Lan-grasp: Using Large Language Models for Semantic Object Grasping

Reihaneh Mirjalili¹, Michael Krawez¹, Simone Silenzi, Yannik Blei¹, and Wolfram Burgard¹

¹Department of Engineering, University of Technology Nuremberg, Germany

Abstract. In this paper, we propose Lan-grasp, a novel approach towards more appropriate semantic grasping. We use foundation models to provide the robot with a deeper understanding of the objects, the right place to grasp an object, or even the parts to avoid. This allows our robot to grasp and utilize objects in a more meaningful and safe manner. We leverage the combination of a Large Language Model, a Vision Language Model, and a traditional grasp planner to generate grasps demonstrating a deeper semantic understanding of the objects. We first prompt the Large Language Model about which object part is appropriate for grasping. Next, the Vision Language Model identifies the corresponding part in the object image. Finally, we generate grasp proposals in the region proposed by the Vision Language Model. Building on foundation models provides us with a zero-shot grasp method that can handle a wide range of objects without the need for further training or fine-tuning. We evaluated our method in real-world experiments on a custom object data set. We present the results of a survey that asks the participants to choose an object part appropriate for grasping. The results show that the grasps generated by our method are consistently ranked higher by the participants than those generated by a conventional grasping planner and a recent semantic grasping approach. In addition, we propose a Visual Chain-of-Thought feedback loop to assess grasp feasibility in complex scenarios. This mechanism enables dynamic reasoning and generates alternative grasp strategies when needed, ensuring safer and more effective grasping outcomes.

1 Introduction

Objects found in household environments often require a specific way of interaction. For artificial objects, such as tools, the deployment mode can be implied by their design which ensures functionality and user safety. For instance, a knife should be typically held by the grip and not the blade. Similarly, a mug with hot tea is best held by the handle and not the rim of the mug. Incorrect handling can also impair the object itself, e.g. trying to carry a plant by the leaves would most likely lead to damage. Through experience, humans develop an intuitive understanding of objects, their parts, and their proper usage. As robots are increasingly involved in human living environments, it is crucial to provide them with the same kind of semantic knowledge.



Fig. 1: Robot performing the command of "Pick up the ice cream please". The grasp on the left is generated without including semantic information while the grasp on the right is performed using our method leveraging a deeper understanding of the task and the object provided by Large Language Models.

Traditional approaches to robotic grasping [1,2,3] only analyze the object geometry and aim to optimize the grasp stability. Without a deeper understanding of semantic aspects as described above, this can limit the usability of tools or result in object or robot impairment. Recent data-driven approaches [4,5,6] also account for the object class and can generate grasps appropriate for the specific object type. Several works [7,8,9] tackle the problem of task-specific grasping where the object is grasped differently depending on the action at hand. However, most of these methods require substantial computational resources for training and can fail to generalize to unseen object categories. Our objective is an approach for object-specific grasping that ensures tool usability and safety without any training.

We proceed towards this goal by introducing Lan-grasp, a zero-shot method built on foundation models. The scale of these models and the massive size and generality of their training data allow us to reason about a large variety of objects without further training or fine-tuning. In particular, Lan-grasp uses a Large Language Model (LLM) to understand which part of an object is suitable for grasping. Next, this information is grounded in the object image by leveraging a Vision Language Model (VLM). Our method uses GPT-4 as the LLM and the OWL-ViT [10] as the VLM. However, due to the modular structure of Lan-grasp, it can easily be adapted to use other LLMs or VLMs. Finally, we use an off-the-shelf grasp proposal tool [2] to plan the grasps in accordance with the admissible parts of the object detected by the deployed foundation models².

In summary, we make the following contributions:

- 1. We propose a novel approach using foundation models for zero-shot semantic object grasping.
- 2. We demonstrate that the presented approach can work with a wide variety of day-to-day objects without the need for additional training.
- 3. We evaluate our approach by asking human participants to choose the appropriate grasps.

² Video available at https://tinyurl.com/5bnwpkuc.

4. We employ a feedback mechanism using Visual Chain-of-Thought prompting to assess grasp feasibility and dynamically generate alternative grasp strategies when needed.

2 Related Work

Traditional grasping algorithms [1,2,3,11,12] analyze the geometry of the object and the gripper to propose and evaluate a grasping pose. Building on decades of development, these methods are fast and reliable off-the-shelf tools. However, they do not incorporate semantic information and operate based on object shape only. Also, such methods rely on a precise object model and thus suffer from partial or noisy geometry. Data-driven approaches regress grasping candidates from either single view RGB images [4,13] or point clouds [14,15], thus mitigating the need for a complete object model. Further, a network can learn a more natural grasping policy if human-like grasps are included in the training data, where such grasps are either created manually [16] or learned through imitation [6].

Our work is closely related to task-oriented grasping (TOG) and affordance detection. TOG methods restrict the grasp candidates to a specific object part or area, conditioned on the action at hand. Murali et al. [7] create a data set with a large number of objects and tasks and manually annotate task-specific grasp poses. Then, the authors use that data to train a grasp evaluation network. Kwak et al. [5] deploy a knowledge graph to select the gripper type and gripping force appropriate for the given object. Chen et al. [17] propose a network that jointly detects an object and generates a grasping pose according to a natural language command. However, the training requires object, grasp, and command ground truth data. Fang et al. [18] introduce TOG-Net, which optimizes task-oriented grasps and manipulation policies using simulated self-supervision.

Similarly to TOG, affordance detection is the problem of identifying objects or object parts that accommodate a certain action. Do et al. [19] propose an end-to-end trained network that detects object instances in an image and assigns pixel-wise affordance masks to object parts. Liu et al. [20] build on the previous work as a backbone for affordance detection and, in addition, infer the material of object parts to further facilitate semantic grasping. Monica and Aleotti [21] propose a system that decomposes an object point cloud into meaningful parts which then serve as grasping targets. However, the part the robot has to grasp is provided by the user whereas in our method the part is suggested by an LLM. Bohg et al. [22] survey data-driven approaches to grasp synthesis, focusing on methods that sample and rank candidate grasps for both familiar and unknown objects, highlighting the role of feature extraction in these approaches. Nasiriany et al. [23] introduce a prompting framework for VLMs (PIVOT) that refines candidate actions iteratively, showing potential for spatial tasks such as grasping, but their focus is broader, addressing both navigation and manipulation tasks. Wei et al.. [24] propose a novel dataset, DexGYSNet, and utilize it to train a model for dexterous grasp generation based on language guidance. Jian et al. [25] introduce AffordPose, a large-scale dataset for affordance-driven hand-object interactions, focusing on part-level affordance labeling to guide the generation of hand-object interactions in fine detail. Zhu et al. [26] propose a framework for human-like dexterous grasping, using semantic touch codes and object functional areas to guide grasps. Ren et al. [27] introduce ATLA, a meta-learning framework that uses LLMs to accelerate tool manipulation by combining language-based policies with affordances, focusing on general tool use rather than grasping specific object parts.

Foundation models have recently attracted a lot of attention in different sub-fields of robotics [28,29,30] and have been also applied to boost TOG and affordance detection. Ngyen et al. [31] train an open-vocabulary affordance detector for point clouds whereby CLIP is deployed to encode the affordance labels. Similarly, Tang et al. [8] use CLIP to facilitate task-specific grasping from RGB images and language instructions. The authors propose to utilize CLIP embeddings from intermediate CLIP layers to allow their affordance detector to reason about fine-grained object parts. Gao et al. [32] annotate a large object data set with physical object properties like mass or fragility and fine-tune a VLM on it to improve manipulation planning. Other methods integrate LLMs for encoding tasks or object parts from natural language. Song et al. [33] use BERT as the language back-end and train a network that grounds object parts in a point cloud from a user instruction. Here, however, the part label is explicitly referred to in the user input. The approach of Tang et al. [9] lifts this limitation by prompting an LLM to describe the shape and parts of an object. The LLM response is then processed by a Transformer-based grasp evaluation network. Our method also relies on an LLM for deciding what object part should be grasped. The crucial difference to the above works is that our approach relies solely on foundation models and does not require any training. Thus, once more powerful foundation models are available, the performance of our approach is easily improved by switching to a novel LLM or VLM. Newbury et al. [34] conduct a systematic review of deep learning methods for six-DoF grasp synthesis, highlighting sampling-based, direct regression, and reinforcement learning methods to generate grasp poses. Wu et al. [35] propose an approach to enhance LMMs' robustness in vision tasks, introducing reasoning capabilities to correct false premises, which improves reasoning for affordance-based grasping tasks. In a similar line of thought, Huang et al. [36] propose VoxPoser, a framework that generates 3D value maps to guide robotic manipulation using affordances extracted from LLMs and visual grounding, however their focus is on manipulation tasks rather than detecting the specific grasping part of an object.

Finally, embodied vision-language models like PALM-E [37] aim to close the gap between language, vision, and robot actions by training the network jointly on these modalities. The recent RT-2 model [38] shows remarkable capabilities of generating robot controls from language instructions. Whether these approaches choose the appropriate object parts for grasping remains an open question, since no explicit experiments were described in the publications and the code is not yet available at the time of writing.

3 Method Description

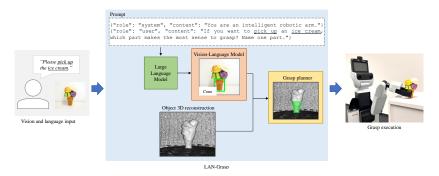


Fig. 2: The core concept of our approach in a nutshell: The command from the user is turned into a prompt suitable for the Large Language Model (LLM). With this prompt as an input, the large language model outputs the proper part for grasping the object, which in this example is the *cone*. This word is then grounded to the object image using a Vision-Language Model (VLM). The grounded grasp part is integrated to the 3D reconstruction model of the object to generate the proper grasp.

In this section, we explain the details of our approach. Lan-grasp generates a grasping pose from an object label, a camera image showing that object, and the corresponding object geometry. The method consists of two parts. In the language module, an LLM first decides what object part to grasp, which is then grounded in the image by a VLM. The resulting bounding box is projected onto the object geometry to mark the grasp target in the grasp planning module. Thanks to the modular structure of Lan-grasp, it is easy to enhance the pipeline by leveraging more advanced models as they emerge. The pipeline of Lan-grasp is depicted in Figure 2.

3.1 Language Module

In the first step, the object label <object> provided by the user is transferred into a LLM prompt. The scheme of the prompt is chosen to be compatible with GPT-4 which is the LLM that we used in the pipeline [39]. We included the last sentence to prevent the LLM from giving extra explanations and thus only output the desired object part. We use OWL-ViT [10] as the VLM for grounding the object part label in the image. It builds on the Vision Transformer Architecture, first presented by Dosovitskiy et al. [40]. The authors then pre-train the model using contrastive learning [41] on a large image-text data set [42]. Afterward, the authors fine-tune the model on publicly available detection data sets. OWL-ViT then detects and marks the desired object part with a bounding box which is projected on the object 3D model.

3.2 Grasp Planning Module

We deploy the GraspIt! simulator [43] as our grasp proposal generator. It is a standard tool that operates on geometric models and evaluates grasps according to physical constraints. Thus, the first step for grasp planning is to create a dense 3D mesh model of the object. In our setup, we use two fixed RGB-D sensors and a turning table for object scanning. We acquire the camera poses from an Aruco board and integrate the depth images via KinectFusion [44]. However, we note that any other suitable reconstruction approach could be used here.

The possible poses for grasping the object are generated by sampling. The initial gripper position is chosen based on object geometry, after that the gripper is iteratively brought closer to the object while avoiding obstacles [45]. In this regard, GraspIt! splits the scene into object and obstacle geometry, and we exploit this mechanism by marking the mesh parts that project into the VLM-generated bounding box as object and the rest as obstacle. This enforces grasping only at the desired object part. The resulting grasp proposals are ranked based on grasp efficiency and finger friction.

We want to point out that our approach is agnostic about the grasp planner and could be potentially replaced by other tools that do not require a complete object model, e.g., the method of Alliegro et al. [15]. In this case, the reconstruction step could be skipped entirely and the grasp candidates could be computed on a point cloud acquired from the robot's sensors.

4 Experimental Evaluation

In this section, we present the details of our experiments and results. Our goal is to demonstrate that our method proposes to grasp object parts that are preferred by humans on a variety of objects. We argue that humans generally choose grasps that enable correct tool usage and ensure safety and that a robot retains these desirable qualities by executing similar grasps. To that end, we first collect a data set of typical household objects. Next, we apply our approach to these objects and execute the grasping on a real robot. Finally, we show that our grasping strategy is similar to human preferences obtained through a survey and that our approach outperforms two baselines based on this similarity metric. In the following, we describe our data set, provide details on the baseline approaches and the performed experiments, and discuss the results. In subsection 4.5 we present an extension to the main algorithm that reasons about the feasibility of a grasp in complex scenarios. Finally, in subsection 4.6 we perform an ablation study on several components of the pipeline.

4.1 Dataset

We collect a data set containing 22 different objects commonly found in household environments. We chose these objects to cover a wide range of situations where semantic knowledge is required for proper grasping. Our first objective was to showcase grasping on functional objects like tools or kitchen supplies, e.g., shovel, hand brush, and knife. Further, we included delicate objects that might be damaged with an improper grasp, for instance, rose, cupcake, and ice cream. For other objects, a wrong grasp can cause a dangerous situation, e.g., candle. Finally, we include objects where an improper grasp might not necessarily be harmful but is rather unnatural to a human observer, for instance, doll, bag, and wine glass. The objects in the data set are shown in Figure 3, Figure 4, and Figure 5.

4.2 Experimental Setup and Baselines

For 11 objects from our data set, listed in Table 1, we perform real-world experiments using the Human Support Robot (HSR) [46]. we first scan each object and than apply GraspIt! to the resulting 3D model as detailed in subsection 3.2. From the top-20 grasp proposals we randomly pick one and execute it using the proprietary HSR motion planner.

Our first baseline is the plain GraspIt! simulator. Here we use the same 3D models as for our approach but do not restrict grasping to the object part selected by the language model. For each object, we evaluate the top-20 grasp proposals and carry two of them out on the HSR, as shown in Figure 3.

The second baseline is GraspGPT [9], a recent approach to task-oriented grasping. This method requires as input an object point cloud and a natural language prompt describing the object, the object class, and the task. We generate the point clouds from the object meshes reconstructed as above and use an object-specific activity as the task label. Again, we retrieve 20 grasps per object but do not carry them out on the robot.

4.3 Qualitative Results

In this section, we present and discuss the grasping results of Lan-grasp and the GraspIt!-baseline. The grasps executed on the robot are shown in Figure 3. For the rest of the objects, the grasping area suggested by our method is presented in Figure 5.

The results suggest that Lan-grasp proposes grasps suitable for the usage of the respective object. For instance, grasping the handle for shovel and broom corresponds to the intended use of these items. For lollipop and cupcake, the grasp is placed away from the edible part at the stick and the wrapper, respectively. It is noteworthy that our method is able to understand the relation between stacked objects, e.g., flowers in a vase or plate of cake. Also, for a single cup, Lan-grasp suggests grasping the handle while for the cup on a saucer the grasp proposal is the saucer. Other objects, e.g., doll, bag, or wine glass, do not possess a critical area where grasping would cause harm or directly interfere with the functionality. However, our method is able to generate grasps that are closer to how a human would handle these items. In contrast to Lan-grasp, the areas suggested by GraspIt! are expectantly random and do not consider semantic intricacies.



Fig. 3: The performed grasps by the HSR robot: Each column presents the grasps for one object. The first two rows for each object, show the grasps generated without semantic knowledge about the objects, while the third and fourth rows show the grasps generated by Lan-grasp.



Fig. 4: The performed grasps by the HSR robot: Each column presents the grasps for one object. The first two rows for each object, show the grasps generated without semantic knowledge about the objects, while the third and fourth rows show the grasps generated by Lan-grasp.

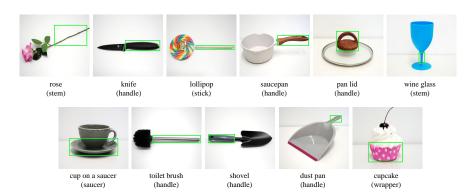


Fig. 5: The results of Lan-grasp on a set of common household objects. The green bounding box shows the area to grasp suggested by the method.

Object	Preferred Part	GraspIt!	GraspGPT	Lan-grasp
doll	torso 92.1%	0.28	0.48	0.92
ice cream	cone 100.0%	0.05	0.40	1.00
candle	base 93.1%	0.22	0.57	0.93
flowers in the vase	vase 93.2%	0.32	0.73	0.93
bag	handle 91.1%	0.79	0.69	0.91
plant	pot 94.3%	0.16	0.56	0.94
hand brush	handle 95.4%	0.65	0.95	0.95
toilet brush	handle 97.6%	0.42	0.52	0.98
cactus	pot 98.8%	0.26	0.99	0.99
cupcake	wrapper 100.0%	0.10	0.40	1.00
cup on a saucer	saucer 81.2%	0.24	0.59	0.81
plate of cake	plate 98.8%	0.11	0.51	0.99
mug	handle 77.1%	0.28	0.73	0.77
saucepan	handle 94.3%	0.36	0.94	0.94
broom	handle 97.6%	0.42	0.98	0.98
Average		0.31	0.67	0.94

Table 1: Similarity of grasping area preferences compared to a human user. The left half of the table lists the objects and the object part the majority of survey participants suggested for grasping, with the corresponding percentage of users. The right half of the table shows the similarity scores per object for the two baselines and the proposed method.

4.4 Quantitative Results

To support the claim that our approach proposes grasps similar to human preferences, we designed a questionnaire on grasping choices. A group of 83 participants were presented with images of all objects used in the experiments and were asked where they would grasp them. For each object, the participants could choose between two parts marked by bounding boxes in the image. The survey results are summarized in Table 1. Per object, we state the preferred part and the percentage of participants that selected it.

For the proposed approach and the baselines, we want to evaluate how similar the generated grasps are compared to the ones suggested by human users. Given that an object is segmented into parts a and b, let $p_a \in [0,1]$ be the empirical probability that a method grasps at part a and $p_b = 1 - p_a$ that part b is grasped. Further, let p_a^b be the human grasping frequency at a according to the survey results and p_a^x the corresponding frequency produced by one of the considered methods. To compute p_a^x for the baselines, we obtained 20 grasp proposals from each algorithm and counted the grasps falling into region a. Lan-grasp restricts the grasps to the object part selected by the LLM, which in our experiments robustly proposed the same part for a given object. Thus, the values of p_a^x were here either 1 or 0. Finally, we computed a per-object similarity score for each method x as $sim_x = 1 - |p_a^b - p_a^x|$. These scores are shown in Table 1 along with the average similarity scores over all objects.

Our method consistently outperforms the baselines on the similarity score and ties only in four cases with GraspGPT. The average similarity score of Langrasp is considerably higher with the value of 0.94 compared to 0.31 achieved by GraspIt! and 0.67 achieved by GraspGPT. We further note that in all cases, the object part choice of Lan-grasp coincides with the majority vote of the survey participants. The low score of GraspIt! is not surprising since it only considers geometric and not semantic aspects of the object. Thus, whether the object is grasped in a particular region is pure chance, and we expect the similarity score to be closer to 0.5 for a larger data set. GraspGPT exhibits a better performance compared to GraspIt! due to leveraging semantic concepts and LLMs. However, it was tuned on a data set mostly containing tools and house supplies and thus does not perform well on objects like a doll or an ice cream.

4.5 Grasp Feasibility Feedback

So far, we assumed that the grasp suggested by the LLM is feasible, however, this is not always the case. For instance, the referenced object part might not be visible in the image, broken, or occluded by other objects. Further, the execution of the grasp could lead to undesirable outcomes. In order to mitigate this issue, we propose a feedback loop consisting of a VLM and an LLM that communicate with each other to find a feasible grasp. The approach and the used prompts are shown in Figure 6. As in our original pipeline, the LLM first suggests an object part based on the object label. Next, the VLM analyzes the image and evaluates

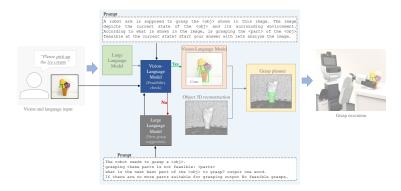


Fig. 6: Schematic diagram of the feasibility feedback loop added to the core pipeline of Lan-grasp.

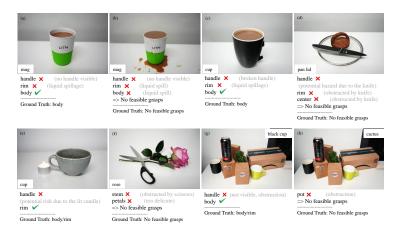


Fig. 7: Qualitative results of our method on complex grasping scenarios. The object part labels below each image show the suggested grasp part and whether it was considered feasible or not by our feedback algorithm.



Fig. 8: Instances of our dataset for complex grasp scenarios.

the feasibility of the grasping part. Crucially, we follow the idea of Chain-of-Thought (CoT) prompting [47] and include the sentence "Start your answer with lets analyze the image." into the prompt. If feasible, we proceed as before. If not, the LLM receives a list of all object parts rejected so far and is asked to propose another grasp. This loop repeats until either a feasible grasp part is found or there are no more suitable parts of the object left for grasping. In our implementation, the roles of LLM and VLM are carried out by the same model (GPT-40).

To evaluate this approach we gathered a dataset of 18 challenging scenarios, shown in Figure 8 and Figure 7, where the initial grasp suggestion is not feasible and requires reasoning to find the right grasp part. We defined the ground truth manually to evaluate the results. We ran the algorithm 5 times to obtain an average success rate of 91.14% over all scenes. The qualitative results are shown for part of the dataset in Figure 7. Our algorithm can take into account different criteria, e.g., the risk due to the proximity of the lit candle, potential mess due to the overfilled cup, or occlusions by nearby objects. We further analyze the effects of VLM choice and CoT in the following section.

4.6 Ablation Study

In this section, we evaluate the influence of different algorithm components on the above results, specifically the choice of the LLM and VLM and the effect of CoT on the feasibility feedback. First, we consider the pipeline without feedback as described in subsection 3.1. Here we consider GPT-3.5-turbo and GPT-4 as LLMs. We further investigate whether replacing the LLM with a VLM improves the performance. To that end, we deploy GPT-40, GPT-40-mini, and the open-source LLaVA-1.5 7B VLM and provide them with an image of the object to grasp.

Here, we run our pipeline only until detecting the grasping area in the image, without executing the grasp nor generating a grasping pose. Therefore, we use different metrics than in subsection 4.4. First, we count the exact matches between the LLM-generated and GT object part labels and report the average success rate. Second, we compute the Intersection-over-Union (IoU) for the proposed grasping regions and the ground truth. All algorithm versions are evaluated on data from subsection 4.1 and the results are summarized in Table 2.

Both text-only GPT versions perform equally with an 81.8% success rate and an IoU score of around 0.63. The VLM variants perform slightly better with a success rate of 86.3%. We note that there is no difference between the flagship GPT-40 and the downsized GPT-40-mini model. LLaVA performed significantly worse with 45.5% success rate and 0.38 IoU score. Analyzing the object parts suggested by LLaVA showed that the model was correct for objects possessing a handle, for other objects the answer was either wrong or referred to generic image locations, e.g., 'bottom' or 'top'. A straightforward explanation could be simply the smaller model size. However, another reason might be that we used the same prompt for LLaVA as for the GPT models, and better results could be achieved with further prompt engineering specifically targeting LLaVA.

method	text only		text and image		
	GPT-3.5-turbo	GPT-4	GPT-40	GPT-40-mini	LLaVA
success rate	81.8%	81.8%	86.3%	86.3%	45.5%
IoU	0.63	0.64	0.65	0.64	0.38

Table 2: Ablation of LLMs and VLMs in the main Lan-grasp pipeline.

For the feasibility feedback algorithm, we compare GPT-4o and GPT-4o-mini. Further, we experiment with two prompt variants, the first with CoT (zero-shot-CoT) and the second without (zero-shot). In the latter, we omit the sentence "Start your answer with lets analyze the image." from the prompt. The experiments were performed in the same fashion as in subsection 4.5 and the results are reported in Table 3. First, we consider the CoT variants. With 62.75% success rate GPT-4o-mini performed significantly worse than the larger GPT-4o (91.14%), which indicates that model size is an important factor for complex reasoning. Without CoT the performance dropped to 65.33% for GPT-4o and 52.78% for GPT-4o-mini. That result demonstrates that our algorithm, in fact, benefits from the CoT approach.

method	GPT-4		GPT-40 mini		
	Zero-shot-CoT	Zero-shot	Zero-shot-CoT	Zero-shot	
success rate	91.14%	65.33%	62.75%	52.78%	

Table 3: Ablation of VLMs and prompting strategies in feasibility feedback.

4.7 Conclusion and Future Work

In this paper, we presented Lan-grasp, a novel approach to semantic object grasping. By leveraging foundation models, we provide our approach with a deep understanding of the objects and their intended use in a zero-shot manner. Through extensive experiments, we showed that for a wide range of objects Langrasp is generating grasps that are preferred by humans and also ensure safety and object usability. In particular, the proposed grasps were compared to human preferences gathered through a questionnaire. The evaluations showed that Langrasp performs consistently better on that metric than the baseline methods. We also proposed a feedback loop approach that reasons about grasp feasibility in complex scenarios. In future work, we plan to test Lan-grasp in more complex, and cluttered environments to evaluate its robustness. Additionally, we aim to enhance the feedback loop by introducing mechanisms for when no feasible grasp is detected. For instance, the robot could ask a human for assistance or employ more sophisticated reasoning strategies to modify the environment to facilitate

grasping. This would make our algorithm capable of handling more complex real-world scenarios. Inspired by these results, in the future we plan to further exploit Large Language Models to not only decide where to grasp an object but also how to grasp and hold it according to a specific task. As an example, we would expect a robot operating in daily environments to hold a knife vertically and downwards when the task is to carry the knife around rather than holding the knife in a horizontal pose. This would be the next step towards more meaningful grasps that help robots with object manipulation and task execution in day-to-day environments.

References

- 1. A. Bicchi, "On the closure properties of robotic grasping," *International Journal of Robotics Research (IJRR)*, 1995.
- 2. A. T. Miller and P. K. Allen, "Graspit!: A versatile simulator for grasp analysis," in ASME International Mechanical Engineering Congress and Exposition, 2000.
- 3. A. Ten Pas, M. Gualtieri, K. Saenko, and R. Platt, "Grasp pose detection in point clouds," *International Journal of Robotics Research (IJRR)*, 2017.
- 4. E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-end learning of semantic grasping," arXiv preprint arXiv:1707.01932, 2017.
- 5. 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 (RA-L)*, 2022.
- Y.-H. Wu, J. Wang, and X. Wang, "Learning generalizable dexterous manipulation from human grasp affordance," in Conference on Robot Learning (CoRL), 2023.
- 7. A. Murali, W. Liu, K. Marino, S. Chernova, and A. Gupta, "Same object, different grasps: Data and semantic knowledge for task-oriented grasping," in *Conference on Robot Learning (CoRL)*, 2021.
- 8. C. Tang, D. Huang, L. Meng, W. Liu, and H. Zhang, "Task-oriented grasp prediction with visual-language inputs," arXiv preprint arXiv:2302.14355, 2023.
- 9. C. Tang, D. Huang, W. Ge, W. Liu, and H. Zhang, "Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping," arXiv preprint arXiv:2307.13204, 2023.
- 10. M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen *et al.*, "Simple openvocabulary object detection," in *European Conference on Computer Vision* (ECCV), 2022.
- 11. B. S. Zapata-Impata, P. Gil, J. Pomares, and F. Torres, "Fast geometry-based computation of grasping points on three-dimensional point clouds," *International Journal of Advanced Robotic Systems*, 2019.
- 12. A. Myers, C. L. Teo, C. Fermüller, and Y. Aloimonos, "Affordance detection of tool parts from geometric features," in 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 1374–1381.
- J. Redmon and A. Angelova, "Real-time grasp detection using convolutional neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015.
- 14. B. Zhao, H. Zhang, X. Lan, H. Wang, Z. Tian, and N. Zheng, "Regnet: Region-based grasp network for end-to-end grasp detection in point clouds," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

- 15. A. Alliegro, M. Rudorfer, F. Frattin, A. Leonardis, and T. Tommasi, "End-to-end learning to grasp via sampling from object point clouds," *IEEE Robotics and Automation Letters (RA-L)*, 2022.
- 16. E. Corona, A. Pumarola, G. Alenya, F. Moreno-Noguer, and G. Rogez, "Ganhand: Predicting human grasp affordances in multi-object scenes," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- 17. Y. Chen, R. Xu, Y. Lin, and P. A. Vela, "A joint network for grasp detection conditioned on natural language commands," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.
- 18. 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.
- T.-T. Do, A. Nguyen, and I. Reid, "Affordancenet: An end-to-end deep learning approach for object affordance detection," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.
- W. Liu, A. Daruna, and S. Chernova, "Cage: Context-aware grasping engine," in IEEE International Conference on Robotics and Automation (ICRA), 2020.
- 21. R. Monica and J. Aleotti, "Point cloud projective analysis for part-based grasp planning," *IEEE Robotics and Automation Letters (RA-L)*, 2020.
- 22. J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—a survey," *IEEE Transactions on robotics*, vol. 30, no. 2, pp. 289–309, 2013.
- S. Nasiriany, F. Xia, W. Yu, T. Xiao, J. Liang, I. Dasgupta, A. Xie, D. Driess, A. Wahid, Z. Xu et al., "Pivot: Iterative visual prompting elicits actionable knowledge for vlms," arXiv preprint arXiv:2402.07872, 2024.
- Y.-L. Wei, J.-J. Jiang, C. Xing, X. Tan, X.-M. Wu, H. Li, M. Cutkosky, and W.-S. Zheng, "Grasp as you say: Language-guided dexterous grasp generation," arXiv preprint arXiv:2405.19291, 2024.
- J. Jian, X. Liu, M. Li, R. Hu, and J. Liu, "Affordpose: A large-scale dataset of hand-object interactions with affordance-driven hand pose," in *Proceedings of* the IEEE/CVF International Conference on Computer Vision, 2023, pp. 14713– 14724.
- T. Zhu, R. Wu, X. Lin, and Y. Sun, "Toward human-like grasp: Dexterous grasping via semantic representation of object-hand," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15741–15751.
- A. Z. Ren, B. Govil, T.-Y. Yang, K. R. Narasimhan, and A. Majumdar, "Leveraging language for accelerated learning of tool manipulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 1531–1541.
- 28. R. Mirjalili, M. Krawez, and W. Burgard, "Fm-loc: Using foundation models for improved vision-based localization," arXiv preprint arXiv:2304.07058, 2023.
- 29. C. Huang, O. Mees, A. Zeng, and W. Burgard, "Audio visual language maps for robot navigation," arXiv preprint arXiv:2303.07522, 2023.
- Y. Cui, S. Karamcheti, R. Palleti, N. Shivakumar, P. Liang, and D. Sadigh, "No, to the right: Online language corrections for robotic manipulation via shared autonomy," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-*Robot Interaction, 2023.
- 31. T. Ngyen, M. N. Vu, A. Vuong, D. Nguyen, T. Vo, N. Le, and A. Nguyen, "Open-vocabulary affordance detection in 3d point clouds," arXiv preprint arXiv:2303.02401, 2023.

- 32. J. Gao, B. Sarkar, F. Xia, T. Xiao, J. Wu, B. Ichter, A. Majumdar, and D. Sadigh, "Physically grounded vision-language models for robotic manipulation," arXiv preprint arXiv:2309.02561, 2023.
- 33. Y. Song, P. Sun, Y. Ren, Y. Zheng, and Y. Zhang, "Learning 6-dof fine-grained grasp detection based on part affordance grounding," arXiv preprint arXiv:2301.11564, 2023.
- 34. R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic et al., "Deep learning approaches to grasp synthesis: A review," *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3994–4015, 2023.
- 35. T.-H. Wu, G. Biamby, D. Chan, L. Dunlap, R. Gupta, X. Wang, J. E. Gonzalez, and T. Darrell, "See say and segment: Teaching lmms to overcome false premises," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13459–13469.
- 36. 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.
- 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.
- 38. 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.
- 39. OpenAI, "Gpt-4 technical report," arXiv:2303.08774, 2023.
- A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.
- X. Zhai, X. Wang, B. Mustafa, A. Steiner, D. Keysers, A. Kolesnikov, and L. Beyer, "Lit: Zero-shot transfer with locked-image text tuning," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- 42. C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *International Conference on Machine Learning (ICML)*, 2021.
- 43. A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine (RAM)*, 2004.
- 44. R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2011.
- 45. A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2003.
- T. Yamamoto, K. Terada, A. Ochiai, F. Saito, Y. Asahara, and K. Murase, "Development of human support robot as the research platform of a domestic mobile manipulator," ROBOMECH journal, 2019.
- 47. 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. 22199–22213, 2022.