involves multiple objects placed in various locations within a composite image. The dataset facilitates the development and evaluation of algorithms and models capable of handling complex multi-object grasping tasks. The corresponding detailed prompt templates for constructing multi-object instruction can be found in Figure 4.

Finally, we create the INSTRUCT-GRASP dataset, which comprises the INSTRUCT-GRASP-NON dataset, the INSTRUCT-GRASP-SINGLE dataset, and the INSTRUCT-GRASP-MULTI dataset. This comprehensive dataset encompasses a total of 250k unique language-image non-instruction samples and 1.56 million instruction-following samples. Among these instruction-following samples, 654k pertain to the single-object scene, while the remaining 654k relate to the multi-object scene.

III. INSTRUCT-FOLLOWING TUNING FOR ROBOTIC GRASPING

KOSMOS-E is a multimodal large language model integrated with strong instruction-following abilities for grasping in complex environments. The model can take instruction data as input and predict the grasp actions.

Instruction for Multi Object

Name: "Grasp the [NAME]. Grasp center point coordinates: [X, Y]."

Color: "Grasp the [COLOR] object. Grasp center point coordinates: [X, Y] Grasp rotation angle in radians: [A ENCODED]."

Shape: "Grasp the [SHAPE] object with buttons. Grasp center point coordinates: [X, Y] Grasp rotation angle in radians: [A ENCODED]."

Purpose: "Grasp the object for [PURPOSE]. Grasp center point coordinates: [X, Y] Grasp rotation angle in radians: [A ENCODED]."

Position: "Grasp the object [POSITION] in the section of this image. Grasp center point coordinates: [X, Y] Grasp rotation angle in radians: [A ENCODED]."

Strategy: "It is rectangular and has a flat side with buttons. Grasping along its longest side ensures stability. The gripper should be oriented perpendicular to the control's length. Grasp center point coordinates: [X, Y] Grasp rotation angle in radians: [A ENCODED]."

Fig. 4: The input instruction for multi objects grasping scene.

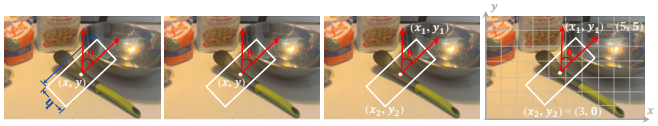
A. Problem Formulation

The problem of robotic grasp detection involves finding a successful grasp representation for a given object image. As shown in Figure 5 (a), a commonly adopted grasp representation in previous works [23] is XYWHA, which is represented as a five-dimensional entity as $g = f(x, y, h, w, \theta)$. Here, (x, y) represents the grasp center in image coordinates, w signifies the distance between parallel plates, h denotes the height of parallel plates, and θ represents the gripper's rotation angle relative to the horizontal axis.

In this work, following RT-Grasp [22], we assume that w corresponds to the maximum width of the gripper and h corresponds to the height of parallel plates for a specific robot, and thus adopting the representation named XYA , which is shown in Figure 5 (b)

In this work, which primarily focuses on assessing the effectiveness of MLLMs in robotic grasping tasks and exploring the role of grasping instructions, we make the assumption that w corresponds to the maximum width of the gripper, while h corresponds to the height of parallel plates for a specific robot in the real-world scenario, and finally adopt XYA shown in Figure 5 (b) as the representation format of output grasp information for KOSMOS-E.

Additionally, we applied a linear encoding to map the original rotation angle from the interval of [-$\frac{\pi}{2}$, $\frac{\pi}{2}$] to the range of $[0, 255]$ as the input. Subsequently, we will decode the predicted encoded value back to radians as the output. The ultimate grasp representation is denoted as $g = (x, y, \theta)$, where (x, y) corresponds to the pixel position of the center point in the image, and the rotation angle θ is expressed in encoded radians, ranging from $[0, 255]$, as illustrated in Figure 5. Moreover, we divided the grasp data into two separate components: one related to the center point and the other related to the rotation angle, which doubled the number of image-text pairs.



(a) XYWHA (b) XYA (c) TLBRA (d) TLBRA†

Fig. 5: **Illustration of different grasp representations**. In our KOSMOS-E, we adopt XYA as the grasp information representations.

Furthermore, we have conducted a thorough analysis of the performance differences between two types of grasp representations, namely "XYWHA" and "XYA". Additionally, we have investigated the impact of encoding the rotation angle and splitting the grasp data into two components. For detailed comparisons, please refer to Section 4.3.

B. Robotic Grasping Multimodal Model

KOSMOS-E is based-on a transformer causal language model and trained through the autoregressive language modeling task. Following the architecture of KOSMOS-1 [3], the training loss only considers discrete tokens, such as text and grasp tokens. As shown in Figure 1, the input image is encoded by a vision encoder, such as CLIP [25], and take the output of the last layer as the image representation (image embedding). The image representation is then combined with the text tokens and grasp tokens, and fed into the multimodal transformer model. Taking Figure 1 as an example, the input of the multimodal transformer model is:

<s><image> Image Embedding </image>Grasp the [Specific Angle or Part] of this object. Grasp center point coordinate: [MASK], Grasp rotation angle in radians: [MASK] </s>.

The model is trained to predict the next token in the sequence given the previous tokens. The training loss is computed based on the cross-entropy loss between the predicted token and the ground truth token.

C. Training

KOSMOS-E is trained upon INSTRUCT-GRASP dataset. The training procedure involves a batch size of 2,048 tokens and 30,000 training steps, utilizing approximately 11.8 billion tokens. We use an AdamW optimizer with $\beta = (0.9, 0.98)$, a weight decay of 0.01, and a dropout rate of 0.1. The learning rate increases to 2e-4 during the first 375 warm-up steps and linearly decays to zero. The model is trained on one DGX2 node for 12 hours.

KOSMOS-E uses the weights of KOSMOS-2 for initialization, and follows it architecture settings. The vision encoder has 24 layers with 1,024 hidden size and 4,096 FFN intermediate size. The multimodal large language model component is a 24-layer MAGNETO Transformer [26] with 2,048 hidden dimensions, 32 attention heads, and 8,192 FFN intermediate size. The total number of trainable parameters amounts to approximately 1.6B. The image resolution is set

to 224×224 and the patch size is 14×14. We update all the parameters during training.

D. Evaluation Metric

In line with previous studies, we adopt a cross-validation methodology and divide the datasets into five folds to evaluate our approach. We employ both image-wise and object-wise splits for comprehensive evaluation.

Image-wise split: The image-wise split involves randomly dividing all the images in the dataset into five folds. This partitioning allows us to assess the generalization ability of the network concerning objects presented in different positions and orientations.

Object-wise split: With the object-wise split, we randomly divide individual object instances, ensuring that all images featuring a particular object are placed in the same validation set. This separation enables us to evaluate the network's generalization performance on previously unseen objects.

For grasp detection, we utilize the rectangle metric [27] to report the system's performance. According to this metric, a grasp is deemed valid if it satisfies the following conditions:

- The difference in grasp orientation between the predicted grasp g_p and the ground truth grasp g_t is less than 30 degrees.
- The intersection over union (IoU) score between the predicted grasp g_p and the ground truth grasp g_t is greater than 25%, where the IoU score is computed as $IoU\left(g_p, g_t\right) = \frac{|g_p \cap g_t|}{|g_p \cup g_t|}$.

To ensure accurate evaluation, the rectangle metric necessitates a grasp representation with the width w and height h of the grasp rectangle, which are not directly provided by our method's grasp prediction. Therefore, we combine the ground truth width w and height h to form the appropriate rectangle representation for evaluation purposes.

IV. Experiments

A. Non-instruction Grasping

Experiment Settings. Following [22], we evaluate the proposed method and two types of comparable baselines: (1) Traditional grasping algorithms GR-ConvNet [28] and GG-CNN2 [29], which utilizing RGB-D images as training input, as well as (2) MLLMs-based grasping algorithms RT-Grasp [22], using the reasoning tuning VLM grasp dataset proposed in [22]. We take grasp prediction accuracy including image-wise and object-wise as our evaluation metric defined in [22]. We follow a cross-validation setup as in previous works and partition the datasets into 5 folds.

Experiment Results. The main results are presented at Table I. In terms of image-wise evaluation, our method outperforms grasp detection algorithms based on MLLMs and the conventional GG-CNN2 approach. Regarding object-wise evaluation, our model exhibits better performance compared to RT-Grasp. It is noteworthy that our method demonstrates significantly lower variance than RT-Grasp, indicating enhanced stability. These outcomes affirm the efficacy of our method in real-world object grasping scenarios.

TABLE I: Results of Non-Instruction Grasping experiments.

Method	Modality	Grasp Accuracy	
		IW	OW
GR-ConvNet [28]	RGBD	97.70	96.60
GG-CNN2 [29]	RGBD	84.00	82.00
RT-Grasp [22] (Numbers Only)	RGB+text	58.44±6.04	50.31±14.34
RT-Grasp [22] (With Prompts)	RGB+text	69.15±11.00	67.44±9.99
KOSMOS-E	RGB+text	85.19±0.27	72.63±4.91

B. Instruction-following Grasping

Experiment Settings. Detailed in Section II, we constructed a novel dataset for instruction-following grasping, encompassing both single-object and multi-object scenarios. Various datasets were utilized to train the baseline models. Specifically, our model was trained using a combination of non-instruction and instruction-following datasets. In contrast, four other baselines were each trained on a distinct dataset: non-instruction, single-object, multi-object, and a combination of single-object and multi-object datasets. We adopted image-wise grasp accuracy as our primary evaluation metric.

Experiment Results. The results of instruction-following are summarized in Table II. Three key insights emerged from the results. First, the multi-object dataset is more challenging compared to the single-object dataset; the highest accuracy achieved by our model on the multi-object dataset is below 40, whereas for the single-object dataset, it exceeds 75. Second, the absence of multi-object scenarios in the training data severely hampers the model's performance on such datasets. This is evident from the significantly reduced accuracy for the non-instruction and single-object instruction baseline models on the multi-object dataset, with some categories registering accuracy below 1. Third, the diverse grasping datasets appears to be mutually beneficial. Our model outperforms in 6 out of 8 categories, suggesting that training on a more varied dataset equips the model to better navigate complex scenarios.

C. Ablation Study

Grasp Representation. We investigated the impact of different representations of grasp poses on the final experimental results. Four distinct representations were explored: (1) XYWHA: This representation utilizes the coordinates of the center point, along with the width and height of the grasp rectangle, to describe a grasp pose. (2) XYA: Similar to XYWHA, but incorporates the ground truth width and height for accurate evaluation. (3) TLBRA: Inspired by the KOSMOS-2 approach, a grasp pose is determined using the coordinates of the **t**op-**l**eft and **b**ottom-**r**ight corners, along with the rotation angle. (4) TLBRA†: Similar to TLBRA, but encodes the two-dimensional point (x, y) into a one-dimensional numerical representation based on the patch number within a 32×32 image grid, more details regarding the encoding can be found in KOSMOS-2 [4]. The corresponding results are presented in Table III. It was observed that employing XYA as the input format generally yielded superior outcomes compared to the other three representations.

Training Data Format. We investigated the effects of different training data formats on model performance. One

TABLE II: Instruction-following Grasping

	Single Object		Multi Object					
	Angle	Part	Name	Color	Shape	Purpose	Position	Strategy
KOSMOS-E	77.98	**82.35**	**31.43**	29.56	**29.49**	**27.93**	**30.44**	**36.16**
- non-instruction	**79.16**	76.80	0.42	4.80	1.48	0.42	7.34	2.47
- single instruction	78.27	80.28	0.49	0.35	0.35	0.46	0.35	0.85
- multi instruction	7.49	8.20	25.99	25.32	24.82	23.87	25.14	27.22
- single & multi instruction	78.02	80.92	30.23	**30.12**	28.46	27.23	29.69	33.58

TABLE III: Comparison of different grasp representations.

Fomat	Accuracy
XYA	**68.74**
XYWHA	65.59
TLBRA	51.16
TLBRA†	51.60

TABLE IV: Impact of different training data format.

Individual Output	Angle Encoding	IW	OW
✓	✓	73.67±1.39	55.11±5.61
✓	✗	67.01±0.18	55.60±7.16
✗	✗	64.87±0.60	52.39±7.90

TABLE V: Different training strategy.

Training Strategy	Angle Encoding	IW	OW
Target Only	✗	68.24	44.70
	✓	71.19	49.91
Full Instruction	✗	68.74	51.10
	✓	71.56	49.97

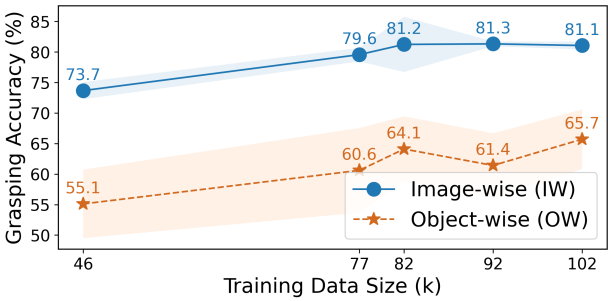


Fig. 6: **KOSMOS-E trained on larger size of training data performs better**. We evaluate KOSMOS-E of varying training data sizes on both image-wise (IW) and object-wise (OW) grasping accuracy, and find that accuracy increases with larger training data size across both wises, indicating the strong training scalability of KOSMOS-E.

aspect of this exploration involved determining whether to output all grasp actions together or to output each attribute individually for each instructional data point. Additionally, we examined the impact of angle encoding variations, specifically comparing the direct use of angles within the range $[-\frac{\pi}{2}, \frac{\pi}{2}]$ against a bucketed encoding transformed to the range [0, 255]. The comparative analysis of these training data formats and angle encoding strategies is presented in Table IV. We can observe that separately output the attributes for each instruction data point and using the bucketed encoding for angles both lead to improved model performance.

Training Strategy. To investigate the effect of training loss format on the performance of the model, comparative experiments are conducted. Table V shows that (1) only target for loss computation and computing loss for all text tokens does not significantly impact the final performance of the model. Calculating loss for all text tokens typically only yields minor performance improvements in most cases. (2) Employing angle encoding format can substantially enhance model performance. For instance, utilizing only target loss resulted in a 5.21 improvement in object-wise performance.

Training Data Size. Our investigation into the data scalability of KOSMOS-E involved augmenting the training dataset size. As detailed in Section II, we employed various rotations and translations of the images, effectively expanding the initial dataset size from 46k to 102k samples. The results, presented in Figure 6, demonstrate that enlarging the training set enhances the downstream performance of KOSMOS-E in both image-wise and object-wise evaluations. Notably, the performance enhancements are more pronounced when the dataset size grows from 46k to 80k samples. Beyond this point, however, the improvements get marginal, indicating that the model has reached its capacity to learn from the data.

V. CASE STUDY

We do case studies of KOSMOS-E for each type of instruction in our proposed INSTRUCT-GRASP. As shown in Figure 7, we find that our KOSMOS-E demonstrates a robust instruction-following capability to comprehend user-provided instructions, thereby facilitating more flexible and effective grasping actions that closely align with real-world application scenarios.

For example, look at Figure 7 (b), KOSMOS-E is adept at understanding the user's intention, as exemplified by the instruction to "Grasp the object for providing elevation to the wearer's feet." It can infer that among the four objects present in the scene, the water cup located in the bottom left corner best meets the user's requirement and proceeds to grasp it. This capability is highly applicable in real-world complex scenarios, i.e., by merely inputting our demand through an instruction, the model can accurately fulfill the grasping task without the need for pre-designed complex and redundant grasping rules. This not only simplifies the grasping process but also enhances the robustness of the operation.

Another example shown in Figure 7 (e), illustrates our KOSMOS-E's ability to grasp specific parts of an object based on the user's instruction, thereby achieving a more stable and effective grasp for subsequent actions. For instance, through an instruction, we can direct the robotic arm to grasp the handle of a spoon (rather than any arbitrary part). This precision enables the correct execution of subsequent operations, such as using the spoon to scoop food.

The strong instruction-following functionality underscores KOSMOS-E's proficiency in interpreting user instructions to facilitate precise and practical manipulations, essential for complex and nuanced tasks.

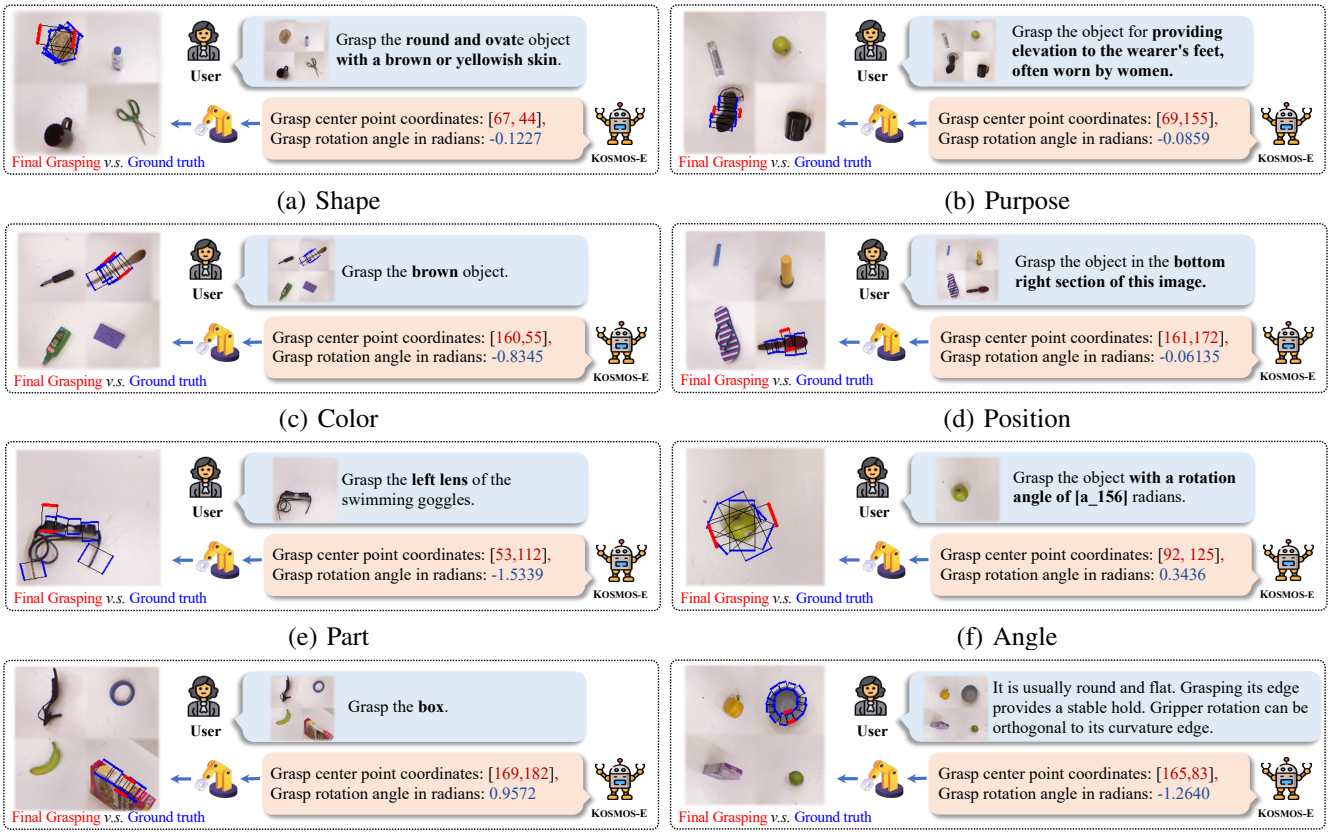


Fig. 7: **Case study for KOSMOS-E**. The examples include all types of instructions in INSTRUCT-GRASP. In each subgraph, red grasping box denotes the "Final Grasping" predicted by KOSMOS-E while blue ones denote the "Ground truth".

VI. RELATED WORK

A. Robotic Grasping

Robotic grasping has been extensively studied in recent years, with two primary approaches: Traditional Robotic Grasping [30], [31] and Learning-based Robotic Grasping [22]. Traditional methods [32], [33], [34] analyze the geometry of the object and gripper to determine a grasping pose but face limitations in real-world applications due to their lack of reasoning capabilities [22]. Learning-based methods [35], [36], [37], which utilize CNN-based architectures to predict grasp poses, achieve accuracy but often struggle to generalize to unseen object categories and are challenging to apply in real-world scenarios.

Recently, Large Language Models (LLMs) have been successfully applied in various robotic tasks, e.g., reasoning [5], [38], planning [39], [40], manipulation [14], [15] as well as grasping [22], [41]. LAN-grasp [41] combines a traditional grasp planner with LLMs to generate grasps, demonstrating a deeper semantic understanding of the objects. RT-Grasp [22] empowers LLMs to generate instructed precise numerical outputs such as grasp poses, breaking the gaps between text-based planning and direct robot control utilizing LLMs. While these works overlook instruction-following in MLLMs, enhancing this capability enables models to manage complex robotic grasping tasks and interpret human commands with higher precision.

B. Language Models for Robotic Manipulation

Language Models are shown to have valuable knowledge for robot manipulation [14], [15], [16] through reasoning and planning. VoxPoser [14] addresses robotic manipulation problems by LLMs and Vision-Language Models (VLMs) to synthesize robot trajectories in free-form natural language instructions, addressing the bottleneck of reliance on predefined motion primitives in existing approaches. RT-2 [19] express the actions as natural text tokens, showing a better generalization and reasoning ability. Similarly, notable works such as EmbodiedGPT [42], SayCan [17], and Palm-E [18] employ multimodal language models trained on robot manipulation data, enabling agents to process visual inputs and execute precise robotic motor control commands.

VII. CONCLUSION

As the use of large pre-trained models in robotics gains traction, exploring their instruction-following capabilities to enhance downstream tasks is emerging as a critical research direction. In this work, we introduce KOSMOS-E, a MLLM specifically trained on proprietary robotic grasping instruction dataset, INSTRUCT-GRASP. KOSMOS-E integrates strong instruction-following abilities and demonstrates improved grasping performance. Extensive experiments validate its efficacy, confirming the benefits of incorporating instruction-following capabilities in robotic grasping tasks.

REFERENCES

[1] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[2] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023.

[3] S. Huang, L. Dong, W. Wang, Y. Hao, S. Singhal, S. Ma, T. Lv, L. Cui, O. K. Mohammed, Q. Liu, K. Aggarwal, Z. Chi, J. Bjorck, V. Chaudhary, S. Som, X. Song, and F. Wei, "Language is not all you need: Aligning perception with language models," *ArXiv*, vol. abs/2302.14045, 2023.

[4] Z. Peng, W. Wang, L. Dong, Y. Hao, S. Huang, S. Ma, and F. Wei, "Kosmos-2: Grounding multimodal large language models to the world," *arXiv preprint arXiv:2306.14824*, 2023.

[5] T. Kojima, S. S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa, "Large language models are zero-shot reasoners," *Advances in neural information processing systems*, vol. 35, pp. 22 199–22 213, 2022.

[6] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," *Advances in Neural Information Processing Systems*, vol. 35, pp. 24 824–24 837, 2022.

[7] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar *et al.*, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[8] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.

[9] B. Roziere, J. Gehring, F. Gloeckle, S. Sootla, I. Gat, X. E. Tan, Y. Adi, J. Liu, T. Remez, J. Rapin *et al.*, "Code llama: Open foundation models for code," *arXiv preprint arXiv:2308.12950*, 2023.

[10] Y. Wang, W. Wang, S. Joty, and S. C. Hoi, "Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation," *arXiv preprint arXiv:2109.00859*, 2021.

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[12] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," *arXiv preprint arXiv:2304.02643*, 2023.

[13] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen *et al.*, "Simple open-vocabulary object detection with vision transformers. arxiv 2022," *arXiv preprint arXiv:2205.06230*.

[14] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[15] R. Wang, J. Mao, J. Hsu, H. Zhao, J. Wu, and Y. Gao, "Programmatically grounded, compositionally generalizable robotic manipulation," *arXiv preprint arXiv:2304.13826*, 2023.

[16] A. Stone, T. Xiao, Y. Lu, K. Gopalakrishnan, K.-H. Lee, Q. Vuong, P. Wohlhart, B. Zitkovich, F. Xia, C. Finn *et al.*, "Open-world object manipulation using pre-trained vision-language models," *arXiv preprint arXiv:2303.00905*, 2023.

[17] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, C. Fu, K. Gopalakrishnan, K. Hausman *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," *arXiv preprint arXiv:2204.01691*, 2022.

[18] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[19] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[20] S. Longpre, L. Hou, T. Vu, A. Webson, H. W. Chung, Y. Tay, D. Zhou, Q. V. Le, B. Zoph, J. Wei *et al.*, "The flan collection: Designing data and methods for effective instruction tuning," *arXiv preprint arXiv:2301.13688*, 2023.

[21] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *Advances in neural information processing systems*, vol. 36, 2024.

[22] J. Xu, S. Jin, Y. Lei, Y. Zhang, and L. Zhang, "Reasoning tuning grasp: Adapting multi-modal large language models for robotic grasping," in *2nd Workshop on Language and Robot Learning: Language as Grounding*, 2023.

[23] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

[24] A. Depierre, E. Dellandréa, and L. Chen, "Jacquard: A large scale dataset for robotic grasp detection," *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3511–3516, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:215827111

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[26] H. Wang, S. Ma, S. Huang, L. Dong, W. Wang, Z. Peng, Y. Wu, P. Bajaj, S. Singhal, A. Benhaim, B. Patra, Z. Liu, V. Chaudhary, X. Song, and F. Wei, "Foundation transformers," *CoRR*, vol. abs/2210.06423, 2022.

[27] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3304–3311.

[28] D. Morrison, P. Corke, and J. Leitner, "Learning robust, real-time, reactive robotic grasping," *The International journal of robotics research*, vol. 39, no. 2-3, pp. 183–201, 2020.

[29] S. Kumra, S. Joshi, and F. Sahin, "Antipodal robotic grasping using generative residual convolutional neural network," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 9626–9633.

[30] Y. Domae, H. Okuda, Y. Taguchi, K. Sumi, and T. Hirai, "Fast graspability evaluation on single depth maps for bin picking with general grippers," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 1997–2004.

[31] M. A. Roa and R. Suárez, "Grasp quality measures: review and performance," *Autonomous robots*, vol. 38, pp. 65–88, 2015.

[32] A. Bicchi, "On the closure properties of robotic grasping," *The International Journal of Robotics Research*, vol. 14, no. 4, pp. 319–334, 1995.

[33] A. T. Miller and P. K. Allen, "Graspit!: A versatile simulator for grasp analysis," in *ASME International Mechanical Engineering Congress and Exposition*, vol. 26652. American Society of Mechanical Engineers, 2000, pp. 1251–1258.

[34] A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.

[35] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-end learning of semantic grasping," *arXiv preprint arXiv:1707.01932*, 2017.

[36] J. H. Kwak, J. Lee, J. J. Whang, and S. Jo, "Semantic grasping via a knowledge graph of robotic manipulation: A graph representation learning approach," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9397–9404, 2022.

[37] Y.-H. Wu, J. Wang, and X. Wang, "Learning generalizable dexterous manipulation from human grasp affordance," in *Conference on Robot Learning*. PMLR, 2023, pp. 618–629.

[38] M. Ahn, D. Dwibedi, C. Finn, M. G. Arenas, K. Gopalakrishnan, K. Hausman, B. Ichter, A. Irpan, N. Joshi, R. Julian *et al.*, "Autort: Embodied foundation models for large scale orchestration of robotic agents," *arXiv preprint arXiv:2401.12963*, 2024.

[39] B. Liu, Y. Jiang, X. Zhang, Q. Liu, S. Zhang, J. Biswas, and P. Stone, "Llm+ p: Empowering large language models with optimal planning proficiency," *arXiv preprint arXiv:2304.11477*, 2023.

[40] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities," *Microsoft Auton. Syst. Robot. Res*, vol. 2, p. 20, 2023.

[41] R. Mirjalili, M. Krawez, S. Silenzi, Y. Blei, and W. Burgard, "Lan-grasp: Using large language models for semantic object grasping," *arXiv preprint arXiv:2310.05239*, 2023.

[42] Y. Mu, Q. Zhang, M. Hu, W. Wang, M. Ding, J. Jin, B. Wang, J. Dai, Y. Qiao, and P. Luo, "Embodiedgpt: Vision-language pre-training via embodied chain of thought," *arXiv preprint arXiv:2305.15021*, 2023.