Semantically Safe Robot Manipulation: From Semantic Scene Understanding to Motion Safeguards

Lukas Brunke, Yanni Zhang, Ralf Römer, Jack Naimer, Nikola Staykov, Siqi Zhou, and Angela P. Schoellig

***Abstract*— Ensuring safe interactions in human-centric environments requires robots to understand and adhere to constraints recognized by humans as "common sense" (e.g., "*moving a cup of water above a laptop is unsafe as the water may spill*" or "*rotating a cup of water is unsafe as it can lead to pouring its content*"). Recent advances in computer vision and machine learning have enabled robots to acquire a semantic understanding of and reason about their operating environments. While extensive literature on safe robot decision-making exists, semantic understanding is rarely integrated into these formulations. In this work, we propose a semantic safety filter framework to certify robot inputs with respect to semantically defined constraints (e.g., unsafe spatial relationships, behaviors, and poses) and geometrically defined constraints (e.g., environment-collision and self-collision constraints). In our proposed approach, given perception inputs, we build a semantic map of the 3D environment and leverage the contextual reasoning capabilities of large language models to infer semantically unsafe conditions. These semantically unsafe conditions are then mapped to safe actions through a control barrier certification formulation. We demonstrate the proposed semantic safety filter in teleoperated manipulation tasks and with learned diffusion policies applied in a real-world kitchen environment that further showcases its effectiveness in addressing practical semantic safety constraints. Together, these experiments highlight our approach's capability to integrate semantics into safety certification, enabling safe robot operation beyond traditional collision avoidance.**

Fig. 1: We propose a semantic safety filter framework that leverages semantic scene understanding and contextual reasoning capabilities of large language models to certify robot motions with "common sense" constraints. For example, if a manipulator is carrying a cup of water, our proposed semantic safety filter prevents moving the cup above a laptop in the environment to prevent potential spillage *(top)*. On the contrary, if the robot is tasked to transport a dry sponge, it is allowed to move over a laptop *(bottom)*. An overview of the work with experiment demonstration results can be found on our website https://utiasdsl.github.io/semantic-manipulation/ and in our short video https://tiny.cc/semantic-manipulation.

I. INTRODUCTION

Safety is a key issue in robotics and has been gaining increasing attention across different communities [1], [2]. In safety-critical control, the goal is usually to guarantee set invariance (i.e., to prevent a system from leaving a certain safe set) [1]. Based on this definition of safety, various safety filters have been developed in recent years, which can be applied to detect unsafe control inputs and modify them into safe ones in a minimally invasive manner [2], [3]. Existing safety filters such as control barrier function (CBF) safety filters [4] or predictive safety filters [3] can provide theoretical safety guarantees in terms of set invariance. Still, they assume that the safety constraints are given and explicitly defined in the robot's state space. As a result, safety filters in robotics are often restricted to geometrically defined constraints (e.g., environment-collision constraints).

For robots to operate safely in human-centric environments, they must not only adhere to such geometrically defined constraints but also to constraints that reflect "common sense" (see Figure 1). In this work, we refer to such constraints as *semantic constraints*. For an example of such semantic constraints, consider a manipulator carrying a filled cup of water over a table. To ensure the robot operates safely, it must avoid going over electronic devices due to the risk of spillage. Hence, the semantic constraint should keep the end effector away from the overhead of entities whose semantic labels identify them as electronic devices. Additionally, the robot should avoid rotating the cup too much to prevent pouring its content and move slowly close to objects sensitive to water. Such semantic constraints are not necessarily "visible," but are critical for real-world applications. Constructing such semantic constraints requires an accurate representation of the 3D environment and a comprehensive understanding of unsafe environment interactions.

The development of large language models (LLMs) [5] and vision-language models (VLMs) [6] has led to significant advances in reasoning about 3D environments [7], [8]. Many

This work was supported by the Robotics Institute Germany, funded by BMBF grant 16ME0997K, and by the European Union's Horizon Europe research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 101155035.

The authors are with the Learning Systems and Robotics Lab and the Munich Institute of Robotics and Machine Intelligence, Technical University of Munich, 80333 Munich, Germany. Email: `firstname.lastname@tum.de`

Lukas Brunke and Angela P. Schoellig are also with the University of Toronto Institute for Aerospace Studies, North York, ON M3H 5T6, Canada, with the University of Toronto Robotics Institute, Toronto, ON M5S 1A4, Canada, and with the Vector Institute for Artificial Intelligence, Toronto, ON M5G 0C6, Canada.

recent works leverage these capabilities for language-based decision-making (e.g., to modify robot behavior [9] or to infer affordability [10]). However, systematically mapping semantic understanding to constraints remains underexplored.

In this work, we focus on robot manipulation and present a semantic safety filter that enables robots to reason about and adhere to semantically defined constraints by tightly coupling safe control, 3D perception, and LLMs (see Figure 1). Our contributions are as follows:

1) We formulate a semantic CBF safety filter framework that exploits the metric-semantic information from a 3D environment map and reasoning capabilities of LLMs for safe robot manipulation.
2) Based on environment perception and reasoning, we define three types of semantic constraints: *(i)* spatial relationship constraints (e.g., "do not move the candle below the balloon"), *(ii)* behavioral constraints (e.g., "be slower or more cautious when holding a knife"), and *(iii)* pose-based constraints (e.g., "a cup of water may not be tilted to avoid spillage").
3) We demonstrate our framework through hardware experiments using teleoperated and learned manipulation tasks. Our results verify the efficacy of our framework in satisfying semantic constraints and highlight the potential of integrating a high-level semantic understanding into safe decision-making.

II. RELATED WORK

A. Safe Robot Manipulation

In robot control, safety is often defined as ensuring the system does not violate state constraints, which can be achieved by guaranteeing set invariance [1]. Traditional approaches achieve safety or collision avoidance through collision-free trajectory generation and high-accuracy tracking control [11]. More recently, model predictive control (MPC), learning-based MPC, and geometric control methods have also been applied to enable collision-free manipulation [12], [1], [13]. Over the past two decades, safety filters, including CBFs [4], [14], Hamilton-Jacobi-reachability analysis [15] and predictive control techniques [3], have evolved, providing a modular approach to address safe control problems [1]. Safety filters can be combined with any controllers and certify potentially unsafe control inputs in a minimally invasive manner [2]. Existing approaches in safe robot control are often used for geometrically defined constraints [14], [13] and often assume the constraints are given ahead of time. How to translate semantically defined constraints to compatible analytical forms has rarely been addressed in the safe control literature.

B. Semantic 3D Representation and Spatial Reasoning

Facilitated by advances in machine learning techniques, semantic representations of robots' operating environments can be efficiently distilled from perception inputs (e.g., through object detection and segmentation) [16]. This semantic information has been integrated into 3D mapping and simultaneous localization and mapping (SLAM) algorithms [17] to create consistent instance-level or object-level maps [18], [19]. To further facilitate their usage in downstream tasks, sparse representations such as 3D scene graphs have been proposed as an abstraction of dense metric-semantic maps to capture essential relationships among entities in the environment [20]. Recent developments in LLMs and VLMs have further enabled open-vocabulary object detection, which has been applied to instance segmentation [8] and scene graph generation [7], extending 3D environment representations beyond closed sets of predefined objects. The spatial reasoning capabilities of VLMs [21], [22] have been integrated into 3D mapping frameworks, for example, to identify affordances [10] or relational keypose constraints [23] for manipulation or to identify safety-critical spatial relationships for navigation [24]. However, the semantic information in state-of-the-art 3D environment representations has not been fully exploited in downstream safe control tasks.

C. Language-Conditioned Robot Decision-Making

Recently, due to the emergence of foundation models such as CLIP [25] and the GPT series [5], there has been a significant advancement in the field of language-conditioned decision-making, including language-aided object grounding [7], [26], manipulation [27], [28], [29] and navigation [30], [31]. The abilities of LLMs and VLMs to understand and output textual information in natural language are used to perform various functions, including code writing [30], [32],[29], task planning [33], [34],[35], verifying robot behavior [36], and preference learning[35]. Hereby, the open-vocabulary capabilities of foundation models are utilized to enable flexible and adaptive reasoning in unstructured, real-world scenarios. Building on these foundations, we leverage LLMs to identify semantically unsafe conditions without being restricted to specific object classes.

III. PROBLEM STATEMENT

In this work, we consider a manipulation setup where objects are arbitrarily placed in the environment, and a robot manipulator is tasked to transport an object in the task space using teleoperation commands or a learned motion policy (see Figure 2). Generally, the teleoperation input or motion policy can be unsafe. Our goal is to design a language-aided safety filter that guarantees safe operation with respect to both semantically defined constraints $\mathcal{C}_{\text{sem}}$ (i.e., spatial relationship-based, behavior-based, and pose-based constraints) and geometrically defined constraints (i.e., environment-collision constraints $\mathcal{C}_{\text{env}}$ and self-collision constraints $\mathcal{C}_{\text{self}}$). We assume that the system can perceive and reason about its environment through a set of RGB-D images $\{\mathbf{I}_{\text{cam},f}\}$ of the scene and the associated camera poses $\{\mathbf{T}_{\text{cam},f}\}$, where f denotes the frame index.

We note that the term semantic constraint has scenario-dependent definitions in the literature (e.g., grasp types and trajectory constraints for robotic hands [37]). We refer to semantic constraints as the task-space constraints on a

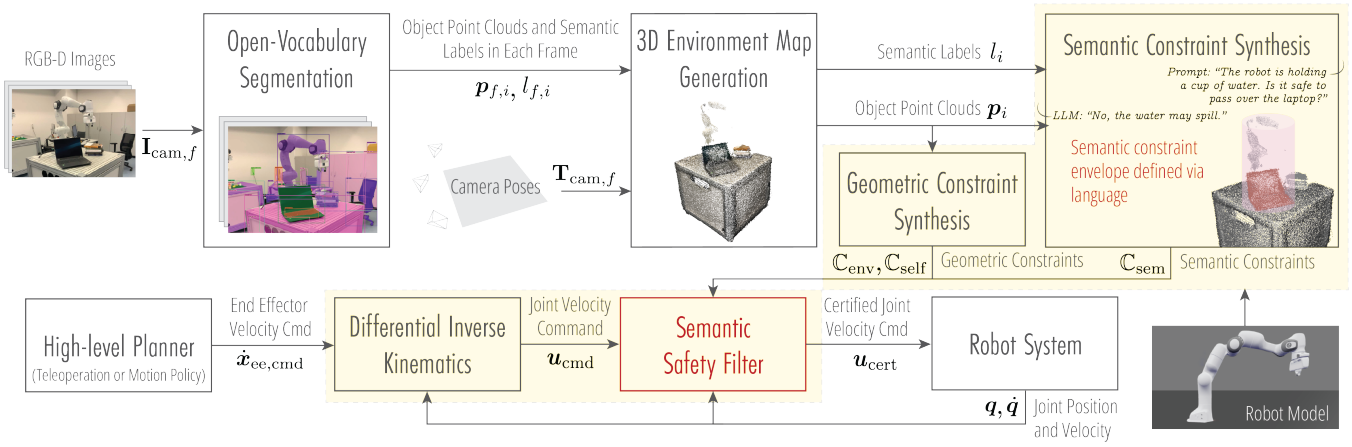

Fig. 2: An overview of our proposed semantic safety filter framework. The perception module segments the visual input and builds a semantic world representation. The LLM is queried based on the list of semantic labels and the manipulated object. It outputs the semantic context $\mathcal{S}$, which contains a list of unsafe spatial relationship-based semantic constraints for each object in the scene, a list of behavioral-based semantic constraints, and a pose-based semantic constraint. The semantic context, together with the point clouds of the objects in the scene, are then used to define safe sets for our proposed semantic safety filter. Additionally, based on the semantic context, the safety filter's parameters are adapted, for example, to prevent end effector rotations or to approach certain objects more carefully. At each time step, a high-level uncertified command from a human operator or a motion policy is mapped to the joint velocity $\boldsymbol{u}_{\text{cmd}}$ through differential inverse kinematics, certified by the proposed semantic safety filter, and then sent to the robot system.

robotic manipulator's end effector that are related to high-level semantic concepts (e.g., *"not moving a filled cup of water over electronic devices"* and *"not rotating a cup of water to avoid spilling its content"*). In contrast to typical collision avoidance constraints, semantically unsafe states are not necessarily "visible" (i.e., occupied by objects), and synthesizing the semantic constraints requires a high-level understanding of the environment and the manipulated object. In this work, we leverage the perception inputs, a model of the robot system, and an LLM to design a safety filter that guarantees semantic safety while also avoiding self-collisions and collisions with the environment.

IV. METHODOLOGY

In this section, we present the components of our proposed semantic safety filter framework, which is visualized in Figure 2. Given a set of RGB-D images and the associated camera poses, we first generate a semantic map of the 3D environment (Section IV-A). Then, a set of semantic constraints is synthesized using the semantic map and the LLM (Section IV-B). Finally, a semantic safety filter is formulated to account for the semantic constraints (Section IV-D).

A. 3D Environment Map Generation

The semantic constraints synthesis depends on a 3D environment representation that supports semantic reasoning for downstream planning and control tasks. This motivates a language-embedded representation approach. In this work, we construct an open-vocabulary object-level representation of the 3D environment [7], [8] to aid our safety filter design.

The input to the 3D environment map generation module is a set of RGB-D frames $\{\mathbf{I}_{\text{cam},f}\}$ along with the camera poses $\{\mathbf{T}_{\text{cam},f}\}$. The RGB-D images are segmented [16], and every resulting segmentation mask is embedded through the CLIP visual encoder [25] to generate segmented point clouds $\boldsymbol{p}_{f,i}$ and their associated visual embeddings $\boldsymbol{f}_{f,i}$ for each object i in each frame f. The segmented object-level point clouds $\boldsymbol{p}_{f,i}$ together with the associated camera poses $\mathbf{T}_{\text{cam},f}$ and feature vectors $\boldsymbol{f}_{f,i}$ are then used to associate objects across multiple views based on geometric and semantic similarities [7]. The per-frame information is incrementally fused to create a consistent object-level point-cloud representation of the 3D environment. The output of the map is a set of point clouds $\boldsymbol{p}_i$ and embeddings $\boldsymbol{f}_i$ for each object in the scene. Similar to [8], [7], we assign labels l_i to objects by comparing the cosine similarity between the object embeddings $\boldsymbol{f}_i$ and the text embeddings derived from the list of object categories in the ScanNet200 dataset [38]. The object's class is assigned based on the pair of embeddings with the highest cosine similarity score.

B. Semantic Constraint Synthesis

We distinguish among three types of semantic safety: *(i)* unsafe spatial relationships between the object manipulated by the robot and the objects in the scene (e.g., "do not move the candle below the balloon"), *(ii)* behavioral constraints, such as constraints on the end effector velocity based on the manipulated object and the scene objects (e.g., "be slower or more cautious when holding a knife"), and *(iii)* pose constraints on the end effector dependent on the manipulated object (e.g., "keep the cup of water upright to avoid spillage"). Such semantic constraints are object- and scene-dependent and tedious to specify manually. Therefore, we employ an LLM to synthesize them in an automated manner.

We design a language prompt for the LLM, which consists of multiple in-context examples and a final request as the true query. For each object in the scene, the requests contain the following components: *(i)* a high-level description of the scene specified by the user directly (or, inferred from a small set of images via VLM), *(ii)* the object the robot is

manipulating, and *(iii)* the object itself. Using these requests, we determine three sets of semantic constraints. First, the set of unsafe spatial relationships is $\mathcal{S}_{\text{r}}(o) = \{(l_i, r_i)\}_{i=1}^{N_{\text{r}}}$, where o is the manipulated object (e.g., `cup of water`), l_i is an object in the scene (e.g., `laptop`, `book`, etc.), r_i is an unsafe spatial relationship (e.g., `above`, `under`, or `around`), and N_{r} is the number of unsafe spatial relationships. Second, the set of unsafe behaviors is $\mathcal{S}_{\text{b}}(o) = \{(l_i, b_i)\}_{i=1}^{N_{\text{b}}}$, where b_i indicates `caution` or `no caution` and N_{b} is the number of unsafe behaviors. Finally, the pose-based constraint set is $\mathcal{S}_{\text{T}}(o) = \{T\}$, where T specifies the end effector orientation constraint (`constrained rotation` or `free rotation`). The set of semantic constraints is the union of all the semantic constraints listed above: $\mathcal{S}(o) = \mathcal{S}_{\text{r}}(o) \cup \mathcal{S}_{\text{b}}(o) \cup \mathcal{S}_{\text{T}}(o)$. For the $o =$ `cup of water` transportation example in the scene with only $l_0 =$ `laptop`, we have $\mathcal{S}_{\text{r}}(o) = \{(\texttt{laptop}, \texttt{above})\}$, $\mathcal{S}_{\text{b}}(o) = \{(\texttt{laptop}, \texttt{caution})\}$, and $\mathcal{S}_{\text{T}}(o) = \{\texttt{constrained rotation}\}$.

Our proposed semantic safety filter is designed based on control barrier certification [4]. In the following, we describe how we design the CBF safety filter using $\mathcal{S}(o)$. We denote the joint positions by $\boldsymbol{q} \in \mathbb{R}^n$ (with $n = 7$ in our case) and, similar to [14], assume direct control over the joint velocity $\dot{\boldsymbol{q}}$, (i.e., $\dot{\boldsymbol{q}} = \boldsymbol{u}$), which can be achieved via standard lower-level motion control techniques [39]. The robot's end effector position and velocity can be related to its joint position and velocity as $\boldsymbol{x}_{\text{ee}} = \boldsymbol{f}_{\text{FK}}(\boldsymbol{q})$ and $\dot{\boldsymbol{x}}_{\text{ee}} = \boldsymbol{J}(\boldsymbol{q})\,\dot{\boldsymbol{q}}$, where $\boldsymbol{f}_{\text{FK}} : \mathbb{R}^n \mapsto \mathbb{R}^3$ and $\boldsymbol{J}(\boldsymbol{q}) \in \mathbb{R}^{3\times n}$ are the translational component of the forward kinematics and the associated Jacobian matrix, respectively.

1) Spatial Relationship Constraints: The semantic constraint sets are parameterized as the zero superlevel sets of continuously differentiable functions $\boldsymbol{h}_{\text{sem}}$. Intuitively, the CBF certification framework ensures the positive invariance of the semantically safe set. This means that if the robot does not violate the semantic constraint initially, it will not violate it for all future times. For each pair (l_i, r_i) in $\mathcal{S}_{\text{r}}(o)$, based on the point cloud $\boldsymbol{p}_i$ of the object l_i and the undesirable spatial relationship r_i, we define a differentiable function $g_i : \mathbb{R}^3 \mapsto \mathbb{R}$ to capture the set of points which the robot end effector should not move into to preserve semantic safety. The semantically safe set can be expressed as

$$\mathbb{C}_{\text{sem}} = \left\{\boldsymbol{x}_{\text{ee}} \in \mathbb{R}^3 \,|\, g_i(\boldsymbol{x}_{\text{ee}}; \boldsymbol{\theta}_i) \geq 1,\ i = 1, \ldots, N_{\text{r}}\right\},$$

where $\boldsymbol{x}_{\text{ee}} = [x, y, z]^{\mathsf{T}} \in \mathbb{R}^3$ denotes the end effector position and $\boldsymbol{\theta}_i$ are parameters dependent on the object point cloud $\boldsymbol{p}_i$ and the relationship r_i.

For the $\{\texttt{laptop}, \texttt{above}\}$ example (as also illustrated in Figure 3), we define the semantically unsafe sets as a differentiable approximation using a superquadric [40]:

$$g_i(\boldsymbol{x}_{\text{ee}}; \boldsymbol{\theta}_i) = \left(\left(\frac{\tau_1(\boldsymbol{x}_{\text{ee}})}{a_{x,i}}\right)^{\frac{2}{\epsilon_{2,i}}} + \left(\frac{\tau_2(\boldsymbol{x}_{\text{ee}})}{a_{y,i}}\right)^{\frac{2}{\epsilon_{2,i}}}\right)^{\frac{\epsilon_{2,i}}{\epsilon_{1,i}}} + \left(\frac{\tau_3(\boldsymbol{x}_{\text{ee}})}{a_{z,i}}\right)^{\frac{2}{\epsilon_{1,i}}},$$

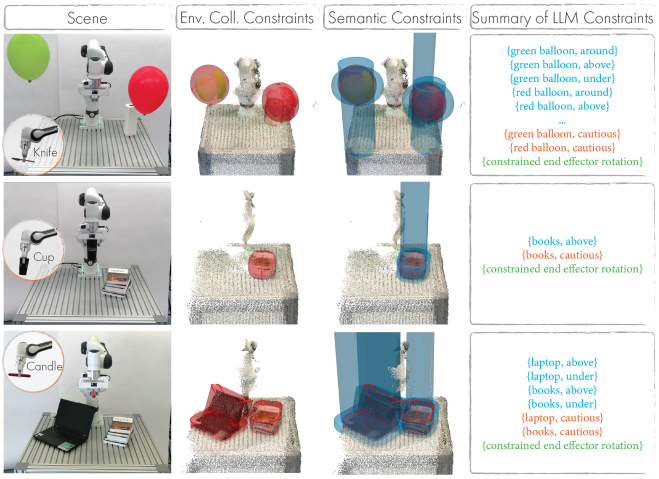

Fig. 3: Examples of the environment collision and semantic constraints enforced by our proposed semantic safety filter. For each scene, environment collision constraints are generated based on the point clouds of individual objects while the semantic constraints are synthesized based on the point clouds and labels of individual objects as well as the semantic safety conditions from the LLM. The semantic safety conditions are further categorized into spatial relationship constraints (blue text), behavioral constraints (orange text), and end effector pose constraints (green text).

where $\epsilon_{1,i}$ and $\epsilon_{2,i}$ define the shape of the superquadric, $a_{x,i}, a_{y,i}$, and $a_{z,i}$ are scaling parameters, and τ_1, τ_2, and τ_3 transform the end effector coordinates into the superquadric's coordinate frame. To improve nonconvex objects' representations, we create unions of superquadrics to accurately fit spatial constraints. For example, we fit separate superquadrics for the part of the `laptop`'s point cloud that resembles the keyboard and the screen. This segmentation by parts can be achieved using plane detection algorithms or learned segmentation models [41]. To account for the spatial relationship `above`, we extend the point cloud in its positive z-direction. For this, we duplicate the point cloud, set the duplicate's z-coordinates to be outside the robot's workspace, and fit the superquadric based on the union of the original and the expanded point cloud. We consider 12 spatial relationships in total, such as `under` and `around`, for which we define similar superquadrics.

To achieve spatial semantic safety with respect to the semantic constraint set $\mathbb{C}_{\text{sem}}$, we define a vector of CBFs $\boldsymbol{h}_{\text{sem}}(\boldsymbol{x}_{\text{ee}})$, where the i-th element is

$$h_{\text{sem},i}(\boldsymbol{x}_{\text{ee}}) = g_i(\boldsymbol{x}_{\text{ee}}; \boldsymbol{\theta}_i) - 1. \tag{1}$$

Using forward kinematics, we can express the semantic constraint set based on the CBFs (1) in the robot's configuration space, which yields our desired safe set

$$\mathbb{C}_{\text{sem}} = \{\boldsymbol{q} \in \mathbb{R}^n \,|\, \boldsymbol{h}_{\text{sem}}(\boldsymbol{f}_{\text{FK}}(\boldsymbol{q})) \geq \boldsymbol{0}\}. \tag{2}$$

2) Behavioral Constraints: The behavioral constraints are implemented using constraints on the time derivative of the CBF, i.e., the control invariance condition [4], of the form

$$\dot{\boldsymbol{h}}_{\text{sem}}(\boldsymbol{q}, \boldsymbol{u}) = \mathbf{H}_{\text{sem}}(\boldsymbol{q})\boldsymbol{J}(\boldsymbol{q})\,\boldsymbol{u} \geq -\boldsymbol{\alpha}_{\text{sem}}(\boldsymbol{h}_{\text{sem}}(\boldsymbol{q}); \mathcal{S}_{\text{b}}(o)),$$

where $\mathbf{H}_{\text{sem}}(\boldsymbol{q}) = \left.\frac{\partial \boldsymbol{h}_{\text{sem}}}{\partial \boldsymbol{x}_{\text{ee}}}\right|_{\boldsymbol{x}_{\text{ee}} = \boldsymbol{f}_{\text{FK}}(\boldsymbol{q})}$ and $\boldsymbol{\alpha}_{\text{sem}}$ is a vector of class $\mathcal{K}_\infty$ functions (i.e., real-valued functions that pass

through the origin and are strictly increasing). Intuitively, the condition bounds how fast the robot system is allowed to approach the semantic safety boundary through the design of $\boldsymbol{\alpha}_{\text{sem}}$ and ensures that the constraints defined by $\boldsymbol{h}_{\text{sem}}$ are always satisfied (i.e., the set $\mathcal{C}_{\text{sem}}$ is forward invariant) [4]. In particular, we design the class $\mathcal{K}_\infty$ to adhere to behavioral semantic constraints b_j from $\mathcal{S}_{\text{b}}(o)$ such that the system approaches the safe set boundary of the object with label l_j more slowly and exhibits the desired level of caution. For example, for the case $b_j = \texttt{caution}$, we reduce the steepness of $\alpha_{\text{sem},j}$. In that case, we also write $\alpha_{\text{sem},j}(\cdot;\texttt{caution}) = \alpha_{\text{sem,c},j}(\cdot)$. This reduction can be achieved by using a class $\mathcal{K}_\infty$ that is strictly smaller than $\alpha_{\text{sem},j}$ for positive $h_{\text{sem},j}$. Such a function can be produced by multiplying the function $\alpha_{\text{sem},j}$ with a scalar $w_{\alpha,j} \in (0,1)$.

3) Pose Constraints: The pose constraint is active if $\mathcal{S}_{\text{T}}(o) = \{\texttt{constrained rotation}\}$. In that case, we add the following constraint:

$$\boldsymbol{\Delta\psi}_{\min} \leq \log(\boldsymbol{R}_{\text{des}}\boldsymbol{R}_{\text{cur}}^{\mathsf{T}})^{\vee} - \boldsymbol{\psi} \leq \boldsymbol{\Delta\psi}_{\max}\,,$$

where $\boldsymbol{R}_{\text{des}}$ is the desired rotation of the end effector (the end effector's initial orientation during the object's pick-up), $\boldsymbol{R}_{\text{cur}}$ is the current rotation of the end effector, $\boldsymbol{\psi} = \boldsymbol{J}_o(\boldsymbol{q})\boldsymbol{u}\Delta t$ is the predicted rotation of the end effector at the next timestep $(t+\Delta t)$ with $\boldsymbol{J}_o(\boldsymbol{q})$ being the Jacobian relating the joint velocity to the angular velocity of the end effector, $(\cdot)^{\vee}$ denotes the inverse of the skew-symmetric operator $(\cdot)^{\wedge}$ [42], and $\boldsymbol{\Delta\psi}_{\min}$ and $\boldsymbol{\Delta\psi}_{\max}$ are the tolerated orientation errors. In our implementation, we leverage a softened formulation for this constraint to make the approach less prone to infeasibility. We express the softened pose constraint using the objective $\boldsymbol{w}_{\text{rot}}(\mathcal{S}_{\text{T}}(o))^{\mathsf{T}}\boldsymbol{L}_{\text{rot}}(\boldsymbol{q},\boldsymbol{u})$. The weight $\boldsymbol{w}_{\text{rot}} \in \mathbb{R}^2$ is determined based on the semantic context T in $\mathcal{S}_{\text{T}}$. The end effector is free to rotate if $T = \texttt{free rotation}$ (e.g., no object is being held) with $\boldsymbol{w}_{\text{rot}} = 0$, but $\boldsymbol{w}_{\text{rot}} > 0$ if $T = \texttt{constrained rotation}$ (e.g., a cup of water is being manipulated to prevent spilling). The vector $\boldsymbol{L}_{\text{rot}}$ is

$$\boldsymbol{L}_{\text{rot}}(\boldsymbol{q},\boldsymbol{u}) = \begin{bmatrix} \| \log(\boldsymbol{R}_{\text{des}}\boldsymbol{R}_{\text{cur}}^{\mathsf{T}})^{\vee} - \boldsymbol{\psi}\|_2^2 & \|\boldsymbol{\psi}\|_2^2 \end{bmatrix}^{\mathsf{T}}\,,$$

where the first element represents the cost for the difference between the predicted orientation at the next timestep and the desired orientation of the manipulator's end effector and the purpose of the second element is to prevent the end effector from rotating too fast and to keep perturbations small.

C. Geometric Constraints

In addition to semantic constraints, we require the robot to adhere to geometric constraints, which include environment-collision and self-collision constraints. We incorporate these additional constraints into two more vectors of CBFs $\boldsymbol{h}_{\text{env}}(\boldsymbol{q})$ and $\boldsymbol{h}_{\text{self}}(\boldsymbol{q})$. The environment-collision constraints are defined based on CBFs using superquadrics fitted to the point clouds $\boldsymbol{p}_i$ (see previous section); the self-collision constraints are formulated by placing multiple spherical CBFs along the body of the robot, similarly as in [14].

D. Semantic Safety Filter Formulation

Given the semantic constraints $\mathcal{C}_{\text{sem}}$ and the set $\mathcal{S}$, our goal is to modify potentially unsafe commands sent by a human operator or coming from a motion policy. As depicted in Figure 2, in our setup, we send the desired end effector velocity commands $\dot{\boldsymbol{x}}_{\text{ee,cmd}}$, which are converted to desired joint velocity commands $\boldsymbol{u}_{\text{cmd}}$ using differential inverse kinematics. The semantic safety filter then computes a certified input $\boldsymbol{u}_{\text{cert}}$ that best matches the desired joint velocity $\boldsymbol{u}_{\text{cmd}}$ while ensuring semantic and geometric constraint satisfaction. The semantic safety filter is formulated as

$$\begin{aligned} \boldsymbol{u}_{\text{cert}} = \underset{\boldsymbol{u}\in\mathbb{U}}{\operatorname{argmin}} \quad & \|\boldsymbol{u}-\boldsymbol{u}_{\text{cmd}}\|_2^2 + \boldsymbol{w}_{\text{rot}}(\mathcal{S}_{\text{T}}(o))^{\mathsf{T}}\boldsymbol{L}_{\text{rot}}(\boldsymbol{q},\boldsymbol{u}) \\ \text{s.t.} \quad & \dot{\boldsymbol{h}}_{\text{sem}}(\boldsymbol{q},\boldsymbol{u};\mathcal{S}_{\text{r}}(o)) \geq -\boldsymbol{\alpha}_{\text{sem}}(\boldsymbol{h}_{\text{sem}}(\boldsymbol{q});\mathcal{S}_{\text{b}}(o)) \\ & \dot{\boldsymbol{h}}_{\text{env}}(\boldsymbol{q},\boldsymbol{u}) \geq -\boldsymbol{\alpha}_{\text{env}}(\boldsymbol{h}_{\text{env}}(\boldsymbol{q});\mathcal{S}_{\text{b}}(o)) \\ & \dot{\boldsymbol{h}}_{\text{self}}(\boldsymbol{q},\boldsymbol{u}) \geq -\boldsymbol{\alpha}_{\text{self}}(\boldsymbol{h}_{\text{self}}(\boldsymbol{q})) \\ & \dot{\boldsymbol{h}}_{\text{lim}}(\boldsymbol{q},\boldsymbol{u}) \geq -\boldsymbol{\alpha}_{\text{lim}}(\boldsymbol{h}_{\text{lim}}(\boldsymbol{q}))\,, \end{aligned} \tag{3}$$

where we made the dependency on the semantic context $\mathcal{S}(o)$ explicit, added joint angle and velocity constraints through additional CBFs $\boldsymbol{h}_{\text{lim}}(\boldsymbol{q})$, and $\boldsymbol{\alpha}_{\text{env}}$, $\boldsymbol{\alpha}_{\text{self}}$, $\boldsymbol{\alpha}_{\text{lim}} \in \mathcal{K}_\infty$. The first term in the cost function minimizes the difference between the certified input and the desired input command, while the second term penalizes rotations away from the desired rotation. The four sets of inequality constraints in (3) correspond to the semantic spatial relationship-based, environment-collision, self-collision, and joint angle and velocity constraints. The class $\mathcal{K}_\infty$ functions define behavioral semantics for each constraint, and the objective provides softened posed-based safety constraints. The semantic safety filter optimization problem (3) is a QP that can be efficiently solved online. Overall, the semantic safety filter in (3) finds the control input that best matches the desired input while ensuring all constraints are satisfied.

V. EXPERIMENTS

In this section, we present the experimental evaluation of our proposed semantic safety filter. In the real-world experiment, a Franka Emika FR3 robotic manipulator is deployed with our proposed semantic safety filter in a closed loop to prevent potentially unsafe commands from a non-expert user or a learned motion policy. A video of the experimental results can be found at https://tiny.cc/semantic-manipulation and on our website https://utiasdsl.github.io/semantic-manipulation/.

A. Semantic Perception

In our evaluation, we consider static (unless manipulated by the robot) scenes, some of which are visualized in Figure 3 and various manipulated objects, including a `dry sponge`, a `cup of water`, a `lit candle`, and a `knife`. The geometries of the manipulated objects and the robot are assumed to be known. However, the environment in which the robot operates is assumed to be unknown; a map for each environment is generated using RGB-D images

TABLE I: The multi-prompt strategy yields higher precision and recall than single-prompting on our benchmarking dataset of ground-truth constraints.

Prompting Strategy	Precision (%)	Recall (%)
Single-Prompt	29	78
Multi-Prompt	60	99

and associated camera frames as described in Section IV-A. The RGB-D images were recorded using a Femto Bolt and the camera poses were obtained by running visual-inertial SLAM [43]. Each scene was reconstructed using approximately 50 to 200 RGB-D images and their associated camera poses, and the semantics are determined as described in Section IV-A. Examples of the reconstructed scenes are shown in Figure 3.

B. LLM Prompting

We created a benchmarking dataset of objects, scenes, and ground-truth constraints to evaluate the semantic constraint generation. The dataset includes over 50 semantic constraints containing all semantic constraint types, as well as objects and scenes not encountered in our experiments, and we use it to evaluate two different prompting strategies on an LLM (GPT-4o [44]). The first strategy (single-prompt) requests the full set $\mathcal{S}(o)$ at once, while the second strategy (multi-prompt) requests only one pair or a singleton (for the semantic pose constraint) for each prompt. The multi-prompt method proved more accurate, as indicated by the higher precision and recall in Table I. We adjusted the final prompt until the desired level of accuracy was achieved on the validation dataset.

For our robot experiments, we follow the methodologies in Section IV-B to identify semantically unsafe object-relationship pairs, behaviors, and poses. Examples are shown in the last column of Figure 3. We query the LLM for each object-relationship pair for each scene multiple times using majority voting to determine if the spatial relationship between the manipulated object and the particular object in the scene is semantically safe. We run additional queries to determine if the object held by the manipulator may be rotated and if increased caution should be exhibited close to each of the objects in the scene. These responses are then used in combination with each object's point cloud to determine the constraint envelopes (see Figure 3), the class $\mathcal{K}_\infty$ function, and the weight $\boldsymbol{w}_{\text{rot}}$.

C. Demonstration in Tabletop Manipulation Tasks

Using our semantic safety filter, we execute various teleoperation and pick-and-place tasks on the robot. We run our semantic safety filter at 45 Hz. Our teleoperation experiments are summarized in Table II. The teleoperation commands are provided through a teleoperation interface as end effector velocities in the Cartesian space and smoothed using a low-pass filter. We calculate the associated joint velocities with differential inverse kinematics. Each scene is tested with multiple held objects, which require different sets of semantic constraints (see Figure 3). The results in the table confirm

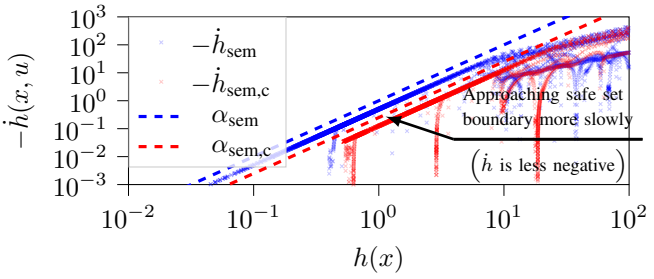

Fig. 4: The level of caution determines how quickly the end effector approaches a safety constraint boundary. In the `books` scene, we increase caution by adjusting the class $\mathcal{K}_\infty$ function when holding a `cup of water` under the same semantic constraint during teleoperation. In the cautious case, the negative time derivatives remain below the red dashed line, satisfying the CBF condition. Since $\alpha_{\text{sem,c}} < \alpha_{\text{sem}}$, the end effector approaches the boundary more slowly. Note that the y-axis is inverted.

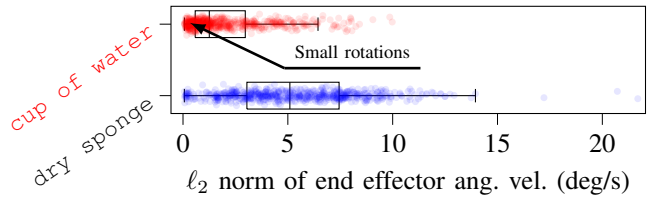

Fig. 5: Demonstration of the active (inactive) rotation constraint when the robot is holding a `cup of water` (`dry sponge`) in the scene `books`. The distribution for the `cup of water` is skewed towards smaller angular velocities; an active rotation constraint (red) generally yields reduced end effector rotations as compared to the inactive case (blue).

that our safety filters can effectively account for collision avoidance constraints and any semantic constraints generated by our synthesis module, as no constraint violations occur in any of our experiments when the safety filter is active.

We highlight how the different levels of caution determine how quickly the end effector holding a specific object may approach the boundary of a safety constraint boundary. For the scene `books`, we show increased caution by modifying the class $\mathcal{K}_\infty$ function when holding a `cup of water` for the same semantic constraint during teleoperation. For the cautious case, the negative time derivatives $(-\dot{h})$ (red) stay below the red dashed line, confirming the CBF condition's satisfaction. As $\alpha_{\text{sem,c}}(h) = \frac{1}{4}h^2$ is strictly smaller than $\alpha_{\text{sem}}(h) = h^2$ on $h > 0$, the end effector approaches the boundary of this semantic constraint slower. Note that we manually overwrote the level of caution for this particular demonstration to compare the closed-loop behavior on the same semantic CBF constraint. Generally, the level of caution is determined through the method outlined in Section IV-B.

Finally, we demonstrate the effectiveness of constraining rotations for different objects based on their semantics in Figure 5. Our semantic safety filter successfully reduces the median of the norm of the end effector's angular velocity by 75.39% if the rotation constraint is active (see `cup of water`). The box plot also highlights that the interquartile range of the end effector's angular velocity norm is reduced by 45.67% compared to the robot holding the `dry sponge`.

TABLE II: A summary table of the mean percentages and their associated standard deviations of time steps that violate any of the constraints $\mathcal{C}_{\text{sem}}$, $\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{self}}, \mathcal{C}_{\text{lim}}$. Our evaluation includes a baseline without a safety filter, a safety filter accounting for geometric constraints, and our proposed semantic safety filter. We use three scenes and five different manipulation cases (four objects and empty-handed) with five teleoperated trajectories each, resulting in a total of 40 trajectories for each method. Each combination of objects and scenes yielded different geometric and semantic constraints.

Scene	Held Object†	No Safety Filter	Nominal Safety Filter (w/o $\mathcal{C}_{\text{sem}}$)	**Our Semantic Safety Filter**
{books}	dry sponge	11.06% ± 13.60%	**0.00% ± 0.00%**	**0.00% ± 0.00%**
	cup of water	70.37% ± 23.51%	64.98% ± 33.42%	**0.00% ± 0.00%**
{laptop, books}	none	36.29% ± 18.29%	**0.00% ± 0.00%**	**0.00% ± 0.00%**
	lit candle	65.21% ± 14.20%	51.33% ± 27.85%	**0.00% ± 0.00%**
	cup of water	59.40% ± 12.02%	41.90% ± 25.46%	**0.00% ± 0.00%**
{balloons, paper towel}	cup of water	28.07% ± 14.77%	**0.00% ± 0.00%**	**0.00% ± 0.00%**
	lit candle	50.33% ± 9.44%	49.89% ± 9.04%	**0.00% ± 0.00%**
	knife	49.07% ± 16.16%	30.85% ± 10.53%	**0.00% ± 0.00%**

†The objects in red result in semantic constraints.

TABLE III: User study for data collection in the bottle transport task.

Data Collection Method	**CPH†**	**Constraint Violations**
Teleoperation with Safety Filter	132	0% of time
Teleoperation without Safety Filter	120	5% of time

†The abbreviation "CPH" denotes the number of task completions per hour, with higher values corresponding to higher efficiency.

We note that, in our implementation, the semantic context $\mathcal{S}_T(o)$ is binary (i.e., either constrained or unconstrained rotation). However, it is generally possible to prompt the LLM with finer granularity and enforce varying levels of cautiousness by appropriately configuring the vector $\boldsymbol{\omega}_{\text{rot}}$.

To further evaluate the scalability of our proposed approach to more complex environments, we applied our semantic safety filter to pick-and-place tasks in a cluttered environment with 17 objects (see our supplementary video). When the robot is holding the dry sponge, its end effector is allowed to rotate and move the object above electronic devices with no additional caution considered; in contrast, the robot's motion is much more constrained when holding the cup of water to prevent potential spillage.

D. Demonstration in a Real-World Kitchen Environment

To demonstrate the applicability of our proposed filter beyond teleoperation, we conducted experiments in a real-world kitchen environment and trained diffusion policies [45] for five different transportation tasks involving various semantically unsafe constraints. These constraints include handling fragile items and preventing fire and electrical hazards. Clips of this set of experiments are included in the supplementary video. Figure 6 compares the normalized CBFs for our proposed semantic safety filter and a nominal geometric safety filter that does not account for semantic constraints. The proposed semantic safety filter successfully prevents unsafe actions such as placing a metal cup inside a microwave or putting a pressurized spray can on a stove. This set of experiments highlights the generalizability of our proposed approach to learned policies and its applicability in real-world settings.

Lastly, we note that our proposed safety filter can also enhance the data collection process for training policies. Table III summarizes the results from a user study, where

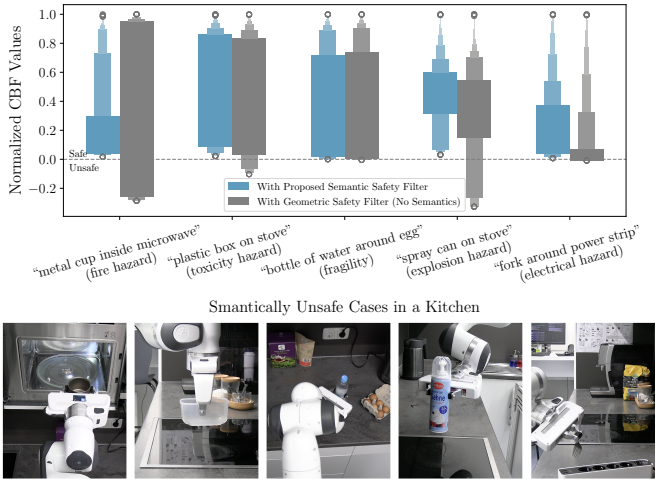

Fig. 6: A comparison of normalized semantic CBF values for applying the proposed semantic safety filter *(top, blue plots)* versus the typical geometric safety filter *(top, grey plots)* to diffusion policies across five different scenarios *(bottom)*. The distribution data includes augmented data from five trials for each scenario. The proposed semantic safety filter effectively addresses common sense constraints of different types, ranging from the considerations for fragile items to the prevention of fire and electrical hazards.

teleoperated data collection with the safety filter achieved zero constraint violations without compromising the speed of the process. This suggests the potential for generating higher-quality training data, particularly for applications where semantic or "common sense" safety is a critical requirement.

VI. CONCLUSION AND FUTURE WORK

This work proposes a semantic safety filter framework combining semantic scene understanding and contextual reasoning capabilities of LLMs with CBF-based safe control. Our framework allows satisfying constraints that are "invisible" in a 3D map but considered "common sense" while also guaranteeing collision-free motion and adherence to robot-specific constraints. We demonstrate the effectiveness of our framework in several real-world manipulation tasks. Our work highlights that integrating semantic understanding into safe decision-making is crucial to going beyond pure collision avoidance and achieving a more general notion of

safety closer to that expected by humans. To the best of our knowledge, our work is the first to integrate semantics and robot control with formal safety guarantees. In the future, we plan to extend our approach by incorporating semantic constraints defined based on spatial verbs (e.g., "blocking,") and further accounting for dynamic environments.

REFERENCES

[1] L. Brunke, M. Greeff, A. W. Hall, Z. Yuan, S. Zhou, J. Panerati, and A. P. Schoellig, "Safe learning in robotics: From learning-based control to safe reinforcement learning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 411–444, 2022.

[2] K.-C. Hsu, H. Hu, and J. F. Fisac, "The safety filter: A unified view of safety-critical control in autonomous systems," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 7, pp. 47–72, 2023.

[3] K. P. Wabersich, A. J. Taylor, J. J. Choi, K. Sreenath, C. J. Tomlin, A. D. Ames, and M. N. Zeilinger, "Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems," *IEEE Control Systems Magazine*, vol. 43, no. 5, pp. 137–177, 2023.

[4] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. of the European Control Conf. (ECC)*, 2019, pp. 3420–3431.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," in *Proc. of the Advances in Neural Info. Processing Systems (NeurIPS)*, vol. 33, 2020, pp. 1877–1901.

[6] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Proc. of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, 2024, pp. 34 892–34 916.

[7] Q. Gu, A. Kuwajerwala, S. Morin, K. M. Jatavallabhula, B. Sen, A. Agarwal, C. Rivera, W. Paul, K. Ellis, R. Chellappa, *et al.*, "ConceptGraphs: Open-vocabulary 3D scene graphs for perception and planning," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5021–5028.

[8] A. Takmaz, E. Fedele, R. W. Sumner, M. Pollefeys, F. Tombari, and F. Engelmann, "OpenMask3D: Open-Vocabulary 3D Instance Segmentation," in *Proc. of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[9] A. Bucker, L. Figueredo, S. Haddadin, A. Kapoor, S. Ma, S. Vemprala, and R. Bonatti, "LATTE: Language trajectory transformer," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 7287–7294.

[10] W. Yuan, J. Duan, V. Blukis, W. Pumacay, R. Krishna, A. Murali, A. Mousavian, and D. Fox, "RoboPoint: A vision-language model for spatial affordance prediction for robotics," in *Proc. of the Conference on Robot Learning (CoRL)*, 2024.

[11] M. W. Spong, "An historical perspective on the control of robotic manipulators," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 5, pp. 1–31, 2022.

[12] J.-R. Chiu, J.-P. Sleiman, M. Mittal, F. Farshidian, and M. Hutter, "A collision-free MPC for whole-body dynamic locomotion and manipulation," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2022, pp. 4686–4693.

[13] N. D. Ratliff, J. Issac, D. Kappler, S. Birchfield, and D. Fox, "Riemannian motion policies," *arXiv preprint arXiv:1801.02854*, 2018.

[14] A. Singletary, W. Guffey, T. G. Molnar, R. Sinnet, and A. D. Ames, "Safety-critical manipulation for collision-free food preparation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10 954–10 961, 2022.

[15] S. Bansal, M. Chen, S. Herbert, and C. J. Tomlin, "Hamilton-Jacobi reachability: A brief overview and recent advances," in *Proc. of the Annual Conf. on Decision and Control (CDC)*, 2017, pp. 2242–2253.

[16] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollar, and R. Girshick, "Segment Anything," in *Proc. of the IEEE/CVF Intl. Conference on Computer Vision (ICCV)*, 2023, pp. 4015–4026.

[17] J. Crespo, J. C. Castillo, O. M. Mozos, and R. Barber, "Semantic information for robot navigation: A survey," *Applied Sciences*, vol. 10, no. 2(497), pp. 1–28, 2020.

[18] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 1689–1696.

[19] S. Leutenegger, "OKVIS2: Realtime scalable visual-inertial slam with loop closure," *arXiv preprint arXiv:2202.09199*, 2022.

[20] J. Wald, H. Dhamo, N. Navab, and F. Tombari, "Learning 3D semantic scene graphs from 3D indoor reconstructions," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3961–3970.

[21] B. Chen, Z. Xu, S. Kirmani, B. Ichter, D. Sadigh, L. Guibas, and F. Xia, "SpatialVLM: Endowing vision-language models with spatial reasoning capabilities," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 14 455–14 465.

[22] Z. Qi, Z. Zhang, Y. Fang, J. Wang, and H. Zhao, "GPT4Scene: Understand 3D scenes from videos with vision-language models," *arXiv preprint arXiv:2501.01428*, 2025.

[23] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *Proc. of the Conference on Robot Learning (CoRL)*, 2024.

[24] L. Santos, Z. Li, L. Peters, S. Bansal, and A. Bajcsy, "Updating robot safety representations online from natural language feedback," *arXiv preprint arXiv:2409.14580*, 2024.

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. of International Conf. on Machine Learning (ICML)*, 2021, pp. 8748–8763.

[26] S. Peng, K. Genova, C. Jiang, A. Tagliasacchi, M. Pollefeys, T. Funkhouser, *et al.*, "OpenScene: 3D scene understanding with open vocabularies," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 815–824.

[27] A. Rashid, S. Sharma, C. M. Kim, J. Kerr, L. Y. Chen, A. Kanazawa, and K. Goldberg, "Language embedded radiance fields for zero-shot task-oriented grasping," in *Proc. of the Conference on Robot Learning (CoRL)*, 2023.

[28] A. Xie, Y. Lee, P. Abbeel, and S. James, "Language-conditioned path planning," in *Proc. of the Conference on Robot Learning (CoRL)*, 2023.

[29] T. Oelerich, C. Hartl-Nesic, and A. Kugi, "Language-guided manipulator motion planning with bounded task space," in *Proc. of the Conference on Robot Learning (CoRL)*, 2024.

[30] C. Huang, O. Mees, A. Zeng, and W. Burgard, "Visual language maps for robot navigation," in *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 10 608–10 615.

[31] M. S. Zhang, K. Qu, V. Patil, C. Cadena, and M. Hutter, "Tag map: A text-based map for spatial reasoning and navigation with large language models," in *Proc. of the Conference on Robot Learning (CoRL)*, 2024.

[32] 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," in *Proc. of the Conference on Robot Learning (CoRL)*, 2023.

[33] K. Rana, J. Haviland, S. Garg, J. Abou-Chakra, I. Reid, and N. Suenderhauf, "SayPlan: Grounding large language models using 3D scene graphs for scalable robot task planning," in *Proc. of the Conference on Robot Learning (CoRL)*, 2023.

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

[35] H. Wang, N. Chin, G. Gonzalez-Pumariega, X. Sun, N. Sunkara, M. A. Pace, J. Bohg, and S. Choudhury, "APRICOT: active preference learning and constraint-aware task planning with llms," in *Proc. of the Conference on Robot Learning (CoRL)*, 2024.

[36] L. Guan, Y. Zhou, D. Liu, Y. Zha, H. B. Amor, and S. Kambhampati, ""task success" is not enough: Investigating the use of video-language models as behavior critics for catching undesirable agent behaviors," *arXiv preprint arXiv:2402.04210*, 2024.

[37] C. Li and G. Tian, "Transferring the semantic constraints in human manipulation behaviors to robots," *Applied Intelligence*, vol. 50, no. 6, pp. 1711–1724, 2020.

[38] D. Rozenberszki, O. Litany, and A. Dai, "Language-grounded indoor

3D semantic segmentation in the wild," in *Proc. of the European Conference on Computer Vision (ECCV)*, 2022, pp. 125–141.

[39] K. M. Lynch and F. C. Park, *Modern Robotics*. Cambridge University Press, 2017.

[40] W. Liu, Y. Wu, S. Ruan, and G. S. Chirikjian, "Robust and accurate superquadric recovery: A probabilistic approach," in *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2676–2685.

[41] G. Sharma, B. Dash, A. RoyChowdhury, M. Gadelha, M. Loizou, L. Cao, R. Wang, E. G. Learned-Miller, S. Maji, and E. Kalogerakis, "PriFit: Learning to fit primitives improves few shot point cloud segmentation," in *Proc. of the Computer Graphics Forum*, vol. 41, no. 5, 2022, pp. 39–50.

[42] T. D. Barfoot, *State Estimation for Robotics*. Cambridge University Press, 2024.

[43] O. Seiskari, P. Rantalankila, J. Kannala, J. Ylilammi, E. Rahtu, and A. Solin, "HybVIO: Pushing the limits of real-time visual-inertial odometry," in *Proc. of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022, pp. 701–710.

[44] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, *et al.*, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[45] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. C. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proc. of Robotics: Science and Systems (RSS) Conference*, 2023.