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

Lukas Brunke[1,2,3], **Yanni Zhang**[1], **Ralf Römer**[1], **Jack Naimer**[1,2], **Nikola Staykov**[1],
Siqi Zhou[1], **Angela P. Schoellig**[1,2,3]
[1]Learning Systems and Robotics Lab, Technical University of Munich, Munich, Germany
[2]University of Toronto, Toronto, ON, Canada
[3]Vector Institute for Artificial Intelligence, Toronto, ON, Canada
firstname.lastname@tum.de

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, behaviours, 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 evaluated our semantic safety filter approach in teleoperated tabletop manipulation tasks and pick-and-place tasks, demonstrating its effectiveness in incorporating semantic constraints to ensure safe robot operation beyond collision avoidance.

1 Introduction

Safety is a key issue in robotics and has been addressed from different perspectives. In safety-critical control, the goal is usually to guarantee set invariance (i.e., to prevent a system from leaving a certain safe set) [Brunke et al., 2022]. 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 [Hsu et al., 2023, Wabersich et al., 2023]. Existing safety filters such as control barrier function (CBF) safety filters [Ames et al., 2019] or predictive safety filters [Wabersich et al., 2023] 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

Preprint. Under review.

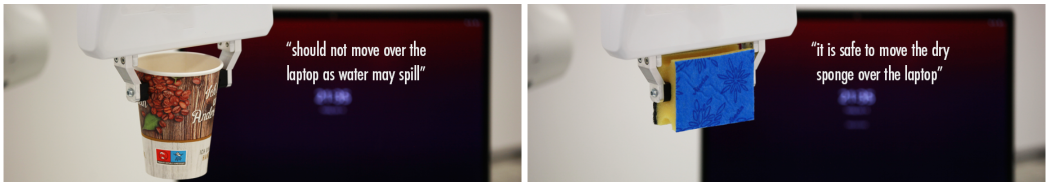
"should not move over the laptop as water may spill"
"it is safe to move the dry sponge over the laptop"

Figure 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/, our full paper https://tiny.cc/semantic-filter-paper, and in our short video https://tiny.cc/semantic-manipulation.

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) [Brown et al., 2020] and vision-language models (VLMs) [Liu et al., 2024] has led to significant advances in reasoning about 3D environments [Gu et al., 2024, Takmaz et al., 2023]. Many recent works leverage these capabilities for language-based decision-making (e.g., to modify robot behavior [Bucker et al., 2023] or to infer affordability [Ahn et al., 2022]). 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 pick-and-place 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.

2 Related Work

2.1 Safe Robot Manipulation

In robot control, safety is often defined as ensuring the system does not violate state constraints. This can be achieved by guaranteeing set invariance, which means that once the system's state is initialized in a predefined safe set, it remains within that set for all future times under the given control inputs [Brunke et al., 2022]. Traditional approaches achieve safety or collision avoidance through collision-free trajectory generation and high-accuracy tracking control [Spong, 2022]. More recently, model predictive control (MPC), learning-based MPC, and geometric control methods have also been applied to enable collision-free manipulation [Chiu et al., 2022, Brunke et al., 2022, Ratliff et al., 2018]. Over the past two decades, safety filters, including control barrier functions

(CBFs) [Ames et al., 2019, Singletary et al., 2022], Hamilton-Jacobi-reachability analysis [Bansal et al., 2017] and predictive control techniques [Wabersich et al., 2023], have evolved, providing a modular approach to address safe control problems [Brunke et al., 2022]. Safety filters can be combined with any controllers and certify potentially unsafe control inputs in a minimally invasive manner [Hsu et al., 2023]. Existing approaches in safe robot control are often used for geometrically defined constraints [Singletary et al., 2022, Ratliff et al., 2018] 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.

2.2 Semantic 3D Representation and Spatial Reasoning

There has been extensive work on efficient semantic representation of robots' operating environments. Facilitated by the advances in machine learning techniques, semantic information can be efficiently distilled from perception inputs (e.g., through object detection and segmentation) [Kirillov et al., 2023]. This semantic information has been integrated into 3D mapping and simultaneous localization and mapping (SLAM) algorithms [Crespo et al., 2020] to create consistent instance-level or object-level maps [Rosinol et al., 2020, Leutenegger, 2022]. 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 of entities in the environment [Wald et al., 2020]. Recently, developments in LLMs and VLMs have further enabled open-vocabulary object detection, which has been applied to instance segmentation [Takmaz et al., 2023] and scene graph generation [Gu et al., 2024], extending 3D environment representations beyond closed sets of predefined objects. While semantic information is becoming an integral part of the state-of-the-art 3D environment representations, the semantic environment understanding has not been fully exploited in downstream safe control tasks.

2.3 Language-Conditioned Robot Decision-Making

Recently, due to the emergence of foundation models such as CLIP [Radford et al., 2021] and the GPT series [Brown et al., 2020], there has been a significant advancement in the field of language-conditioned decision-making, including language-aided object grounding [Gu et al., 2024, Peng et al., 2023], manipulation [Rashid et al., 2023, Xie et al., 2023] and navigation [Huang et al., 2023a]. LLMs and VLMs are deployed to perform the following functions in this regard such as code writing [Huang et al., 2023a,b], task planning [Rana et al., 2023, Huang et al., 2022], and verifying robot behavior [Guan et al., 2024], where the ability to understand or output textual information in natural language is crucial for these applications. Notably, the open-vocabulary capabilities of foundation models are utilized in this field. Similarly, we leverage these capabilities in LLMs to identify semantically unsafe conditions without pre-defining object classes.

3 Problem Statement

In this work, we consider a tabletop manipulation setup where objects are placed on a flat surface, and a robot manipulator is tasked to transport an object in the task space using teleoperation commands or a high-level 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, behaviour-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., such as grasp types and trajectory constraints for robotic hands [Li and Tian, 2020]). We refer to semantic constraints as the task-space constraints on a 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

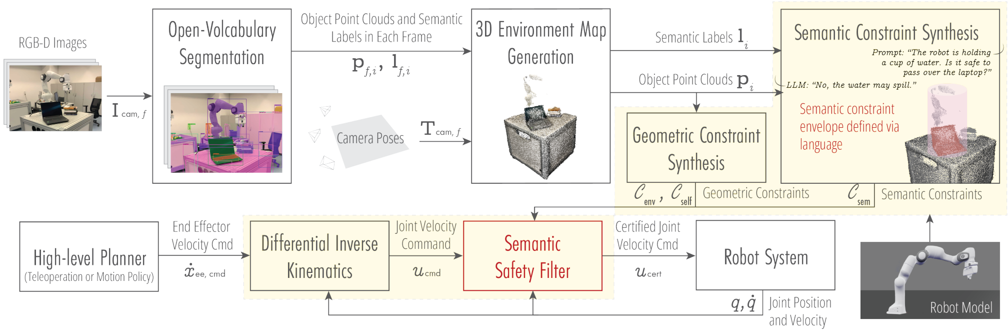
RGB-D Images
$\mathbf{I}_{\text{cam}, f}$
Open-Volcabulary Segmentation
Object Point Clouds and Semantic Labels in Each Frame
$\mathbf{p}_{f,i}, \mathbf{l}_{f,i}$
Camera Poses
$\mathbf{T}_{\text{cam}, f}$
3D Environment Map Generation
Semantic Labels $\mathbf{l}_i$
Object Point Clouds $\mathbf{p}_i$
Semantic Constraint Synthesis
Prompt: "The robot is holding a cup of water. Is it safe to pass over the laptop?"
LLM: "No, the water may spill."
Semantic constraint envelope defined via language
Geometric Constraint Synthesis
$\mathcal{C}_{\text{env}}, \mathcal{C}_{\text{self}}$ Geometric Constraints
$\mathcal{C}_{\text{sem}}$ Semantic Constraints
High-level Planner (Teleoperation or Motion Policy)
End Effector Velocity Cmd
$\dot{x}_{\text{ee, cmd}}$
Differential Inverse Kinematics
Joint Velocity Command
u_{cmd}
Semantic Safety Filter
Certified Joint Velocity Cmd
u_{cert}
Robot System
$q, \dot{q}$ Joint Position and Velocity
Robot Model

Figure 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.

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.

4 Methodology

In this section, we present the components of our proposed semantic safety filter framework. An overview of our proposed framework is shown 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 4.1). Then, a set of semantic constraints is synthesized using the semantic map and the LLM (Section 4.2). Finally, a semantic safety filter is formulated to account for the semantic constraints (Section 4.4).

4.1 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 [Gu et al., 2024, Takmaz et al., 2023] 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 [Kirillov et al., 2023], and every resulting segmentation mask is embedded [Radford et al., 2021] to generate segmented point clouds $\boldsymbol{p}_{f,i}$ and their associated class-agnostic 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 [Gu et al., 2024]. 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 [Takmaz et al., 2023, Gu et al., 2024], we map the embeddings to labels l_i by embedding a list of object classes and selecting the pair of embeddings with the highest similarity as the object's label.

4.2 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 manually specifying them would be tedious. Therefore, we employ an LLM to synthesize semantic constraints in an automated manner.

We design a language prompt for the LLM, which consists of multiple request-answer pairs as examples and a final request as the true query. Each request contains the following components: *(i)* a high-level description of the scene, *(ii)* the object the robot is manipulating, and *(iii)* a particular object that appears in the scene. The requests are repeated for every object in the scene. 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 behaviours 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 behaviours. Finally, the pose-based constraint set is $\mathcal{S}_{\text{T}}(o) = \{T\}$, where T specifies the end effector orientation constraint (`rotation locked` 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 $o = \texttt{cup of water}$ transportation example in the scene with only $l_0 = \texttt{laptop}$, we have $\mathcal{S}_{\text{r}}(o) = \{(\texttt{laptop}, \texttt{above})\}$, $\mathcal{S}_{\text{b}}(o) = \{(\texttt{laptop}, \texttt{caution})\}$, and $\mathcal{S}_{\text{T}}(o) = \{\texttt{rotation locked}\}$.

Our proposed semantic safety filter is designed based on the control barrier certification framework. In the following, we describe how we design the CBF safety filter using $\mathcal{S}(o)$. To facilitate our discussion, we first introduce the necessary notation for the robot kinematics. We denote the joint positions by $\boldsymbol{q} \in \mathbb{R}^n$ (with $n = 7$ in our case) and, similar to [Singletary et al., 2022], 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 [Lynch and Park, 2017]. 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.

4.2.1 Spatial Relationship Constraints

The semantic constraint sets are parameterized as the super-level 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

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

where $\boldsymbol{x}_{\text{ee}} = [x, y, z]^\intercal \in \mathbb{R}^3$ 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 can define the semantically unsafe sets as a differentiable approximation using a superquadric [Liu et al., 2022]:

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

where $\epsilon_{1,i}$ and $\epsilon_{2,i}$ define the shape of the superquadric and $a_{x,i}$, $a_{y,i}$, 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 by leveraging plane detection algorithms or learned segmentation models [Sharma et al., 2022]. To account for the spatial relationship `above`, we extend the point cloud in its positive z-direction. For the extension, we duplicate the point cloud and set the duplicate's z coordinates to be outside of the robot's workspace and fit the superquadric based on the union of the original and the expanded point cloud. For other spatial relationships such as `under` and `around`, we define similar superquadrics.

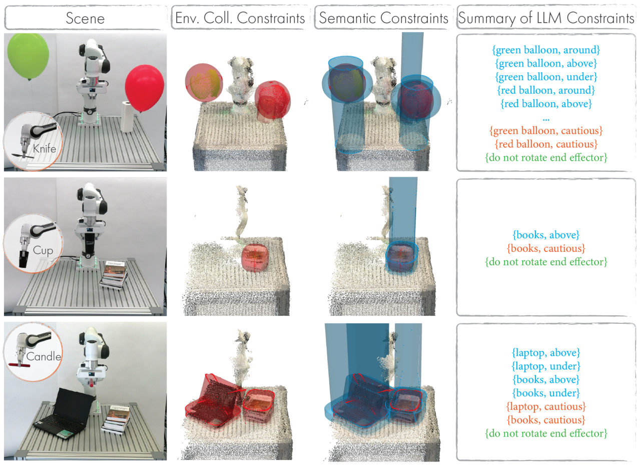

Figure 3: Examples of the environment collision and semantic constraints enforced by our proposed semantic safety filter. For each scene, environment collision constraints are generated based on the point clouds of individual objects while the semantic constraints are synthesized based on the point clouds 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), behavioural constraints (orange text), and end effector pose constraints (green text).

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

$$\boldsymbol{h}_{\text{sem},i}(\boldsymbol{x}_{\text{ee}}) = g_i(\boldsymbol{x}_{\text{ee}}; \boldsymbol{\theta}_1) - 1. \tag{1}$$

By using the forward kinematics model, we can express the semantic constraint set based on the CBFs in (1) in the robot's configuration space as

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

which yields our desired safe set.

4.2.2 Behavioral Constraints

The behavioral constraints are implemented using constraints on the time derivative of the CBF, which is the control invariance condition in [Ames et al., 2019]. The constraint is

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

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) [Ames et al., 2019]. 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 $\boldsymbol{\alpha}_{\text{sem},j}$. In that case, we also write $\alpha_{\text{sem},j}(\cdot; \texttt{caution}) = \alpha_{\text{sem,cautious},j}(\cdot)$. This reduction can be achieved by using a class $\mathcal{K}_\infty$ that is strictly smaller than $\alpha_{\text{sem},j}$ for positive $h_{\text{sem},j}$. A straightforward way to generate such a function is by multiplying the function $\alpha_{\text{sem},j}$ with a positive scalar less than 1.

4.2.3 Pose Constraints

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

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

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 $\boldsymbol{J}_o(q)$ 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 $(\cdot)^\wedge$ [Barfoot, 2024], and $\boldsymbol{\Delta\psi}_{\text{min}}$ and $\boldsymbol{\Delta\psi}_{\text{max}}$ 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 $T =$ `free rotation` (e.g., no object is being held) with $\boldsymbol{w}_{\text{rot}} = 0$, but $\boldsymbol{w}_{\text{rot}} > 0$ if $T =$ `minimize 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.

4.3 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 [Singletary et al., 2022].

4.4 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 a control input that best matches the desired input while ensuring all constraints are satisfied.

5 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 closed-loop to prevent potentially unsafe commands from a non-expert user or a motion policy.

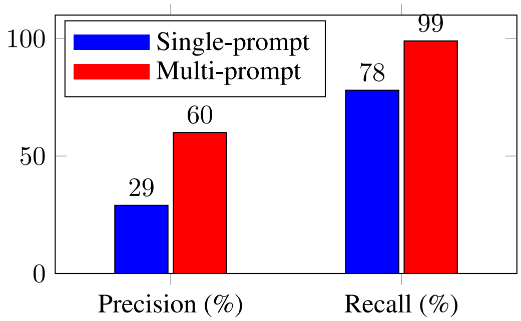

Figure 4: The multi-prompt strategy demonstrates improved precision and recall over the single-prompt method on our benchmarking dataset of ground-truth constraints.

5.1 Semantic Perception

In our evaluation, we consider four static (unless manipulated by the robot) tabletop scenes, which are visualized in Figure 3 and in Figure 7 and four manipulated objects a `dry sponge`, a `cup of water`, a `lit candle`, and a `knife`. The geometries of the table, the manipulated objects, and the robot are assumed to be known. However, the robot's relative position to the objects in the environment is unknown. The map for each tabletop environment is generated using RGB-D images and associated camera frames as described in Section 4.1. The RGB-D images were recorded using a Femto Bolt and the camera poses were obtained by running visual-inertial SLAM [Seiskari et al., 2022]. In total, each scene was reconstructed with approximately 50 to 200 posed RGB-D images, and the semantics are determined as described in Section 4.1. All computations were performed on a workstation with an Nvidia GPU RTX 3080. Examples of the reconstructed scenes are shown in Figure 3.

5.2 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. We evaluated two different prompting strategies on this dataset on an LLM (here, we use GPT-4o). The first strategy requested the full set $\mathcal{S}(o)$ at once, while the second strategy would only request one pair or a singleton (for the semantic pose constraint) for each request. In the following, we refer to these strategies as single- or multi-prompt strategies, respectively. The multi-prompt method proved to be more accurate than the single-prompt approach, as indicated by the higher precision and recall in Figure 4. The final prompt was adjusted until the desired level of accuracy was achieved on the validation dataset split.

For our robot experiments, we follow the methodologies in Section 4.2 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 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 w_{rot}.

5.3 Safe Robot Manipulation

Using our semantic safety filter, we execute various teleoperation and pick-and-place tasks on the robot. We run our semantic safety filter at 30 Hz. Our teleoperation experiments are summarized in Table 1. The teleoperation commands are provided through a keyboard 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

Table 1: 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. [†]The objects in red result in semantic constraints.

Scene	Held Obj.[†]	No Safety Filter	Nom. Safety Filter	**Ours**
`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%**
`balloon,tissue`	`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%**

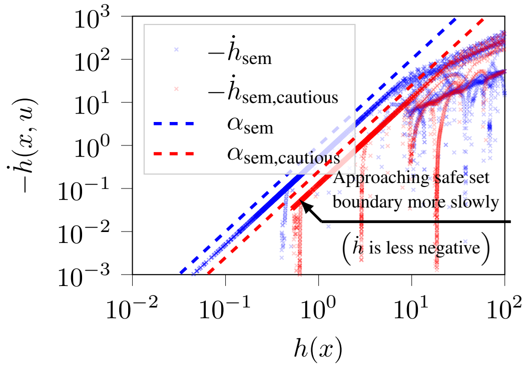

Figure 5: The level of caution determines how quickly the end effector approaches a safety constraint boundary. In the `books` scene, we increase caution by adjusting the class $\mathcal{K}_\infty$ function when holding a `cup of water` under the same semantic constraint .

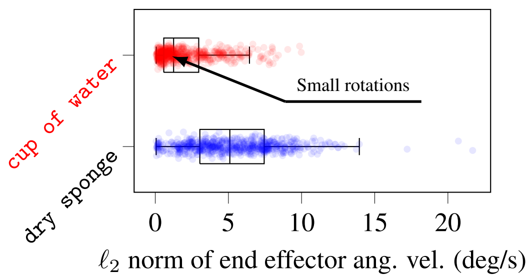

Figure 6: Demonstration of the active (inactive) rotation minimization when the robot is holding a `cup of water` (`dry sponge`) in the scene `books`. The distribution for the `cup of water` is skewed towards smaller angular velocities; an active rotation minimization (red) generally yields reduced end 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 $\alpha_{\text{sem,cautious}}(h) = \frac{1}{4}h^2$ 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 4.2.

Finally, we demonstrate the effectiveness of minimizing rotations for different objects based on their semantics in Figure 6. Our semantic safety filter successfully reduces the median of the norm of the end effector's angular velocity by 75.39% if the rotation minimization 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`.

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 ob-

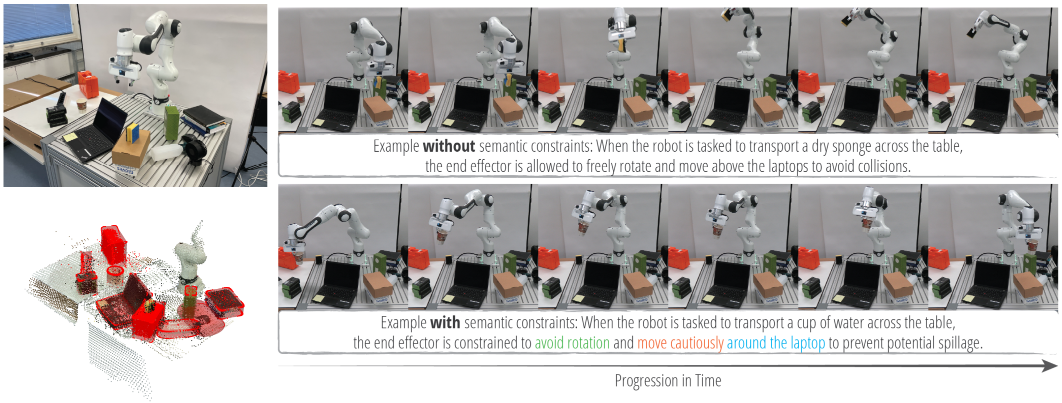
Example **without** semantic constraints: When the robot is tasked to transport a dry sponge across the table,
the end effector is allowed to freely rotate and move above the laptops to avoid collisions.
Example **with** semantic constraints: When the robot is tasked to transport a cup of water across the table,
the end effector is constrained to avoid rotation and move cautiously around the laptop to prevent potential spillage.
Progression in Time

Figure 7: The cluttered environment and its superquadric representation *(left)* used in the pick-and-place tasks and associated manipulation sequences *(right)*. The scene has 17 objects of various types, some of which are stacked *(left, top)*. The scene is represented using fitted superquadrics shown in red *(left, bottom)*. Examples of tabletop manipulation sequences without semantic constraints *(right, top)* and with semantic constraints *(right, bottom)* in a cluttered scene. In the top row, the robot is tasked to transport a dry sponge across the table from left to right. There is no unsafe semantic constraint between the manipulated object and the objects in the scene. The robot's end effector is allowed to rotate freely and move above the laptop while avoiding collisions. On the other hand, in the bottom row, the robot is tasked to transport a cup of water across the table from right to left. One semantically unsafe condition is {laptop, above}. Moreover, the robot is also required to move cautiously and without rotating the end effector. With these semantic constraints, the robot end effector slowly moves around the laptop to avoid the potential spillage of water over the laptop.

jects, including a laptop, a mouse, a power strip, a fan, a a dry sponge, a cup of water, a bottle, a first-aid kit, books, and boxes (see Figure 7). The robot is tasked to move the dry sponge and then the cup of water across the table. For the sequential pick-and-place tasks, we use an underlying resolved-rate motion controller that is unaware of any safety constraints and certify the action using our proposed safety filter. Two representative sequences of motion are depicted in Figure 7 to highlight the effect of applying the semantic filter to different manipulated object cases. 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; in this case, the robot is constrained to not pass over the electronic devices nor rotate, and overall execute the task more cautiously.

6 Conclusion

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.

References

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. In *Proc. of the Conference on Robot Learning (CoRL)*, 2022.

Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. Control barrier functions: Theory and applications. In *Proc. of the European Control Conference (ECC)*, pages 3420–3431, 2019.

Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-jacobi reachability: A brief overview and recent advances. In *IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253, 2017.

Timothy D Barfoot. *State Estimation for Robotics*. Cambridge University Press, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proc. of the Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 1877–1901, 2020.

Lukas Brunke, Melissa Greeff, Adam W Hall, Zhaocong Yuan, Siqi Zhou, Jacopo Panerati, and Angela P Schoellig. Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:411–444, 2022.

Arthur Bucker, Luis Figueredo, Sami Haddadin, Ashish Kapoor, Shuang Ma, Sai Vemprala, and Rogerio Bonatti. LATTE: Language trajectory transformer. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 7287–7294, 2023.

Jia-Ruei Chiu, Jean-Pierre Sleiman, Mayank Mittal, Farbod Farshidian, and Marco Hutter. A collision-free MPC for whole-body dynamic locomotion and manipulation. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 4686–4693, 2022.

Jonathan Crespo, Jose Carlos Castillo, Oscar Martinez Mozos, and Ramon Barber. Semantic information for robot navigation: A survey. *Applied Sciences*, 10(2):497, 2020.

Qiao Gu, Alihusein Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Concept-Graphs: Open-vocabulary 3D scene graphs for perception and planning. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.

Lin Guan, Yifan Zhou, Denis Liu, Yantian Zha, Heni Ben Amor, and Subbarao 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.

Kai-Chieh Hsu, Haimin Hu, and Jaime F Fisac. The safety filter: A unified view of safety-critical control in autonomous systems. *Annual Review of Control, Robotics, and Autonomous Systems*, 7, 2023.

Chenguang Huang, Oier Mees, Andy Zeng, and Wolfram Burgard. Visual language maps for robot navigation. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 10608–10615, 2023a.

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. *arXiv preprint arXiv:2207.05608*, 2022.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. VoxPoser: Composable 3d value maps for robotic manipulation with language models. In *Conference on Robot Learning*, pages 540–562. PMLR, 2023b.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment Anything. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.

Stefan Leutenegger. Okvis2: Realtime scalable visual-inertial slam with loop closure, 2022. URL `https://arxiv.org/abs/2202.09199`.

Cici Li and Guohui Tian. Transferring the semantic constraints in human manipulation behaviors to robots. *Applied Intelligence*, 50(6):1711–1724, 2020.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, 2024.

Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Robust and accurate superquadric recovery: A probabilistic approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2676–2685, 2022.

Kevin M Lynch and Frank C Park. *Modern Robotics*. Cambridge University Press, 2017.

Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. OpenScene: 3D scene understanding with open vocabularies. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–824, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proc. of International Conference on Machine Learning (ICML)*, pages 8748–8763. PMLR, 2021.

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko 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.

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 *Proc. of the Conference on Robot Learning (CoRL)*, 2023.

Nathan D Ratliff, Jan Issac, Daniel Kappler, Stan Birchfield, and Dieter Fox. Riemannian motion policies. *arXiv preprint arXiv:1801.02854*, 2018.

Antoni Rosinol, Marcus Abate, Yun Chang, and Luca Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, 2020.

Otto Seiskari, Pekka Rantalankila, Juho Kannala, Jerry Ylilammi, Esa Rahtu, and Arno Solin. HybVIO: Pushing the limits of real-time visual-inertial odometry. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 701–710, January 2022.

Gopal Sharma, Bidya Dash, Aruni RoyChowdhury, Matheus Gadelha, Marios Loizou, LiangLiang Cao, Rui Wang, Erik G Learned-Miller, Subhransu Maji, and Evangelos Kalogerakis. Prifit: Learning to fit primitives improves few shot point cloud segmentation. In *Computer Graphics Forum*, volume 41, pages 39–50. Wiley Online Library, 2022.

Andrew Singletary, William Guffey, Tamas G Molnar, Ryan Sinnet, and Aaron D Ames. Safety-critical manipulation for collision-free food preparation. *IEEE Robotics and Automation Letters*, 7(4):10954–10961, 2022.

Mark W Spong. An historical perspective on the control of robotic manipulators. *Annual Review of Control, Robotics, and Autonomous Systems*, 5:1–31, 2022.

Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *Proc. of the Conference on Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Kim P Wabersich, Andrew J Taylor, Jason J Choi, Koushil Sreenath, Claire J Tomlin, Aaron D Ames, and Melanie N Zeilinger. Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023.

Johanna Wald, Helisa Dhamo, Nassir Navab, and Federico Tombari. Learning 3D semantic scene graphs from 3D indoor reconstructions. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3961–3970, 2020.

Amber Xie, Youngwoon Lee, Pieter Abbeel, and Stephen James. Language-conditioned path planning. In *Conference on Robot Learning*, pages 3384–3396. PMLR, 2023.