3D Understanding and Zero-Shot Task-Oriented Grasping via Grounding Symbolic Representations

Samuel Li

CMU-RI-TR-25-20

May 2025

School of Computer Science

The Robotics Institute

Carnegie Mellon University

Pittsburgh, Pennsylvania

Thesis Committee

Prof. Katia Sycara, Chair

Prof. Shubham Tulsiani

Brian Yang

Submitted in partial fulfillment of the requirements for the Degree of Master of Science in Robotics.

Copyright © 2025 Samuel Li. All rights reserved.

Abstract

Task-oriented grasping requires robots to reason not only about the geometry of objects but also about the function and semantics of their parts in context. While large language models (LLMs) offer a powerful source of commonsense knowledge, they are not grounded in physical geometry. This thesis explores how symbolic object representations can bridge this gap, enabling LLMs to guide grasp selection in a zero-shot setting. We first propose a method that decomposes objects into convex parts and represents them as symbolic graphs, allowing the LLM to assign semantic roles and select task-relevant grasps using only an object name and task description. Building on this, we introduce a 3D-aware approach that encodes object structure as semantic Constructive Solid Geometry (CSG) trees, which are optimized to match real-world observations. These symbolic representations enable the LLM to reason about part affordances directly in 2D or 3D space. Through extensive real-world experiments, we demonstrate that grounding language-based reasoning in structured geometry yields robust, interpretable, and generalizable task-oriented grasping, outperforming prior baselines across varied objects, viewpoints, and occlusions.

Acknowledgements

I would like to thank the many wonderful people I've been fortunate to know and work with during my time at the Robotics Institute and beyond.

First and foremost, I am deeply grateful to my advisor, Professor Katia Sycara. Her wisdom, encouragement, and unwavering support have been instrumental in my development as a researcher. Her high standards and sharp insight have consistently pushed me to think more deeply, communicate more clearly, and approach my work with greater care and rigor. At the same time, her genuine commitment to her students and thoughtful mentorship have created an environment where I've been able to grow, take risks, and learn with confidence. I am a stronger researcher, with many exciting opportunities ahead, because of the time I've spent working with her.

I am especially thankful to Simon Stepputtis. He is a truly sharp researcher and deeply caring mentor. His day-to-day guidance—whether through feedback on ideas, discussions on experiments, or advice on writing and presentation—has been invaluable. Just as importantly, he has been a steady and supportive presence, helping me navigate grad school with perspective and encouragement. As he starts his own lab as a professor, any student or collaborator who works with him will be incredibly fortunate.

I would also like to thank my undergraduate advisor, Professor Yuxiong Wang, who helped set me on this research path. I'm grateful for his continued support and encouragement.

Many others have helped shape this work. I am thankful to Yaqi Xie, Woojun Kim, and Joseph Campbell, who have been generous coauthors and collaborators, for their ideas and support.

Within the lab, I've been lucky to be surrounded by an incredibly thoughtful and supportive group. For encouragement, conversations, and friendship, I'd like to thank (in alphabetical order): Aryan Mangal, Sarthak Bhagat, Silong Yong, Srujan Deolasee, Venkata Nagarjun, and Zifu Wan.

I'm also grateful to my collaborators at Wayve AI, where I spent a memorable and enriching summer. I especially want to thank my fellow interns, Pujith Kachana and Brian Yang, for their energy, insight, and enthusiasm, and my mentors, Prajwal Chidananda and Saurabh Nair—both former Robotics Institute MS students—for their guidance, support, and encouragement. I'm thankful to everyone at Wayve who made it such a rewarding experience.

Finally, I owe everything to my family. I'm endlessly thankful to my parents, whose hard work, sacrifices, and belief in me have made every opportunity possible. My every success rests on the foundation they built. And to my sister, Tracy—thank you for the laughs and your grounding presence throughout this journey.

Funding

This work was supported by the following research grants: DARPA under grants FA8750-23-2-1015 and HR00112490409, ARL under grant W911NF-2320007, and ONR under grant N00014-23-1-2840.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

As robots begin to operate in increasingly dynamic and human-centric environments, they must go beyond merely picking up objects—they must understand how to grasp them in ways that are contextually appropriate and task-relevant. This problem, known as *task-oriented grasping*, requires robots to reason not only about object geometry but also about the semantic roles and functions of different object parts. For example, when asked to hand over a pair of sunglasses, a robot should grasp the frame by the temples, not the lenses, to preserve visibility and comfort for the recipient.

Humans perform such reasoning effortlessly. When we encounter unfamiliar objects, we rely on our world knowledge and visual intuition to identify functional parts and select appropriate grasp locations, even without prior experience or category-level training. Enabling robots to replicate this ability requires bridging two key capabilities: the ability to interpret physical geometry and the ability to reason symbolically about semantics and function.

Recent advances in large language models (LLMs) offer a promising foundation for this type of commonsense and task-informed reasoning. LLMs encode a vast amount of world knowledge and have demonstrated strong performance in tasks that require generalization, analogy, and abstract understanding. However, while LLMs are adept at reasoning over symbolic representations, they are not inherently grounded in physical geometry or capable of directly interpreting sensor data from the real world. To leverage their capabilities for robotic manipulation, we must establish an interface between symbolic reasoning and geometric perception.

This thesis investigates how symbolic object representations can serve as a bridge between the perceptual input of robotic systems and the symbolic reasoning capabilities of LLMs. We discuss two proposed approaches that embed geometry into symbolic structures interpretable by LLMs, enabling zero-shot task-oriented grasping with minimal prior knowledge or supervision.

In Chapter 2, we introduce *ShapeGrasp*, a method that constructs a symbolic representation of an object by decomposing it into simple 2D convex parts using geometric heuristics. These parts are organized into a graph structure that encodes their spatial relationships and basic geometric attributes. An LLM is then prompted with the graph and tasked with assigning semantic labels (e.g., ”handle”, ”blade”) and evaluating which part is most suitable for a specified task. ShapeGrasp requires only the object’s name and the task description as input and performs all reasoning zero-shot

using the LLM. On a real-world robot platform, ShapeGrasp achieves 92% accuracy in identifying the correct part to grasp and successfully completes the task in 82% of trials.

While ShapeGrasp demonstrates the power of symbolic structure for LLM-guided grasping, its decomposition is purely geometric and view-dependent, limiting robustness in complex 3D scenes or under occlusion. To address these limitations, Chapter 3 presents *SemanticCSG*, a follow-up approach that builds a semantically meaningful 3D object representation using Constructive Solid Geometry (CSG) trees. In this method, the LLM generates a symbolic hypothesis about the object's structure—composed of primitive 3D shapes and Boolean operations—and this hypothesis is optimized to match a single-view observation using differentiable rendering techniques. The result is a grounded symbolic model aligned with the physical object. This representation allows the LLM to reason about part semantics and task relevance directly in 3D space. SemanticCSG achieves 85% accuracy in part identification and a 77% task-specific grasping success rate across diverse viewing angles and occlusions.

Together, these works highlight a general strategy for enabling robots to benefit from the knowledge embedded in LLMs while remaining grounded in physical space and geometry. By transforming perceptual input into symbolic forms—whether through geometric decomposition or learned CSG representations—robots can query LLMs in a structured and interpretable way, making task-oriented manipulation more flexible, generalizable, and safe. This symbolic grounding of semantics into geometry represents a promising step toward more intelligent and collaborative human-robot interaction.

Chapter 2

ShapeGrasp: Zero-Shot Task-Oriented Grasping with Large Language Models through Geometric Decomposition

2.1 Introduction

In-home environments present a significant challenge for real-world robotics, primarily due to their highly unstructured nature. As a result, these environments frequently contain novel objects not present in the robot's training environment; however, interacting with such objects, particularly grasping them in a task-dependent manner, is a necessary skill. Grasping an object in a way that facilitates a certain task, namely *task-oriented grasping*, requires a system to not only detect an object but also to reason over the utility of its parts. For example, when picking up a hammer with the goal to "hand it over" (see Fig. 2.2), the robot should grasp the hammer by the head to promote ease and safety for the human receiving it. Large Language Models (LLMs) provide the capability of such commonsense reasoning and can be utilized for task-oriented grasping with only a minimal set of contextual information, namely the object's name and desired task [28, 33].

However, zero-shot task-oriented grasping remains challenging, particularly since current approaches are computationally expensive and may require additional information beyond the object's name and intended task for high performance, such as desired object part names [28, 33], thus limiting their zero-shot performance. In this work, we propose `ShapeGrasp`, **a robust grasping framework based on a geometric decomposition of the target object and shape-based semantic part reasoning leveraging LLMs, capable of conducting task-oriented reasoning and grasping of novel objects**. This approach is inspired by the human ability to interact with a novel object by analyzing its geometric composition, relating it to prior knowledge, and inferring each part's utility [8] before utilizing this knowledge to identify a suitable part for an intended task. Unlike prior approaches, we introduce a multi-step reasoning approach that leverages a symbolic graph of basic shapes and geometric attributes that compose the object and a multi-stage LLM prompt that a) assigns semantic meaning to each part of the object given its name and b) reasons over the affordances and task utility of each part to select the most suitable part for the specific task.

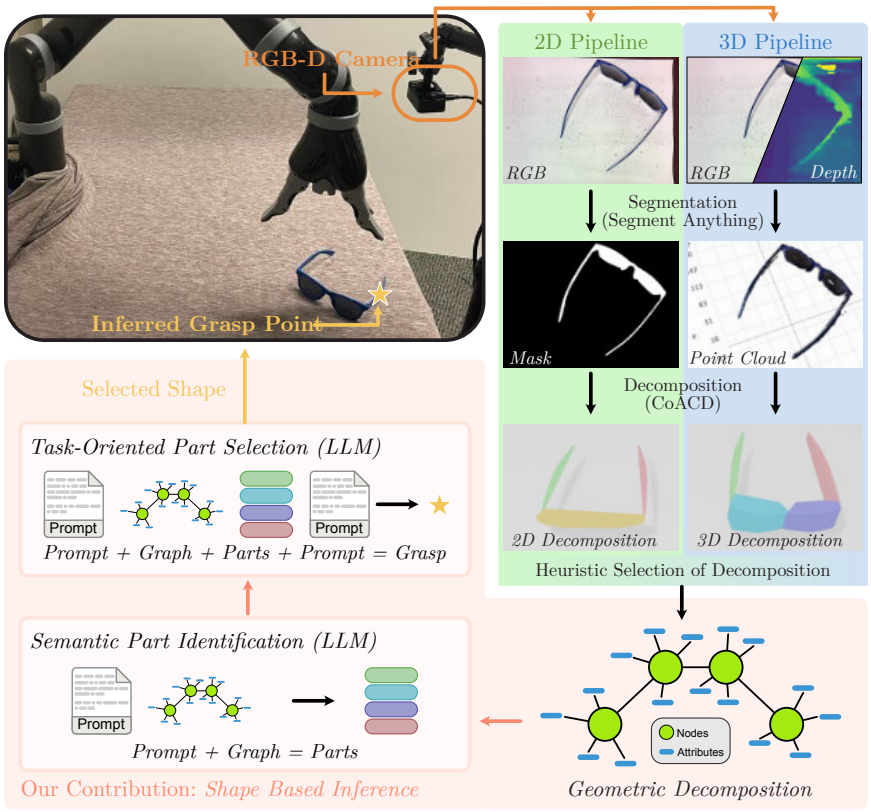

Figure 2.1: **The `ShapeGrasp` Pipeline**: Given a target object, our RGB+D-based approach decomposes the object into basic convex parts. We propose a heuristic approach to decide which decomposition to use before converting it into a shape graph, allowing an LLM to utilize its commonsense reasoning to identify part semantics and task suitability.

Figure 3.1 provides a high-level overview of the functionality proposed in `ShapeGrasp`. Starting with a passive monocular RGB+D image, we segment the target object using a pre-trained Segment Anything Model (SAM) [17], receiving a 2D image mask and the object's relevant point cloud by applying the mask to the depth image. We then calculate two approximate convex decompositions using a pre-trained CoACD [42]: 1) a 3D decomposition using the masked point cloud and 2) a 2D decomposition using the image's SAM mask. `ShapeGrasp` heuristically identifies a suitable object decomposition from the two decomposition procedures, approximates each convex hull with a basic geometric shape, including a combination of rectangles, circles, triangles, and ellipses, and represents these parts and their spatial relationships as a graph. We then utilize an LLM to determine the semantic meaning of each shape given the name of the object and identify the best part to grasp using the assigned fine-grained semantic part labels and their suitability for the desired task. Through this process, inspired by the Chain of Thought [41] approach, we improve the reasoning capabilities of our method. Finally, the object is grasped at the identified part by calculating the centroid and principle components of its masked point cloud to determine the grasping location and orientation. Our work provides the following contributions:

- We introduce `ShapeGrasp` – a novel approach for zero-shot task-oriented grasping lever-

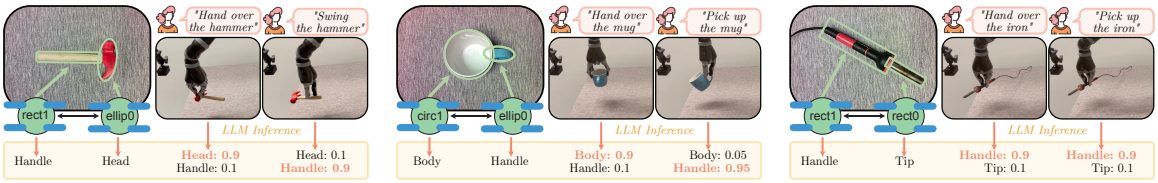

Figure 2.2: Different resulting grasps given our shape-based inference pipeline. Parts in orange-boldface are ultimately grasped. Green circles and blue lines represent the part-graph decomposition with each entity's associated attributes.

aging a previously unseen object's geometry by creating a graph of basic geometric shapes from decomposed convex parts, allowing for improved reasoning over each part's semantic significance and task-oriented utility.

- Our proposed method is a lightweight approach in both the vision-based graph construction stage and reasoning stage compared to other zero-shot grasping methods, utilizing only basic, mininal information and a single static RGB+D image.
- Through extensive experiments on a real-world robotic platform, we demonstrate that our proposed approach outperforms current state-of-the-art methods methods.

2.2 Related Works

Robotic grasping [48] aims to determine the best way to grasp objects and conventionally includes two approaches: the analytical approach [39, 47] and the data-driven approach [13, 49, 3]. The **analytical approach** aims to identify an appropriate grasp pose through analytic models that consider geometric conditions. For instance, [47] analyzes the geometry of the point cloud and identifies the appropriate grasping points based on a set of geometric conditions. Prior works have shown that understanding geometric information dramatically benefits robotic grasping [39, 47], as well as tasks including 3D geometry reasoning [32] and building synthetic tools [22]; however, these approaches require the availability of 3D geometry, which is computationally expensive and sensitive to noise, thereby restricting their applicability in real-world environments. On the other hand, the **data-driven approach** [13, 49, 3, 1] train models to predict grasp points. Prior works propose end-to-end grasp detection networks for partial, noisy point clouds [49, 3] and for leveraging semantic information [13] in a supervised manner. In addition, techniques based on leveraging a semantic knowledge graph [29, 4, 5], shape segmentation techniques [20], and a physics simulator [10] have been considered. These approaches are limited in their ability to generalize to unseen objects due to requiring computationally expensive training.

Besides the approaches mentioned above, LLMs and VLMs (Vision-Language Models) have recently been successfully utilized in robotics grasping tasks due to their reasoning ability [25, 33, 37, 38]. Closest to our work is LERF-TOGO [33], which utilizes a vision-language model in a zero-shot fashion to output a ranking of sampled grasping points given a natural language

query indicating the object as well as the part-name that should be grasped. Specifically, LERF-TOGO reconstructs a 3D object mask based on DINO embeddings [7] and then uses it to output a ranking of grasping points via language models with conditional LERF [15]. While LERF-TOGO requires detailed natural language queries containing the object and object part name, as well as many views to reconstruct the scene and 3D mask, our approach, `ShapeGrasp`, is able to generate grasping points with minimal semantic information about the object, based solely on a single passive RGB+D top-down view of the scene. `ShapeGrasp` achieves this by decomposing the object into a graph containing geometric shapes, their spatial relationships, and basic attributes, before utilizing the graph as a prompt component for an LLM. Thus, our approach leverages the advantages of the analytic approach—*understanding geometric information* [9]—and the large models-based approach—*the zero-shot reasoning ability* [11]—which has been shown to be effective.

LLMs and VLMs have successfully been applied to the robotics domain [2, 21, 12, 36, 50], including task-oriented grasping [33, 27]. In such context, LLMs have been integrated into planning procedures in various ways, such as providing semantic knowledge [2] and performing complex reasoning in the form of an inner monologue [12]. In addition, [21] leverages the zero-shot reasoning capabilities of VLMs for the pick-and-drop task requiring object detection, navigation, and grasping. Despite the benefits of utilizing LLMs and VLMs, which include no need for additional training and providing commonsense reasoning capability, naive utilization of LLMs and VLMs still have limitations stemming from their inherent shortcomings, such as indecisiveness, lack of domain knowledge, hallucination, and the black-box problem [31]. To address these limitations and enhance reasoning, we infuse LLMs with structured knowledge in the form of a symbolic graph that captures an object's geometric composition by describing, among other properties, the shape and size of decomposed convex parts, and the spatial relationships between them. This infusion has been shown to be effective in preventing LLMs from deviating into the realm of fictitious information, thereby ensuring a connection to factual data [9, 26, 31, 44] and enhancing reasoning capabilities for task-oriented grasping.

2.3 Shape-Based Grasping

In this section, we introduce `ShapeGrasp`, our approach to zero-shot task-oriented grasping of novel objects by leveraging a graph of basic geometric shapes that compose the object. Given an RGB+D input image $\boldsymbol{I} \in \mathbb{R}^{H \times W \times C}$, our approach $\boldsymbol{g}, \theta = f_{SG}(\boldsymbol{I})$ estimates a grasp location $\boldsymbol{g} \in \mathbb{R}^3$ and rotation $\theta \in \mathbb{R}^1$ for the robot to pick up the object.

The function $f_{SG}(\dots)$ represents our modular grasping pipeline, `ShapeGrasp`, composed of the following modules: First, we introduce our approach to segmenting and retrieving convex decompositions of the target object (see Sec. 2.3.1) by using a pipeline of pre-trained models. Then, we discuss our first contribution, selecting a suitable decomposition through an automatic heuristic (see Sec. 2.3.2). Finally, we introduce our novel approach of reasoning over the geometric composition of the target object by using a graph-based representation of the object's decomposed parts

and utilizing an LLM for a multi-stage reasoning process (see Sec. 2.3.3).

2.3.1 Segmenting the Object

In this section, we introduce our image-processing pipeline utilizing SAM [17] to retrieve an object mask and CoACD [42] to generate a convex decomposition of the object.

Retrieving the Object Mask and Point Cloud

To obtain the segmentation mask for input image $\boldsymbol{I}$, we utilize user input defining a set of *"in-points"* on the target object, used by *Segment Anything Model* (SAM) to retrieve the object's full 2D segmentation mask $\boldsymbol{M}$. Following this step, in-points are discarded, and a point cloud $\mathcal{P}$ of the object is retrieved utilizing the mask and depth information in our input image $\boldsymbol{I}$. We posit that the particular method for retrieving such object masks is not central to our overall approach and can be facilitated by various alternative approaches.

Convex Decomposition with CoACD

Given an object mask $\boldsymbol{M}$ and point-cloud $\mathcal{P}$, we independently retrieve a convex decomposition for each of the two inputs. To this end, we utilize CoACD, which is a recent approach for the convex decomposition of 3D meshes, specifically designed to retain fine-grained object features, which are important to preserving the original functionality, particularly in interactive settings. To utilize this approach, we convert the 2D mask $\boldsymbol{M}$ into a 3D mesh by interpreting it as a plane and retrieving a mesh from point cloud $\mathcal{P}$ through voxelization. The depth data utilized in the 3D pipeline allows for more intricate decomposition with CoACD, uncovering features at various elevations within the object that might be overlooked by the 2D method. However, 2D decompositions are robust to depth inaccuracies and provide a fast approximation that can be beneficial, particularly for concave objects like mugs. We compute two separate convex decompositions, one from the 2D mask and one from the 3D point cloud, termed $\mathcal{C}_{2D}$ and $\mathcal{C}_{3D}$, respectively, as each decomposition exhibits a set of different desirable properties. Selecting the appropriate decomposition depends on the object the system interacts with. In the next section, Sec. 2.3.2, we discuss our automated heuristic utilized for this purpose.

2.3.2 Geometric Decomposition

In this section, we discuss the following contributions: a) our heuristic approach to finding a suitable decomposition of a particular object, and b) how we create object-graph $\mathcal{G}$ that relates part shapes with each other and stores their attributes.

Heuristic Selection of Decomposition

To address the problem that certain objects (e.g., concave mugs) are inherently unsuitable for convex decomposition, while other objects may have little visual features or reflective surfaces that result

Figure 2.3: `ShapeGrasp` Prompting: Given a geometric decomposition graph, we infer a suitable grasping point through a chain of four consecutive prompts: two for semantic part identification, and two for task-oriented reasoning and selection. Ablations of these prompts can be found in Table 2.3.

in low-confidence depth maps, we introduce a heuristic $\mathcal{C}^* = h(\mathcal{C}_{2D}, \mathcal{C}_{3D})$ to choose the optimal decomposition $\mathcal{C}^*$ from the two described decomposition procedures. We first lower the decomposition threshold from initial value γ until both the 2D and 3D pipelines result in more than a single part. After that, our heuristic $h(\dots)$ chooses a preferred decomposition given the following criteria: If the 3D decomposition $\mathcal{C}_{3D}$ results in more than a set threshold of parts, ω, or if the percentage of depth points with high confidence α is too low, our heuristic $h(\dots)$ chooses the 2D decomposition $\mathcal{C}_{2D}$. Formally, $h(\dots)$ selects between 2D and 3D decompositions at valid thresholds as follows:

$$h(\mathcal{C}_{2D}, \mathcal{C}_{3D}) = \begin{cases} \mathcal{C}_{3D}, & \text{if } |\mathcal{C}_{3D}| \leq \omega \ \wedge \ \text{conf.} \geq \alpha \\ \mathcal{C}_{2D}, & \text{otherwise} \end{cases} \tag{2.1}$$

where $|\mathcal{C}|$ is the number of parts found in the respective decomposition. Section 2.4.1 provides empirical evidence for our heuristic parameters. With the selected decomposition $\mathcal{C}^*$, we create the object graph $\mathcal{G}$ that describes the object's decomposed parts and their spatial relationships.

Structured Object-Graph Creation

After selecting an appropriate decomposition $\mathcal{C}^*$, we project it back onto the original input image $\boldsymbol{I}$ and create an object-graph $\mathcal{G}$ describing the composition of the target object. Each decomposed part is represented as a node in the graph accompanied by attributes derived from the segmented image. The primary attribute is an approximating shape primitive, chosen from an isosceles triangle, rectangle, circle, or ellipse. To select an appropriate shape primitive, we approximate each convex hull with a simplified polygon given a pre-defined approximation threshold ε. The resulting points from this simplification dictate the fitted shape as follows:

- **Isosceles triangles** are formed by modifying any three points to equalize the leg lengths and adjust the base.
- **Rectangles** are formed by finding the rotated rectangle of the minimum area enclosing part.
- **Circles** are identified by measuring the shape factor, or the ratio of the part's area to the area of the bounding circle, with a threshold of 0.9 for circle classification.
- **Ellipses** are formed by fitting the part inside a rectangle (see above) if the fit reduces errors further.

These geometric shapes allow for the determination and inclusion of additional attributes, namely the aspect ratio, calculated based on the long and short sides of each shape (or major/minor axes), and the angle of the long side. The original decomposed convex hulls provides the centroid and area attributes. Object color is also incorporated an as attribute, and is derived by bucketing the RGB color spectrum based on the standard 16 web colors and selecting the most prevalent color. Edges within graph $\mathcal{G}$ are drawn to connect nodes whose convex parts share boundaries or intersect in the segmentation. The length of the connection is included as the edge attribute.

Figure 2.4: Overview of the 38 experimental objects from [33].

2.3.3 Grasp Inference through Shape Reasoning

To select an appropriate grasping point for the object, we propose to leverage the commonsense knowledge encoded in LLMs across two interaction stages, inspired by the Chain of Thought approach [41], further improving our approach's reasoning capabilities: 1) semantic reasoning over each shape described in the graph $\mathcal{G}$ and 2) selecting the most appropriate shape that facilitates the task-oriented grasp. We leverage a prompt template, described in Fig. 2.3, that depends only on the target object, desired task, and constructed graph $\mathcal{G}$. To ensure the intended output structure, we utilize TypeChat [24] in addition to the presented prompts.

Semantic Part Identification

In the first stage, given the target object name, task, and constructed graph, the LLM is instructed to reason about the nodes in the graph and what semantic part (e.g., "handle" and "blade", for a knife) each represents in the target object. To conduct this reasoning, we first ask for an unstructured, free-form answer in which the LLM explicitly explains its thoughts. As a follow-up to this response, the LLM is tasked to assign a single semantic part to each graph node in a structured manner.

Task-Oriented Part Selection

After the semantic reasoning is complete, the LLM is instructed to reason about the task utility of each part, given its graph representation and assigned semantics from the first stage. Similar to the first stage, this is accomplished in two steps: a free-form reasoning and explanation stage, which the LLM then uses to assign a final task-oriented suitability score to each node, which determines the selected grasp.

2.3.4 Selecting a Grasp Pose

In the final step, we select the graph node with the highest predicted score. To facilitate the grasp, we calculate the centroid of the chosen part and derive its corresponding 3D coordinates from the depth information in input image $\boldsymbol{I}$. To consider rotations, we calculate the principle components of the masked subpart in the point cloud and grasp along the largest component.

2.4 Experiments

We evaluate our approach in real-world experiments and demonstrate its effectiveness in grasping a diverse range of household objects across a variety of tasks. Figure 2.4 presents the 38 objects from 12 different categories used in the 49 tasks in our experiments. Section 2.4.1 discusses our heuristic that both dynamically sets the 2D and 3D decomposition's threshold (Sec. 2.4.1) and selects between their final outputs (Sec. 2.4.1). Section 2.4.2 demonstrates our approach on a real-world robotic platform and compares it against state-of-the-art baselines, and also includes additional qualitative experiments in section 2.4.2 and an LLM ablation study in section 2.4.2. All of our experiments are conducted with a Kinova Jaco robotic arm equipped with a three-finger gripper, coupled with a fixed Oak-D SR passive stereo-depth camera for RGB and depth perception.

Pipeline Configuration: To retrieve convex decompositions of sufficient quality, we empirically set the threshold ω, deciding when to select the 2D decomposition over the 3D decomposition, to $\omega = 10$. Additionally, we set parameter α, defining the minimal percentage of depth points that exhibit high confidence, to $\alpha = 85\%$. We note these values are not sensitive and are set emprically without intricate tuning.

Dataset: We recreate a dataset of 38 objects covering 12 general categories and 49 tasks (Fig. 2.4) introduced by LERF-TOGO [33], which we refer to for object and task details.

Metrics: To evaluate the effectiveness of our approach, we employ three metrics: "Part Identification", "Part Selection", and "Lift Success". "Part Identification" measures the accuracy of the semantic part assignment, across all parts globally and across only the ground truth parts. "Part Selection" quantifies the proficiency of our model to select the correct part for the task-oriented grasp. For this metric, the selection is deemed correct if it aligns with ground-truth parts based on LERF-TOGO [33]'s part queries and commonsense human judgment with respect to the task, safety, and stability. "Lift Success" indicates the percentage of successfully lifted objects at the selected part for the given task. Incorrectly selected parts and failed grasps at correctly selected parts are both counted as failures here.

2.4.1 Automatic Geometric Decomposition

Dynamic Threshold Selection

In both the 2D and 3D pipelines, the convex decomposition threshold γ, that controls the number of decomposed object parts, is an important parameter grasping success is sensitive to. This threshold

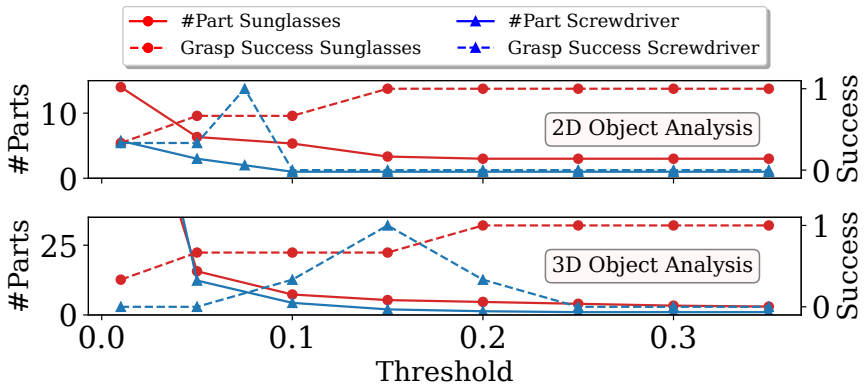

Figure 2.5: Threshold analysis for 2D (top)/3D (bottom) decompositions on sunglasses (complex) and screwdriver (simple).

balances the accuracy of the decomposition with the manageability of the resulting segmentation. A high error threshold $\gamma > 0.2$ that does not over-segment a geometrically complex object may fail to decompose a simple object – the entire object may be approximated as a single convex hull. Conversely, a low error threshold $\gamma < 0.1$ may decompose complex objects into overly many parts, complicating the resulting graph and making reasoning challenging. As shown in Figure 2.5, we run the pipeline at thresholds between 0.01 and 0.35 and evaluate the number of decomposed parts and the lifting success rate on sunglasses and screwdriver, respectively representing geometrically complex and simple objects, as evidenced by the consistent larger number of decomposed parts for the sunglasses at each threshold. This experiment reveals that any single threshold may not be sufficient to accommodate all objects. At thresholds conducive to successful reasoning over and grasping the sunglasses, the screwdriver often fails to properly decompose; naively taking the centroid of an entire object as the grasp point does not allow room for reasoning and is inherently unsafe, which we count as a failure. While this threshold may easily be set and tuned by a user, our heuristic fully automates the inference process.

Our pipeline provides a generalizable zero-shot approach for grasping by balancing object complexities and geometries. We propose the algorithm described in Section 2.3.2 that sets the object-specific threshold γ. Based on the above analysis, an initial threshold is set at $\gamma = 0.2$ for 3D meshes and $\gamma = 0.15$ for 2D meshes. It is reasonable to start with a higher error allowance for 3D objects due to the noise and dimension introduced by depth perception. Intuitively, we begin with a high threshold, which generally avoids over-segmenting complex objects but may under-segment simple ones. To address this, we iteratively decrease the threshold by 0.025 for each object until a valid decomposition of two or more parts is achieved. For example, a valid decomposition and successful lift are achieved within 2-3 iterations for the screwdriver, while the initial threshold is satisfactory for sunglasses. We further find that in our experiments, high thresholds that do not decompose an object execute in less than a second, making this search both efficient and effective. It is important to note that while this heuristic can dynamically adjust to objects of varying geometries, it cannot guarantee that decomposed parts are semantically reasonable.

Model *(using GPT-4)*	Part Selection
1 `ShapeGrasp` (2D only)	0.86
2 `ShapeGrasp` (3D only)	0.73
3 `ShapeGrasp` (Heuristic, no obj)	0.51
4 `ShapeGrasp` (Heuristic)	**0.92**

Table 2.1: Part selection accuracy of `ShapeGrasp` variants. For "no obj" we omit the object name from the prompt.

2D vs 3D Decompositions

After setting appropriate thresholds, the choice between the resulting 2D and 3D decompositions used to build the graph is a critical step. As discussed in Section 2.3.2, the heuristic algorithm we employ uses the depth confidence and number of decomposed parts (too many parts may indicate noisy, concave, or overly complex surfaces) to facilitate this choice. Table 2.1 shows the "Part Selection" performance across all object-task pairs between the 2D and 3D pipelines and heuristic that selects between them, each of which uses the heuristic threshold search. An interesting observation is that while our heuristic selection does indeed result in the best performance (92%), the 2D pipeline exhibits stronger performance (86%) than the 3D pipeline (83%). While this may be initially counterintuitive, this result demonstrates and is consistent with the conclusion that the flexibility of our pipeline allows us to dynamically adapt to settings where depth information may be inconsistent, low quality, or incompatible with the convex decomposition method. In our real-world experiments, while the Oak-D SR passive depth camera provides high-quality depth estimates in well-lit environments on matte objects with noticeable elevated features—the same features we may care to segment for grasping—results deteriorate with concave, reflective, and transparent surfaces, where the 2D pipeline excels due to the sole reliance on RGB data. The complexity and prevalent noise in real-world settings often necessitates a reliance on 2D decompositions for accuracy and robustness while still leveraging depth data in 3D decompositions when it is confident and useful via the heuristic.

The threshold search, coupled with the automatic selection between the 2D and 3D pipelines, forms an algorithm that determines decomposition outputs that are suitable and useful for semantic reasoning and grasping.

2.4.2 Zero-Shot Task-Oriented Grasping

For real-world task-oriented grasping experiments, we pair the `ShapeGrasp` pipeline with a Kinova Jaco arm designed to grasp and lift objects at the selected part (see Table 3.1). This setup employs a straightforward method for selecting grasp poses drawing directly from the object graph and masked point cloud (see 2.3.3) to execute a top-down grasp.

We employ two baselines to evaluate the efficacy of our proposed approach `ShapeGrasp` on the "Part Selection" and "Lift Success" metrics: **GraspGPT [37]**, which is a current state-of-the-art

Model	Part ID	Part Sel.	Lift Succ.	Time
1 GraspGPT [37]	N/A	0.37	0.31	150
2 GPT4-Vision [30]	N/A	0.82	0.73	20
3 `ShapeGrasp` (Starling)	0.54 (0.63)	0.65	0.57	25
4 `ShapeGrasp` (GPT-4)	**0.84** (**0.90**)	**0.92**	**0.82**	30

Table 2.2: Results on `ShapeGrasp` compared to GraspGPT and GPT4-Vision baselines. *Part ID* is the identification accuracy across all parts (and across target parts) and *Time* is the typical inference time in seconds.

approach for zero-shot task-oriented grasping and **GPT4-Vision [30]**, a foundation model trained with internet-scale data with visual input modality, prompted with language to select the correct task-oriented part.

Our empirical findings indicate a significant performance advantage of our method over GraspGPT [37], underscoring the efficacy of our structured, symbolic object part graph in conjunction with LLM reasoning. The performance gain using our pipeline over GraspGPT [37] is 55% and 51% (see rows 1 and 4 in Table 3.1) for the "Part Selection" and "Lift Success" metrics, respectively. We further analyze the discrepancy between `ShapeGrasp` and GraspGPT results; we note that GraspGPT depends on GraspNet [27] for grasp sampling, which may be inaccurate or fail on certain objects when the depth quality is noisy or poor, which may occur due to our static monocular depth camera. While GraspGPT is limited to tasks and objects related to previously known concepts, `ShapeGrasp` demonstrates robustness to the same noisy depth inputs, while featuring zero-shot and being more lightweight than GraspGPT (see "Time" in Table 3.1).

An important hypothesis that motivates our vision-based pipeline that constructs the object graph is that directly processing object part features and spatial relationships, and providing this information in a structured way for LLM reasoning, is more robust and performant than relying on VLMs for end-to-end reasoning. Though VLMs are considerably larger and more expensive models, performance on low-level features and relationships within parts of an object image may be unreliable and subject to hallucinations [46].

To test this hypothesis and directly compare our graph-construction and reasoning pipeline to a VLM, we establish a privileged GPT4-Vision baseline that benefits from the same heuristic-selected object segmentations and skips the graph-based reasoning stages (contributing to the faster runtime). This baseline is grounded by coloring each part and assigning them integer index labels for clarity. We confirm GPT4-Vision's capability to interpret segmented and grounded input object images through a series of questions and human-verified responses. We provide the VLM with the same object, task, and instruction and use the same grasping method to ensure comparability. Our method shows significant success rate gains over GPT4-Vision, by 10% and 9% on the evaluation metrics (see rows 2 and 4 in Table 3.1).

We further test the modularity of `ShapeGrasp` by evaluating the full pipeline using Starling [51], a much smaller and more efficient open-source LLM, as the inference backend instead

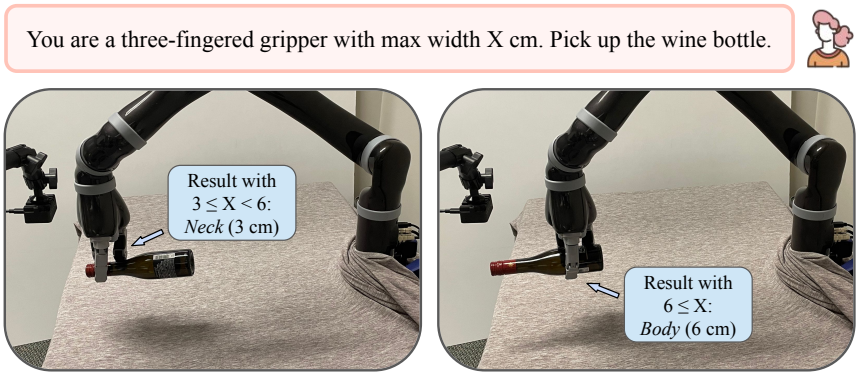

Figure 2.6: `ShapeGrasp` results incorporating width attributes and variable gripper width constraints.

of GPT-4. Performance across all metrics, while lower than the larger and more powerful GPT-4, remain meaningful and higher than the GraspGPT [37] baseline. Additional experimental results are on the project website: `https://shapegrasp.github.io/`.

Qualitative Results

The flexibility and generalizability of `ShapeGrasp` allow us to explore more complex interactions by incorporating additional object and robot attributes. For example, Fig. 2.2 shows how LLM semantic reasoning over our shape graph enables effective execution of the "hand over" task. As the LLM understands which node in the graph corresponds to the handles in objects (semantic identification stage), it can prioritize that part in the "hand over" interaction (task-oriented selection stage). Additional attributes can also be incorporated and reasoned over in an object-specific way; while both the "mug" and the "soldering iron" are given the attribute of being "hot", the task-oriented reasoning stage can make the commonsense inference that the level of heat and risk exhibited by these two objects differ dramatically. For the "hand over" task, while the mug is grasped by the hot body, which "minimizes the risk of spilling hot liquid and ensures a comfortable handover", the soldering iron is still grasped by the handle, which "positions the hot tip away from both the robot and the human" (see Fig. 2.2).

Fig. 2.6 demonstrates how the task-oriented selection stage is also able to consider robot attributes. Here, a "width" attribute is added to each node in the object graph, along with a variable maximum gripper width in the prompt. While the robot prefers to "pick up the wine bottle" by the bottle body, when the body width exceeds the maximum gripper width, it switches to the narrower neck, ensuring a successful grasp.

LLM Ablations

To validate the efficacy of our model and the comprehensiveness of the LLM interaction stages, we conducted an ablation study with a focus on the "Part Selection" evaluation metric, as detailed

	Task Reasoning	Part Identification	Part Selection
1			0.43
2	✓		0.57
3		✓	0.84
4	✓	✓	**0.92**

Table 2.3: Ablation study on `ShapeGrasp` LLM reasoning. While a part to grasp is always produced, reasoning can be done without identifying parts and/or without reasoning over the task. See Fig. 2.3 for detailed prompt information.

in Table 2.3. Applying the previously described heuristic decomposition selection algorithm, we examine the impact of each LLM reasoning stage: the *semantic part identification* stage and *task-oriented part reasoning* stage that comes before the final task score assignments. We evaluate the results using neither reasoning stage (LLM directly assigns only task scores), only one reasoning stage, and the full reasoning procedure.

The ablation study indicates that the most effective performance is achieved when both the part identification and task reasoning stages are employed in the LLM interaction (92%). The part identification stage is empirically the more important of the two stages, leading to a performance of (87%). This stage is likely important because when employed, the LLM's final selection is forced to depend on its own semantic part assignments. Intuitively, the semantics of a part are essential in determining its affordances and suitability for a task. This argument is further supported by the accuracy in the part identification stage (see Table 3.1), 84% globally, and 90% across the ground truth parts. These numbers lead to an important observation: not every part in an object, or even the ground-truth part, needs to be correctly identified; it may be sufficient if even a single critical part to either grasp or avoid is correctly identified. For example, in a pair of blue sunglasses, the ground truth part "arm" was misidentified as the "frame", but the "lens"—which is critical to avoid—was correctly identified. As such, the task-oriented selection stage correctly prioritized avoiding the "lens" segment. Put in another way, the quality of the automatic heuristic decomposition and the accuracy of the part identification stage effectively provides another novelty: *semantic part segmentation*, which naturally may be correlated with task-oriented part selection. Note that the "Part Identification" and "Part Selection" accuracies are closely coupled with both the GPT-4 and Starling models.

Including the task-oriented reasoning stage demonstrates a notable performance gain as well, both when the part-identification stage is ablated (14%) and included (8%); this may indicate that even once semantics are assigned, explicit task-conditioned reasoning is still beneficial, especially as a single object may have numerous appropriate uses and tasks.

In Table 2.1, we also explore the performance when the target object name is withheld. This leads to a significant performance drop (51%), likely due to the challenge of assigning semantic parts to an unknown object and highlighting `ShapeGrasp`'s effective use of this basic information.

In this setting, assigned semantics may not be meaningful, with the results largely determined by geometric attributes.

Chapter 3

SemanticCSG: Zero-Shot Task-Oriented Grasping via Symbolic Grounding of LLM Hypotheses

3.1 Introduction

Robots operating in real-world environments must navigate unstructured settings [16], particularly in ways that align with human intuition for effective interaction. Homes and other everyday environments present novel objects and tasks that a system must handle. *Task-oriented grasping* (TOG)—grasping objects to facilitate a given task—is particularly challenging. To interact with a novel object, a system must understand its purpose and reason about both its overall utility and that of its individual parts. For example, when asked to pick up a pair of sunglasses, the robot should recognize the distinct parts—lenses and temples—and determine that grasping the temples is the safest and most effective approach without damaging the glasses.

Large Language Models (LLMs) provide world knowledge and commonsense hypothesis generation for TOG [28, 33, 19]. However, they struggle with spatial reasoning [14] and grounding their open-ended knowledge, particularly in 3D, which is essential for robust inference under varying viewing angles and occlusions. These methods can also be computationally intensive and require additional external information, such as predefined object part names [28, 33], limiting their general applicability and zero-shot performance.

To address these limitations, we propose `SemanticCSG`, a robust framework for zero-shot 3D semantic segmentation and TOG that grounds LLM-generated hypotheses by optimizing CSG trees to fit observed object instances. CSG trees are structured representations composed of 3D shape primitives (e.g., spheres, cylinders, cuboids) combined using Boolean operations (union, intersection, difference). They provide a natural framework for LLM-driven hypothesis generation, enabling the LLM to infer how fundamental shapes and operations compose diverse objects. By optimizing this hypothesis to match real-world observations, our approach aligns semantics with geometry, mirroring how humans integrate prior knowledge with novel 3D observations to infer part affordances and utility [8].

Figure 3.1 provides a high-level overview of our approach. Starting with an RGB image, we use off-the-shelf models to segment the target object [17] and lift it to 3D [40]. We separately obtain a

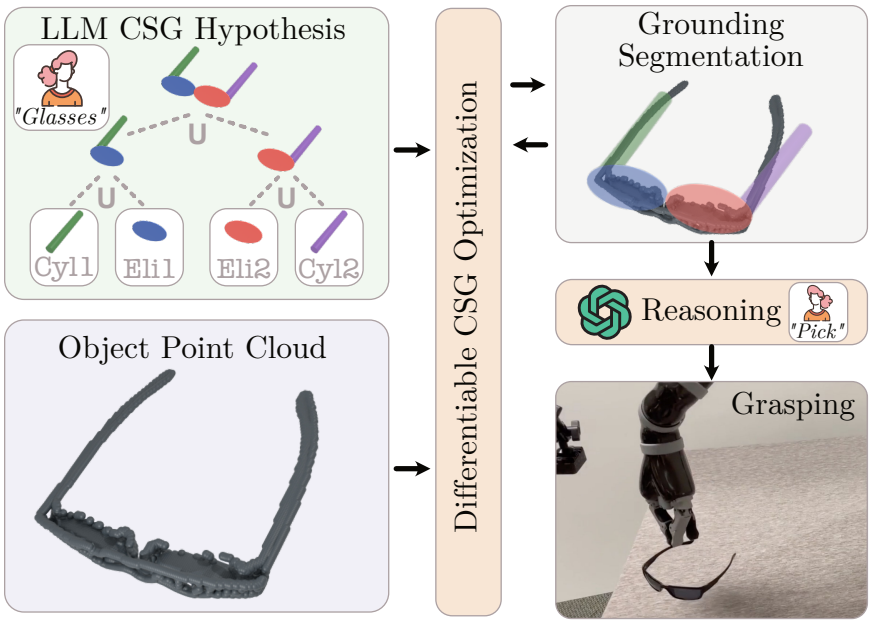

Figure 3.1: Overview of our CSG hypothesis generation, semantic grounding, and grasp selection pipeline.

CSG hypothesis by querying an LLM to infer how the object's 3D geometry relates to a set of shape primitives. `SemanticCSG` constructs and evaluates the hypothesis CSG tree, optimizing the shape primitives' parameters to fit the observed object. Once aligned, the grounded CSG tree is used to partition the object's point cloud and assign attributes for 3D semantic segmentation. Finally, the LLM leverages the symbolic structure as additional context for task-oriented reasoning, identifying the most suitable part to grasp. This process enhances both the representational power and reasoning capabilities of our method. Our key contributions include:

- We introduce `SemanticCSG`, a novel approach for zero-shot 3D semantic segmentation and grasping that leverages LLMs to generate structured hypotheses about object parts, their utilities, and relevance to grasping.
- Our method enables efficient visual understanding and reasoning by grounding LLM-generated hypotheses in a symbolic CSG tree, requiring minimal user input.
- Extensive real-world experiments demonstrate that our approach outperforms state-of-the-art methods in challenging single-view settings by generating and optimizing a CSG tree as part of the inference pipeline.

3.2 Related Works

LLMs and VLMs have been widely applied in robotics for their reasoning ability [25, 50, 36], including for TOG [33, 37]. In this context, LLMs have been integrated into planning in various ways, such as providing semantic knowledge and performing complex tasks via inner monologue [12].

However, many of these approaches require extensive training, multi-view reconstructions, or expensive inference, and suffer from hallucination, lack of domain knowledge, and interpretability challenges [31].

3D Semantic Segmentation enables fine-grained object understanding by jointly encoding vision and language signals. Recent deep learning approaches leverage large-scale datasets to learn language-aligned 3D feature embeddings, enabling open-vocabulary segmentation without human annotation. Unlike traditional methods limited to closed sets and vocabularies, these approaches allow zero-shot part segmentation, retrieving and segmenting parts from diverse objects based on arbitrary text queries. While models such as Find3D [23] demonstrate strong generalization, they require full 3D point clouds and lack structural priors for geometric reasoning. Similarly, 2D segmentation models [17] have shown impressive open-world segmentation capabilities but struggle with direct extension to 3D understanding.

Constructive Solid Geometry (CSG) is a procedural modeling technique that represents 3D objects as compositions of shape primitives (e.g., spheres, cubes, cylinders) using Boolean operations (union, intersection, difference). CSG trees provide a structured representation useful in CAD modeling, simulation, and 3D reasoning [18, 34]. The inverse problem of recovering CSG trees from 3D data has been explored in shape abstraction and optimization [35, 45], but these methods remain geometry-based and do not effectively bridge the gap to semantic reasoning and understanding.

ShapeGrasp [19], the work most closely related to ours, introduces a structured framework where an object's 2D convex decomposition [42] is represented as a symbolic graph to prompt an LLM. This approach focuses LLM reasoning, grounding semantic properties and enhancing reliability in TOG [4, 5, 9, 26, 31]. By integrating symbolic knowledge with classical vision techniques, ShapeGrasp improves geometric reasoning, semantic understanding, and grasping performance. However, its symbolic representation is restricted to 2D and assumes a fixed viewpoint where target parts remain unoccluded, limiting its robustness in real-world scenarios. Additionally, the convex decomposition is purely geometry-based, lacking semantic alignment, which can lead to suboptimal decompositions and incomplete semantic assignments. These constraints make ShapeGrasp less effective in complex 3D settings, where occlusions and varying viewpoints challenge its reasoning pipeline.

In contrast, `SemanticCSG` introduces a novel 3D-aware symbolic representation that builds on ShapeGrasp's structured reasoning framework while extending beyond prior work by incorporating semantic knowledge. Our method optimizes a differentiable CSG representation, enabling direct alignment with real-world objects. This approach combines geometric reasoning with LLM-driven zero-shot task understanding, leveraging the strengths of both symbolic and data-driven neural methods. It overcomes the limitations of ShapeGrasp in two key ways: (1) it handles occlusions more effectively, and (2) it can decompose arbitrary 3D shapes.

3.3 Methodology: SemanticCSG

We introduce `SemanticCSG`, which grounds LLM hypothesis about objects and how to interact with them by constructing and optimizing CSG tree representations for 3D semantic segmentation and zero-shot TOG. Given an object and task instruction, `SemanticCSG` queries an LLM to propose a CSG hypothesis, refines it via differentiable optimization, and uses the LLM to reason over the aligned structure for task reasoning and grasp selection.

Given an RGB image $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3}$, our method predicts:

- A 3D semantic segmentation $S = \{S_1, \ldots, S_n\}$, where S_i is a disjoint subset of the object point cloud $\mathbf{P} \subset \mathbb{R}^3$.
- A grasp location $\mathbf{g} \in \mathbb{R}^3$ and rotation $\theta \in \mathbb{R}$, specifying how the robot should grasp the object for a given task.

Specifically, `SemanticCSG` is composed of the following modules: The visual processing module, which masks and lifts a 2D object image to a partial point cloud, from which we compute a ground truth signed-distance field (SDF) optimization target (see Sec. 3.3.1) through the use of pre-trained models and custom SDF generation procedure. The LLM hypothesis generation module, which provides semantic CSG tree representations of abstract objects (see Sec. 3.3.2). An optimization module, which differentiably reconstructs the CSG tree into a predicted SDF, computes a loss with respect to the target SDF, and optimizes the parameters of the CSG leaf shape primitives to match the ground truth target, thereby aligning and grounding the abstract representation to the observed instance (see Sec. 3.3.3). The semantic reasoning and TOG module, which employs the LLM to reason over the optimized CSG tree structured, leveraging the aligned semantic and geometric properties to determine the best task-conditioned grasp (see Sec. 3.3.4).

The core contribution of our method is the creation of hypothesis CSG trees, which extract and encode LLM knowledge in a structured form that can be differentiably reconstructed and optimized to fit real-world instances, thereby grounding LLM knowledge in observed 3D geometry.

3.3.1 Image Processing and Target SDF Generation

Our visual processing module computes a SDF from $\boldsymbol{I} \in \mathbb{R}^{H \times W \times 3}$. We first retrieve an object mask using [17], then lift it to 3D via monocular depth estimation [40], producing a set of object points $\mathbf{P} \subset \mathbb{R}^3$. We apply a custom sampling procedure to generate a ground truth SDF for optimization.

A SDF represents the distance of points in space to the object surface, with the sign indicating whether the point is inside ($S < 0$) or outside ($S > 0$) the object. Formally, we define the SDF as a function $S : \mathbb{R}^3 \to \mathbb{R}$ that assigns each point $\mathbf{x}$ the signed distance to the nearest surface point:

$$S(\mathbf{x}) = \operatorname{sgn}(\mathbf{x}) \cdot \min_{\mathbf{p} \in \mathbf{P}} \|\mathbf{x} - \mathbf{p}\|. \tag{3.1}$$

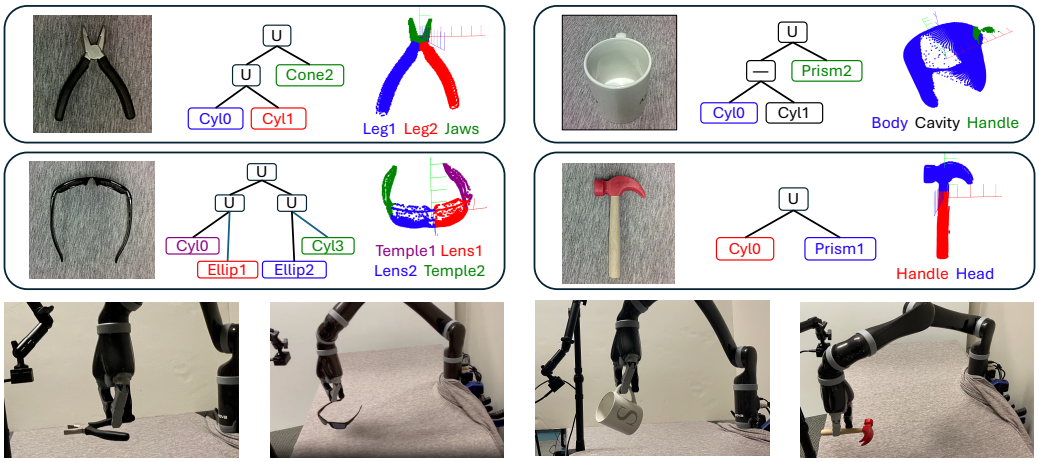

Figure 3.2: Example CSG trees, 3D semantic segmentation, and grasping results from the `SemanticCSG` pipeline.

By construction, surface points $\mathbf{P}$ form the zero level set:

$$S(\mathbf{p}) = 0, \quad \forall \mathbf{p} \in \mathbf{P}. \tag{3.2}$$

To get a denser field, we sample interior and exterior points within a local radius r around surface points, forming set $\mathbf{Q}$:

$$\mathbf{Q} = \{\mathbf{q} \in \mathbb{R}^3 \mid \|\mathbf{q} - \mathbf{p}\| \leq r, \mathbf{p} \in \mathbf{P}\}. \tag{3.3}$$

Each point $\mathbf{q} \in \mathbf{Q}$ is assigned a signed distance using $S(\mathbf{q})$, where the sign is determined by checking whether the point lies inside or outside the object contour. For interior points, we further differentiate between those above or below the nearest surface point. The final SDF is obtained by combining the elements across both $\mathbf{P}$ and $\mathbf{Q}$. With this ground truth SDF, we can compute a loss against a predicted SDF reconstructed from the hypothesis CSG tree, enabling optimization to fit the object's geometry (see Sec. 3.3.3).

3.3.2 LLM CSG Hypothesis Generation

We design an LLM-based module that, given an object name, instructs the LLM to relate its knowledge of an object's semantics and high-level geometry to a set of 3D shape primitives (e.g., cylinder, prism, cone). The LLM generates a hypothesis CSG tree, representing an abstract instance of the target object, allowing us to extract and structure LLM world knowledge in a form that can be aligned with geometry and observed instances. The CSG tree is a strict binary tree, where leaf nodes represent shape primitives parameterized by location, size, rotation, etc. Parent nodes apply Boolean operations (union, intersection, difference) to compose their two child nodes. This process holds until the final object is defined at the root (see Fig. 3.2).

To generate structured hypotheses, we use OpenAI's *o3-mini* model, which balances efficiency and generation capability. We enforce a structured output using TypeChat [24], ensuring hypothe-

seœs are serialized in a consistent JSON format using only our defined primitives and operations.

3.3.3 CSG Hypothesis Optimization

Given the target SDF and an initial hypothesis CSG tree, we optimize each shape primitive's parameters (location, scale, rotation) to align the abstract representation with the observed object's point cloud. This is achieved by differentiably reconstructing the CSG tree into a predicted SDF, computing a loss against the target SDF, and optimizing the parameters of the CSG leaf nodes to minimize this loss.

Differentiable CSG-SDF Reconstruction

Each employed shape primitive has an associated signed distance function $S_i : \mathbb{R}^3 \to \mathbb{R}$, parameterized by its center $\mathbf{c}_i$, scale $\mathbf{s}_i$, and rotation R_i. Boolean operations define parent nodes using standard SDF compositions:

$$S_{\cup}(\mathbf{x}) = \min(S_1(\mathbf{x}), S_2(\mathbf{x})) \tag{3.4}$$

$$S_{\cap}(\mathbf{x}) = \max(S_1(\mathbf{x}), S_2(\mathbf{x})) \tag{3.5}$$

$$S_{\setminus}(\mathbf{x}) = \max(S_1(\mathbf{x}), -S_2(\mathbf{x})) \tag{3.6}$$

Since these operations are differentiable, the full CSG tree defines a differentiable function $S_{\text{CSG}}(\mathbf{x}; \theta)$, where $\theta = \{\mathbf{c}_i, \mathbf{s}_i, R_i\}$ represents the shape parameters being optimized.

Coarse Alignment

Before refinement, we establish a coarse alignment between the predicted and real-world objects, as the reconstruction from the LLM's initial generated hypothesis may not align with the observed object. Given system and camera calibration, we define a world frame where the object's point cloud center and scale are estimated and normalized relative to the hypothesis point cloud (where $S_{CSG} \leq 0$). In our experiments, we assume a canonical object pose as a reference for alignment. For other applications, external object pose estimators can be used [6]. The real-world point cloud is then transformed to coarsely align with the hypothesis before refinement.

Multi-Stage Optimization

After coarse alignment, we refine the shape parameters by minimizing the SDF loss:

$$\mathcal{L}_{\text{SDF}} = \sum_{\mathbf{x} \in \mathbf{Q} \cup \mathbf{P}} \| S_{\text{CSG}}(\mathbf{x}; \theta) - S_{\text{target}}(\mathbf{x}) \|^2 \tag{3.7}$$

where S_{CSG} is the predicted SDF with the current shape parameters θ and S_{target} is the observed ground truth SDF.

To ensure stability, we optimize in stages:

1. Centers $\mathbf{c}_i$ are optimized first for positional alignment.
2. Scales $\mathbf{s}_i$ are adjusted next to refine proportions.
3. Rotations R_i are optimized last to tune orientation.

This staged optimization prevents overfitting and collapse, allowing the model to make *small, incremental adjustments* before handling more complex transformations.

Regularization for Semantic Consistency

Since multiple solutions can minimize $\mathcal{L}_{\text{SDF}}$, we introduce a regularization term to encourage solutions that remain close to the initial, coarsely-aligned hypothesis, ensuring that the optimized parameters stay semantically meaningful and do not deviate excessively from the LLM's prior knowledge:

$$\mathcal{L}_{\text{reg}} = \sum_i \|\theta_i - \theta_i^{\text{init}}\|^2 \tag{3.8}$$

where θ_i^{init} represents the initial hypothesis parameters.

Final Optimization Objective

The total loss function is a weighted combination of the SDF loss and regularization:

$$\mathcal{L} = \mathcal{L}_{\text{SDF}} + \alpha \mathcal{L}_{\text{reg}} \tag{3.9}$$

where α controls the influence of the regularization in preserving the semantic prior. At convergence, the optimized CSG tree aligns with the real-world instance and partitions the object's point cloud based on its constituent parts. The semantic and geometric properties from the CSG representation are then mapped to the segmented points.

3.3.4 Task-Reasoning and Grasp Inference

We adopt the LLM task reasoning framework used in ShapeGrasp [19]. This framework requires structured input, which our CSG trees provide, and operates in two stages. In the reasoning stage, the LLM leverages the aligned CSG tree with its semantic and geometric properties to evaluate each part's task utility, generating free-form explanations. It then assigns task-oriented likelihood scores to each leaf node in a structured JSON output, determining the most suitable part for grasping. The grasp pose at the selected part is directly computed from the geometric parameters of its associated shape primitive and executed by an end-effector controller.

3.4 Experiments

We evaluate our approach in real-world experiments across diverse household objects, tasks, and viewing angles/occlusions. Section 3.4.1 presents results on 3D semantic segmentation and LLM

Model	Part ID ↑				Part Sel. ↑				Lift Succ. ↑				Time ↓
	Easy	Medium	Hard	Avg	Easy	Medium	Hard	Avg	Easy	Medium	Hard	Avg	
1 GraspGPT [37]	–	–	–	–	0.36	0.27	0.23	0.29	0.27	0.18	0.18	0.21	150
2 GPT-4o [30]	–	–	–	–	0.86	0.45	0.41	0.57	0.77	0.41	0.36	0.51	**20**
3 Find3D [23] (+MCC [43])	–	–	–	–	0.64	0.59	0.50	0.58	0.55	0.45	0.41	0.47	180
4 ShapeGrasp [19]	0.88	0.43	0.31	0.54	**0.95**	0.50	0.36	0.60	**0.91**	0.41	0.36	0.56	30
5 `SemanticCSG`	**0.96**	**0.82**	**0.76**	**0.85**	**0.95**	**0.82**	**0.77**	**0.85**	**0.91**	**0.73**	**0.68**	**0.77**	25

Table 3.1: Performance of `SemanticCSG` and baselines on semantic segmentation (*Part ID*), task-oriented selection (*Part Sel.*), and grasping success (*Lift Succ.*) across three difficulty levels of viewing angle/occlusion, with inference time (seconds).

grounding. Section 3.4.2 evaluates affordance reasoning and task-oriented part selection. Section 3.4.3 examines grasp execution using a Kinova Jaco robotic arm with a three-finger gripper. We use an Oak-D SR camera for single RGB image capture. To assess performance under varying conditions, we evaluate across three difficulty levels based on *viewing angle and occlusions*:

- **Easy:** Clear, front-facing view with minimal occlusion.
- **Medium:** Off-angle view with moderate occlusion.
- **Hard:** Challenging perspective angle with significant occlusion, partially obscuring key features.

Dataset: We use a real-world dataset of 48 object and 66 task instances, from [33, 19], at the three difficulty levels.

Baselines: We employ a series of baselines to evaluate the efficacy of `SemanticCSG` for zero-shot TOG.

- *GraspGPT* [37]: Uses an LLM to relate novel object and task descriptions to training samples, reasoning about similar grasps from seen examples.
- *Find3D* [23]: A SOTA deep learning approach for open-vocabulary 3D semantic segmentation. Find3D requires a full object point cloud and predefined part names. Following their paper, we use MCC [43] to reconstruct the full object and provide the part names from our CSG tree. We use an LLM to select the task-relevant part.
- *GPT-4o* [30]: A foundation model that reasons over both language and vision. Spatial grounding remains a challenge. To address this, we use the convex decomposition identified by ShapeGrasp [19] to annotate the image, coloring each part and labeling it with an integer ID, enabling selections that map to real-world geometry.
- *ShapeGrasp* [19]: Uses convex decomposition [42] to construct a geometric object graph encoding object parts and their spatial relationships. This structured representation guides LLM reasoning for part affordance evaluation and task-oriented grasp selection.

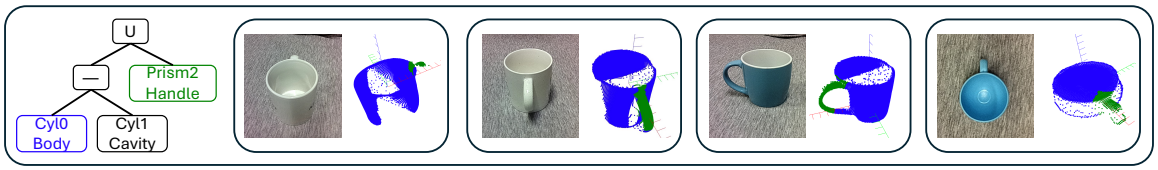

Figure 3.3: Qualitative outputs from `SemanticCSG`. The same hypothesis CSG tree, which is both 3D- and semantics-aware, can be robustly optimized to fit different instances of an object category across varying viewing angles and occlusions.

3.4.1 CSG Alignment and 3D Semantic Segmentation

Metrics: We evaluate LLM grounding and 3D semantic segmentation using the *Part Identification* (Part ID) metric, which measures how accurately an object is segmented into meaningful semantic parts and assigned correct labels, inferring its structure solely from a given object name.

Table 3.1 presents results for ShapeGrasp and `SemanticCSG`, the only methods that autonomously infer semantic parts. Other baselines rely on user-provided inputs: Find3D requires predefined part names, GPT-4o depends on annotated parts in the image, and GraspGPT ranks grasp poses without explicit semantic reasoning, making them unsuitable for the Part ID task. ShapeGrasp performs well in non-occluded cases; however, while `SemanticCSG` outperforms it by only 8% in these cases, the gap increases to 45% under more challenging conditions (see Table 3.1 lines 4 and 5). This decline highlights ShapeGrasp's reliance on a 2D decomposition that changes with the viewpoint, leading to inconsistencies despite the underlying object remaining unchanged. More complex geometries and occlusions further degrade performance, as convex decomposition struggles to represent such structures. `SemanticCSG`, in contrast, maintains stable performance across all tasks, including those with occlusions. The semantic and 3D-aware representation remains consistent across viewpoints, enabling stable segmentation even when parts are partially occluded or viewed from different angles. By optimizing directly in 3D space, `SemanticCSG` ensures robustness across diverse conditions. Figure 3.2 illustrates real-world objects, their CSG hypothesis trees, and the resulting 3D semantic segmentation after optimization. Figure 3.3 further demonstrates how `SemanticCSG` adapts to different object instances, viewpoints, and occlusions through its stable 3D representation and optimization, ensuring consistent and reliable segmentation.

3.4.2 Zero-Shot Task-Oriented Reasoning

We evaluate semantic reasoning and task-oriented grasp selection using the *Part Selection* (Part Sel.) metric, which measures how accurately a system selects the correct semantic part for a given task based on ground-truth parts from [33] and human judgment. When applicable, a part must be correctly identified to be successfully selected for the task.

Table 3.1 presents results across all methods. The trend from Part ID continues: ShapeGrasp and `SemanticCSG` perform similarly in non-occluded cases, but as the viewpoint becomes more challenging, `SemanticCSG` maintains stable performance while ShapeGrasp deteriorates. While

achieving the same success rate in the easy setting, SemanticCSG outperforms ShapeGrasp by a 41% margin in the hard setting (Tab. 3.1 lines 4 and 5). Further note that in the non-occluded case, ShapeGrasp's Part Sel. accuracy exceeds its Part ID accuracy by 7% (Line 4). This occurs because only the correct part needs to be identified for selection, even if other parts are misclassified. In contrast, SemanticCSG exhibits a stronger coupling between Part ID and Part Sel., as its optimization aligns the object globally.

GraspGPT performs about 55% worse than our method on average due to its reliance on retrieving similar grasps from training examples. In a zero-shot, single-view setting, it lacks the ability to generalize effectively to novel object instances and occlusions. Similarly, Find3D has an average success rate of 58% due to real-world 3D inputs falling outside its training distribution. However, like SemanticCSG, Find3D's performance only degrades by 14% as viewpoint changes, as it operates in a full 3D space rather than a view-dependent representation.

Between ShapeGrasp and the GPT-4o baseline (Table 3.1 line 2), which is privileged with the optimal decomposition identified by ShapeGrasp and visually annotated in the image, performance is similar. In the non-occluded case, ShapeGrasp outperforms GPT-4o by 9% due to its structured input, which provides a suitable context that constrains reasoning to task-relevant parts. As the viewpoint changes and the decomposition degrades, ShapeGrasp's symbolic representation becomes less reliable, leading to a drop in accuracy. By contrast, even receiving a suboptimal decomposition in the occluded case, GPT-4o still leverages pixel information to select the correct part, improving from -9% to $+5\%$ compared to ShapeGrasp (see lines 4 and 5). This aligns with our hypothesis: when the decomposition loses semantic meaning (e.g., a complete grid), GPT-4o's unconstrained reasoning allows it to rely on direct visual cues.

SemanticCSG outperforms all methods across all difficulty levels while maintaining high efficiency. By integrating symbolic knowledge with geometric grounding in a 3D structure, SemanticCSG provides a stable representation for affordance reasoning across varying viewing angles, occlusions, and object instances.

3.4.3 Task-Oriented Grasping Execution

Grasp Lift Success (Lift Succ.) measures the success rate of stable grasp execution at correctly selected parts. The GraspGPT and Find3D baselines predict grasps using GraspNet [27]. In contrast, similar to ShapeGrasp and GPT-4o, SemanticCSG infers grasp poses directly from the geometric parameters of symbolic parts, extended to 3D primitives. Methods relying on grasp pose estimation show a larger gap between selection and execution (e.g., Table 3.1 lines 3 and 5), as estimators struggle with occlusions and incomplete data in single-view settings, while SemanticCSG remains robust by leveraging a structured, 3D-aware representation.

This is particularly beneficial in cases where partial or noisy observations degrade estimator performance. While our end-effector control for grasp execution is straightforward, the results demonstrate that the utility of the CSG tree extends beyond semantics. The geometric parameters encoded in the 3D representation provide valuable structural information that aids in grasp

inference. Figure 3.2 illustrates this advantage: in the case of a mug where the handle is largely occluded, `SemanticCSG` successfully fits a 3D shape to the object. Since the system models the full 3D structure rather than relying solely on visible pixels, it can infer a grasp even in occluded regions, demonstrating robustness in reasoning over partial observations.

Chapter 4

Conclusion, Limitations, and Future Work

This work presents two complementary approaches for bridging perception and reasoning in robotic manipulation through symbolic object representations. `ShapeGrasp` demonstrates the effectiveness of lightweight 2D geometric abstractions for semantic reasoning, while `SemanticCSG` extends this idea into 3D, leveraging symbolic shape programs grounded in real-world geometry. Together, they show how structured representations can unlock zero-shot task reasoning and grasping in varied environments by connecting perceptual input to commonsense knowledge in large language models.

Despite these contributions, several limitations and open challenges remain. One key limitation lies in the granularity of symbolic abstraction. Both methods simplify complex objects into symbolic parts—convex 2D segments or primitive 3D shapes—which may not fully capture fine-grained details or nuanced geometry. In such cases, over- or under-segmentation can lead to mismatches between the symbolic structure and the actual physical object, affecting reasoning accuracy.

Our methods are most effective in object-centric settings where discrete parts and clear spatial structure are present. Performance may degrade in more complex or cluttered environments, especially those involving occlusions, deformable materials, or articulated objects. Such cases challenge the underlying assumptions of part-based decomposition and may require more adaptive strategies for parsing and reasoning about scene geometry.

Symmetry and geometric ambiguity also pose challenges. Since part semantics are inferred from a single RGB image, symmetrical or visually ambiguous objects can lead to incorrect or inconsistent labeling. This can result in plausible yet functionally inaccurate hypotheses, particularly when visual cues are limited or misleading.

Additionally, while `SemanticCSG` improves robustness over purely bottom-up approaches, it still depends on the initial quality of the symbolic hypothesis generated by the LLM. When the generated structure deviates significantly from the true object geometry—due to under-specified task input or ambiguous imagery—optimization may converge to suboptimal results.

Looking ahead, several promising directions could enhance the capabilities and generalization of our framework. Expanding the symbolic vocabulary to represent more complex, deformable, or articulated shapes would increase expressiveness. Combining symbolic reasoning with learned

visual features could improve robustness to noise and scene diversity. Incorporating these methods into a closed-loop perception and manipulation pipeline would allow robots to refine both geometry and semantics through interaction. Finally, hybrid approaches that dynamically switch between 2D and 3D representations depending on the task could offer an efficient trade-off between interpretability and fidelity.

Together, these contributions and future directions point toward a broader vision for intelligent, interpretable, and adaptable robotic systems. By connecting language-based reasoning with structured geometric understanding, this work takes a step toward enabling robots to act meaningfully in complex, real-world settings.

Bibliography

[1] Ananye Agarwal, Shagun Uppal, Kenneth Shaw, and Deepak Pathak. Dexterous functional grasping. In *Conference on Robot Learning*, 2023.

[2] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[3] Antonio Alliegro, Martin Rudorfer, Fabio Frattin, Aleš Leonardis, and Tatiana Tommasi. End-to-end learning to grasp via sampling from object point clouds. *IEEE Robotics and Automation Letters*, 7(4), 2022.

[4] Sarthak Bhagat, Simon Stepputtis, Joseph Campbell, and Katia Sycara. Sample-efficient learning of novel visual concepts. In *Proceedings of The 2nd Conference on Lifelong Learning Agents*, 2023.

[5] Sarthak Bhagat, Simon Stepputtis, Joseph Campbell, and Katia P. Sycara. Knowledge-guided short-context action anticipation in human-centric videos. *ArXiv*, abs/2309.05943, 2023.

[6] Jan Kautz Stan Birchfield Bowen Wen, Wei Yang. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *Conference on Computer Vision and Pattern Recognition*, 2024.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[8] Hans P. Op de Beeck, Katrien Torfs, and Johan Wagemans. Perceived shape similarity among unfamiliar objects and the organization of the human object vision pathway. *Journal of Neuroscience*, 28(40), 2008.

[9] Wenxuan Ding, Shangbin Feng, Yuhan Liu, Zhaoxuan Tan, Vidhisha Balachandran, Tianxing He, and Yulia Tsvetkov. Knowledge crosswords: Geometric reasoning over structured knowledge with large language models. *ArXiv*, abs/2310.01290, 2023.

[10] Kuan Fang, Yuke Zhu, Animesh Garg, Andrey Kurenkov, Viraj Mehta, Li Fei-Fei, and Silvio Savarese. Learning task-oriented grasping for tool manipulation from simulated self-supervision. *The International Journal of Robotics Research*, 39:202 – 216, 2018.

[11] Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey. *ArXiv*, abs/2212.10403, 2022.

[12] Wenlong Huang, Fei Xia, Ted Xiao, Harris Chan, Jacky Liang, Pete Florence, Andy Zeng, Jonathan Tompson, Igor Mordatch, Yevgen Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *Conference on Robot Learning*, pages 1769–1782. PMLR, 2023.

[13] Eric Jang, Sudheendra Vijayanarasimhan, Peter Pastor, Julian Ibarz, and Sergey Levine. End-to-end learning of semantic grasping. *arXiv preprint arXiv:1707.01932*, 2017.

[14] Subbarao Kambhampati. Can large language models reason and plan? *Annals of the New York Academy of Sciences*, 1534(1):15–18, 2024.

[15] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

[16] Alexander Khazatsky and Karl Pertsch et al. Droid: A large-scale in-the-wild robot manipulation dataset. *Robotics: Science and Systems*, 2024.

[17] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arXiv:2304.02643*, 2023.

[18] David H Laidlaw, W Benjamin Trumbore, and John F Hughes. Constructive solid geometry for polyhedral objects. In *Proceedings of the 13th annual conference on Computer graphics and interactive techniques*, pages 161–170, 1986.

[19] Samuel Li, Sarthak Bhagat, Joseph Campbell, Yaqi Xie, Woojun Kim, Katia Sycara, and Simon Stepputtis. Shapegrasp: Zero-shot task-oriented grasping with large language models through geometric decomposition. In *Intelligent Robots and Systems*, 2024.

[20] Yunzhi Lin, Chao Tang, Fu-Jen Chu, and Patricio A Vela. Using synthetic data and deep networks to recognize primitive shapes for object grasping. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10494–10501. IEEE, 2020.

[21] Peiqi Liu, Yaswanth Orru, Chris Paxton, Nur Muhammad Mahi Shafiullah, and Lerrel Pinto. Ok-robot: What really matters in integrating open-knowledge models for robotics. *arXiv preprint arXiv:2401.12202*, 2024.

[22] Zhijian Liu, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Physical primitive decomposition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–19, 2018.

[23] Ziqi Ma, Yisong Yue, and Georgia Gkioxari. Find any part in 3d, 2024. URL https://arxiv.org/abs/2411.13550.

[24] Microsoft. URL https://microsoft.github.io/TypeChat/.

[25] Reihaneh Mirjalili, Michael Krawez, Simone Silenzi, Yannik Blei, and Wolfram Burgard. Lan-grasp: Using large language models for semantic object grasping. *arXiv preprint arXiv:2310.05239*, 2023.

[26] Fedor Moiseev, Zhe Dong, Enrique Alfonseca, and Martin Jaggi. SKILL: Structured knowledge infusion for large language models. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

[27] Arsalan Mousavian, Clemens Eppner, and Dieter Fox. 6-dof graspnet: Variational grasp generation for object manipulation. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2019.

[28] Adithyavairavan Murali, Weiyu Liu, Kenneth Marino, Sonia Chernova, and Abhinav Gupta. Same object, different grasps: Data and semantic knowledge for task-oriented grasping. In *Conference on Robot Learning*, 2020.

[29] Adithyavairavan Murali, Weiyu Liu, Kenneth Marino, Sonia Chernova, and Abhinav Gupta. Same object, different grasps: Data and semantic knowledge for task-oriented grasping. In *Conference on robot learning*. PMLR, 2021.

[30] OpenAI. Gpt-4 technical report. 2023.

[31] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. Unifying large language models and knowledge graphs: A roadmap. *ArXiv*, abs/2306.08302, 2023.

[32] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1060–1070, 2020.

[33] Adam Rashid, Satvik Sharma, Chung Min Kim, Justin Kerr, Lawrence Yunliang Chen, Angjoo Kanazawa, and Ken Goldberg. Language embedded radiance fields for zero-shot task-oriented grasping. In *Conference on Robot Learning*, 2023.

[34] Jaroslaw R Rossignac. Constraints in constructive solid geometry. In *Proceedings of the 1986 workshop on Interactive 3D graphics*, pages 93–110, 1987.

[35] Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. Csgnet: Neural shape parser for constructive solid geometry. In *CVPR*, 2018.

[36] Simon Stepputtis, Joseph Campbell, Mariano Phielipp, Stefan Lee, Chitta Baral, and Heni Ben Amor. Language-conditioned imitation learning for robot manipulation tasks. *Advances in Neural Information Processing Systems*, 33:13139–13150, 2020.

[37] Chao Tang, Dehao Huang, Wenqi Ge, Weiyu Liu, and Hong Zhang. Graspgpt: Leveraging semantic knowledge from a large language model for task-oriented grasping. *IEEE Robotics and Automation Letters*, 2023.

[38] Chao Tang, Dehao Huang, Lingxiao Meng, Weiyu Liu, and Hong Zhang. Task-oriented grasp prediction with visual-language inputs. *arXiv preprint arXiv:2302.14355*, 2023.

[39] Andreas Ten Pas, Marcus Gualtieri, Kate Saenko, and Robert Platt. Grasp pose detection in point clouds. *The International Journal of Robotics Research*, 36(13-14):1455–1473, 2017.

[40] Ruicheng Wang, Sicheng Xu, Cassie Dai, Jianfeng Xiang, Yu Deng, Xin Tong, and Jiaolong Yang. Moge: Unlocking accurate monocular geometry estimation for open-domain images with optimal training supervision. In *Conference on Computer Vision and Pattern Recognition*, 2024.

[41] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[42] Xinyue Wei, Minghua Liu, Zhan Ling, and Hao Su. Approximate convex decomposition for 3d meshes with collision-aware concavity and tree search. *ACM Transactions on Graphics*, 41(4):1–18, July 2022. ISSN 1557-7368. doi: 10.1145/3528223.3530103. URL `http://dx.doi.org/10.1145/3528223.3530103`.

[43] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compressive coding for 3D reconstruction. *Conference on Computer Vision and Pattern Recognition*, 2023.

[44] Lin F. Yang, Hongyang Chen, Zhao Li, Xiao Ding, and Xindong Wu. Chatgpt is not enough: Enhancing large language models with knowledge graphs for fact-aware language modeling. *ArXiv*, abs/2306.11489, 2023.

[45] Fenggen Yu, Qimin Chen, Maham Tanveer, Ali Mahdavi Amiri, and Hao Zhang. D^2csg: Unsupervised learning of compact csg trees with dual complements and dropouts, 2023.

[46] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *ICLR*, 2023.

[47] Brayan S Zapata-Impata, Pablo Gil, Jorge Pomares, and Fernando Torres. Fast geometry-based computation of grasping points on three-dimensional point clouds. *International Journal of Advanced Robotic Systems*, 2019.

[48] Hanbo Zhang, Jian Tang, Shiguang Sun, and Xuguang Lan. Robotic grasping from classical to modern: A survey. *ArXiv*, abs/2202.03631, 2022.

[49] Binglei Zhao, Hanbo Zhang, Xuguang Lan, Haoyu Wang, Zhiqiang Tian, and Nanning Zheng. Regnet: Region-based grasp network for end-to-end grasp detection in point clouds. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13474–13480. IEEE, 2021.

[50] Yifan Zhou, Shubham Sonawani, Mariano Phielipp, Heni Ben Amor, and Simon Stepputtis. Learning modular language-conditioned robot policies through attention. *Autonomous Robots*, 47(8):1013–1033, 2023.

[51] Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, and Jiantao Jiao. Starling-7b: Improving llm helpfulness & harmlessness with rlaif, November 2023.