2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

SPOTS: Stable Placement of Objects with Reasoning in Semi-Autonomous Teleoperation Systems

Joonhyung Lee[1], Sangbeom Park[1], Jeongeun Park[1], Kyungjae Lee[2], and Sungjoon Choi[1]

***Abstract*— Pick-and-place is one of the fundamental tasks in robotics research. However, the attention has been mostly focused on the "pick" task, leaving the "place" task relatively unexplored. In this paper, we address the problem of placing objects in the context of a teleoperation framework. Particularly, we focus on two aspects of the place task: stability robustness and contextual reasonableness of object placements. Our proposed method combines simulation-driven physical stability verification via real-to-sim and the semantic reasoning capability of large language models. In other words, given place context information (e.g., user preferences, object to place, and current scene information), our proposed method outputs a probability distribution over the possible placement candidates, considering the robustness and reasonableness of the place task. Our proposed method is extensively evaluated in two simulation and one real world environments and we show that our method can greatly increase the physical plausibility of the placement as well as contextual soundness while considering user preferences. Code, video, and details are available at: https://joonhyung-lee.github.io/spots/**

I. Introduction

Providing a small number of effective options to users in an autonomous system (i.e., a semi-autonomous teleoperation system) has been actively studied [1]–[3] to mitigate possible malfunctioning of a fully autonomous system. However, these options often fall short of considering the situational context. In this paper, we focus on the task of placing objects considering two different aspects, physical robustness and reasonableness considering contextual information, within a semi-autonomous teleoperation framework to provide effective place candidates to a user.

To achieve robust placement, we utilize simulators to forward simulate multiple place candidates by assuming that the simulation environment is reconstructed from the real world observations (i.e., real-to-sim). Contextual soundness is mostly achieved via the reasoning capabilities of large language models (LLMs). To be specific, scene information (e.g., object to place and other objects in the scene) and optional human preference (e.g., "I would like to sort objects based on colors.") are fed into LLMs via vision-language models to restrict the possible place locations.

While the task of picking and placing objects is a fundamental task, more research focus has been concentrated on picking objects [4], [5]. However, we argue that the place task is also an essential problem to consider, especially when it comes to semi-autonomous teleoperation (e.g., recommending users a set of appropriate place locations). Unlike the picking task, where the target object to pick is usually predetermined, there may exist multiple possible placeable locations, and this makes it more attractive for the semi-autonomous teleoperation framework to be utilized.

We mainly focus on two aspects of the place task: physical stability and contextual soundness of the placement. The former is achieved via simulation-driven verification combined with real-to-sim. In other words, given the current scene, we first reconstruct the scene using a physics-based simulator (e.g., MuJoCo [6]) and check the stability by solving forward dynamics with additional perturbation of the object poses. The latter uses the reasoning performance of large language models (LLMs). In particular, the current scene and the object to be placed are described with languages utilizing vision-language models (e.g., OWL-ViT [7]). Then, the place candidates that passed the physical stability check are examined using the LLM by simply adding the prompt describing the current place candidate to the scene-describing prompt.

In summary, we made two key contributions to this paper. We present **S**table **P**lacement of **O**bjects with reasoning in semi-autonomous **T**eleoperation **S**ystems (**SPOTS**), an approach to a semi-autonomous teleoperation framework that focuses on verifying placement positions with 1) a **Stability Verification** (i.e., physics-based simulation) step and 2) a **Receptacle Reasoning** (i.e., common knowledge) step by utilizing LLMs that understand scene contexts and reason about the corresponding task without learning.

II. Related Work

A. Semi-Autonomous Teleoperation

Recently, research on semi-autonomous teleoperation has been actively conducted to minimize human burden while effectively teleoperating a robot. In [1], [2], Losey et al.

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00079 , Artificial Intelligence Graduate School Program (Korea University)), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00871, Development of AI Autonomy and Knowledge Enhancement for AI Agent Collaboration), Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00612, Geometric and Physical Commonsense Reasoning based Behavior Intelligence for Embodied AI), and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2022-0-00480, Development of Training and Inference Methods for Goal-Oriented Artificial Intelligence Agents)

[1]Joonhyung Lee, Sangbeom Park, Jeongeun Park and Sungjoon Choi are with the Department of Artificial Intelligence, Korea University, Seoul, Korea (email: {dlwnsgud8823, sangbeom-park, baro0906, sungjoon-choi}@korea.ac.kr)

[2]Kyungjae Lee is with the Department of Artificial Intelligence, Chungang University, Seoul, Korea (email: kyungjae.lee@ai.cau.ac.kr)

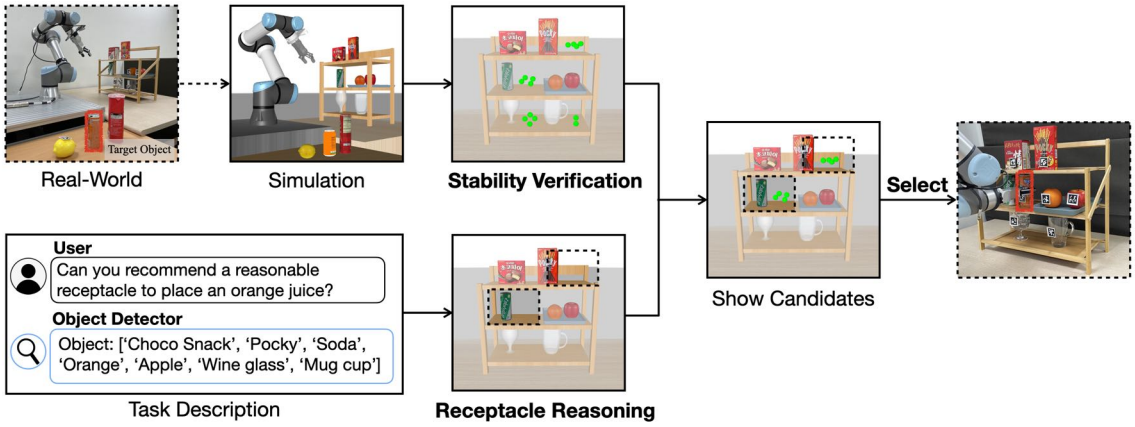

Fig. 1: Overall pipeline of the proposed teleoperation framework. With the scene input, the system checks the physical stability of the correct placement. Then, the system verifies the contextually reasonable positions based on the receptacle reasoning step, considering the scene's context, and recommends the coordinates obtained from both processes to the user.

have addressed a framework to reduce the dimensionality of control inputs for efficient control of high-dimensional robots by embedding high-dimensional trajectories to a low DoF latent actions. However, it uses kinesthetic demonstrations to show the robot arm how to perform a variety of tasks. Park et al. [3] have performed a non-prehensile manipulation task using a high-level option via reinforcement learning (RL) (e.g., pushing away obstacles in a cluttered environment). However, training a policy to handle a complex task, such as placing a book on a bookshelf, is exceedingly challenging. Acknowledging these limitations, our method provides various candidates for where to place in a given scene, which is validated in the physics-based simulation environment and fits with common sense using LLMs [8]–[10].

B. Leveraging Simulation for Interactions

Physics-based simulators have made significant progress in the robotics field [6], [11]–[13]. In addition, there have been only a few studies that have explored the translation from real world to simulation environments, in contrast to Sim2Real, with a focus on physical interactions using a physics-based simulator [14]–[17]. Lv et al. [16] have conducted a sensing-aware approach to find the robot's next best viewpoint posture in simulation based on the image taken from a specific viewpoint in the real world (i.e., transferring the real world to simulation and updating the posture based on the value function). Mo et al. [17] proposed a pipeline to leverage differentiable simulation that creates an affordance map by checking the behavior of objects before and after interaction through a simulator when performing a robotic task. Similarly, we use the physics-based simulator to validate the interaction of placing objects in the simulator. This allows us to ensure physical plausibility and verify stability where the target object should be placed.

C. Reasoning capabilities of LLMs for Robotics

Alongside the recent success of LLMs [9], [18], [19], LLMs have been actively studied in robotics. Some methods [20], [21] utilize LLMs for task planning a robot, where LLM predicts a sequence of low-level subgoals that are driven by prompt structures. In [22], [23], they use LLMs to write robot policy codes given language commands by prompting the model with a few demonstrations. LLMs have also been utilized as a reward function for inferring user intentions in negotiation games [24], collaborative human-AI interaction games [25], and automating parameterization of reward functions [26]. These methods leverage the semantic priors stored in LLMs to compose new plans or parameterize primitive APIs. Mirchandani et al. [27] have utilized the LLMs as a pattern machine for spatial rearrangement via predicting simple forward dynamics (e.g., moving a red bowl to a green plate). Inspired by this, we validate the physical plausibility by placing a target object directly in a simulation environment, and we leverage the reasoning capabilities of LLMs to screen out unreliable receptacles considering the spatial relationships (e.g., 6D pose and size of objects) and semantic representations (e.g., color, and flavor).

III. Problem Formulation

We focus on the problem of recommending object place locations to users in the context of semi-autonomous teleoperation. Specifically, we target this issue in complicated and restricted environments with limited free space for placement, such as placing a plate in a dish rack [28] or a book on a shelf [29], [30]. Given an RGBD image as an input, we aim to predict a set of candidate placement locations that balance the physical stability and reasoning suitability. Here, physical stability (i.e., robustness) refers to whether an object will stand up as intended without falling over when placed in its space. Reasoning suitability (i.e., contextual reasonableness) involves whether the object is placed in a reasonable space. To accomplish this task, our proposed framework takes as inputs a set of three-dimensional points $\mathcal{P} = \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_n\}$ in $\mathbb{R}^3$ that exist within the robot's accessible workspace, where $\mathbf{p}_i = (x_i, y_i, z_i)$, as well as a task description that specifies the user's requirements. The framework outputs a density distribution $\mathcal{D}$ to identify

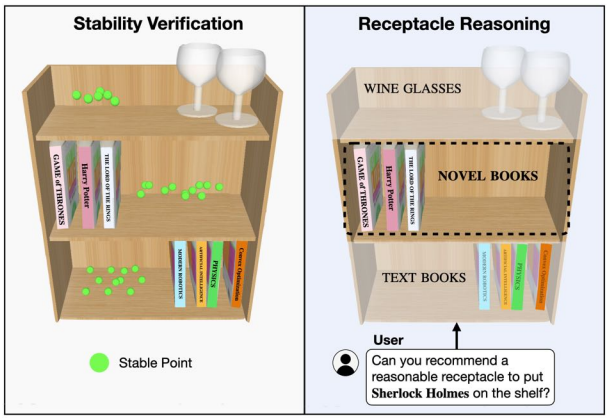

Fig. 2: Description of the modules for stability verification and receptacle reasoning. The stability verification module determines if a location is physically stable, and the receptacle reasoning determines the appropriate receptacle.

regions where objects can be stably placed.

IV. Proposed Method

In a complex environment, there are a limited number of options to place objects (e.g., placing a plate on a plate rack, placing a book on a bookshelf, or placing an item on a shelf according to its categorization). The goal is to find a probability distribution over those coordinates, not just a finite, limited, fixed set of coordinates that can be physically placed. Our framework has two modules: a) the physics-based screening module (i.e., real-to-sim), which will be introduced in Section IV-A, and b) the reasoning-based suitability check module, which will be illustrated in Section IV-B. We would like to emphasize that the robot agent only provides the user with coordinates for the placement of the object, and the user chooses the most appropriate coordinate based on their own needs or preferences. The overall pipeline is illustrated in Figure 1.

A. Physics-Based Stability Verification

The first step of the proposed stability verification is the 3D reconstruction of the real world environment to a physics-based simulator. To accomplish this using an ego-centric RGBD image, two key factors must be considered: the objects' spatial and semantic configuration. The spatial arrangements can be measured from depth information based on the camera parameters (i.e., intrinsic matrix and extrinsic matrix). In addition, the semantic attribute of the objects in the scene is captured from object detection models (e.g., Vision Language Models [7], [31]–[33]), which identify and cluster objects in the scene with zero-shot manners. Based on spatial and semantic configuration, we match each object with its corresponding mesh model, which we assume is available. These mesh models are then parsed into the simulator to closely mimic the real world environment.

We aim to identify regions where objects can be stably placed over a given interaction time T in simulation. For each point $\mathbf{p}_i$ and at each time $t = 1, 2, \ldots, T$, we have a sequence of scalar components of quaternions belonging to object orientations, denoted as $q(t, \mathbf{p}_i)$. We define the set of points $\mathcal{P}_\text{s}$ to represent coordinates where objects can be placed stably:

$$\mathcal{P}_\text{s} = \left\{ \mathbf{p}_i \in \mathcal{P} : \forall t, \|q(t, \mathbf{p}_i) - q(1, \mathbf{p}_i)\|^2 \leq \sigma^2 \right\} \quad (1)$$

where $q(1, \mathbf{p}_i)$ is the quaternion scalar component at $t = 1$ for each point $\mathbf{p}_i$, and σ^2 is the variance of $\{q(t, \mathbf{p}_i)\}_{t=1}^{T}$. More specifically, to determine the robustness of the placement stability, small perturbations are injected after the object has been placed. These perturbations can imitate real world conditions where external factors, such as vibrations or impacts, may affect the stability of the placed object. If the objects can tolerate these perturbations, the robustness and real world applicability of placement tasks will be increased.

Specifically, for all $\mathbf{p} \in \mathcal{P}_s$, to compute the stability reward $\text{r}_\text{s}(\mathbf{p})$ for each coordinate, we use a simple linear reward function as follows:

$$\text{r}_\text{s}(\mathbf{p}) = 100 \cdot (q_\text{max}(\mathbf{p}) - q_\text{min}(\mathbf{p})) \quad (2)$$

where $q_\text{max}(\mathbf{p})$ and $q_\text{min}(\mathbf{p})$ are the maximum and minimum quaternion scalar components associated with point $\mathbf{p}$, respectively.

B. Context Understanding via Reasoning

Though the set of $\mathcal{P}_\text{s}$ points, that are verified in Sec IV-A, are determined to be feasible, it may contain some options that do not take into account the context of the scene (e.g., putting a book on a desk instead of a bookshelf, putting an item in a different category when it should be categorized). Therefore, we aim to analyze the reasonableness within the limited range of $\mathcal{P}_\text{s}$ that corresponds to the current scene's situation and context. An LLM [18] uses the input of object labels from the same RGBD image in Sec. IV-A and generates one or more words representing the specific receptacle where the object could reasonably be located (i.e., reasonableness).

Since we have transferred the robot's ego-centric view from the real world to the simulator in Sec IV-A, we can automatically identify the region where the coordinates of $\mathcal{P}_\text{s}$ are located. Based on the generated receptacle, $\mathcal{P}_\text{r}$ is a set of points that are within 0.1 meters of the receptacle object, which is included in $\mathcal{P}_\text{s}$. For all $\mathbf{p} \in \mathcal{P}_\text{r}$, the reward $r_\text{r}(\mathbf{p}) = +1$ when the receptacle is a reasonable region, $r_\text{r}(\mathbf{p}) = 0$ otherwise.

C. *SPOTS*: Reward-Guided Sampling

With a set of rewards based on both the stability verification step in Sec. IV-A and receptacle reasoning Sec. IV-B, the final procedure is to generate candidate place locations. To characterize the distribution of feasible and suitable 3D points for object placement, we utilize a kernel density estimation (KDE) [34]. Specifically, we use the radial basis function (RBF) kernel. The KDE for each point cloud is

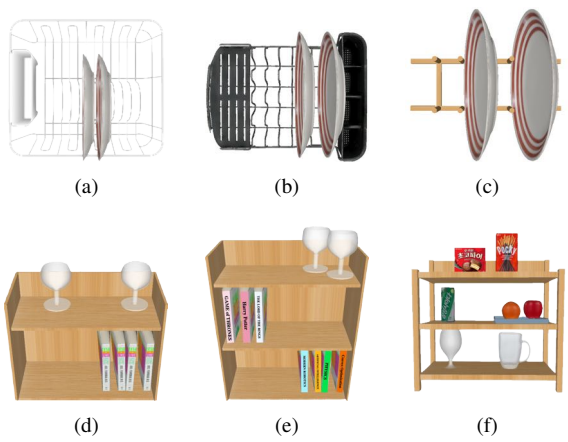

Fig. 3: Randomization for the Dish Rack environment is (a) Small Gap, (b) Medium Gap, (c) Large Gap. Randomization for the Bookshelf environment is (d) Two-Tiered Bookshelf, (e) Three-Tiered Bookshelf. Category environment uses (f) Three-Tiered Shelf.

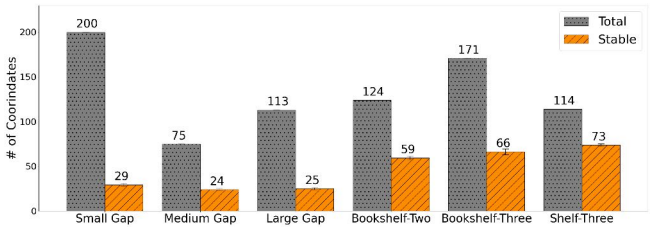

Fig. 4: Result of stability verification module: Performed for all environments in our experiments, not including reasoning module. The ratio of stable coordinates to the total number of coordinates is very low. This indicates that the task we are assuming is physically difficult to be stably located.

defined as:

$$\hat{f}_h(\mathbf{p}, r(\mathbf{p})) = \frac{1}{N \cdot h} \sum_{\mathbf{p}_i \in \mathcal{P}} r(\mathbf{p}_i) \cdot K\left(\frac{\mathbf{p} - \mathbf{p}_i}{h}\right) \quad (3)$$

where N is the total number of coordinates, h is the bandwidth parameter, K is the kernel function, $\mathbf{p}$ is a random variable, $r(\mathbf{p}_i)$ is the weight of the i^{th} coordinate in $\mathcal{P}$, and $\{\mathbf{p}_i\}_{i=1}^N$, $\{r(\mathbf{p}_i)\}_{i=1}^N$ are obtained in previous steps.

To emphasize the reward values for the receptacles within the set of physically stable coordinates $\mathcal{P}_s$, we formulate the combined density distribution $\mathcal{D}_c$ as follows:

$$\mathcal{D}_c(\mathbf{p}) = \hat{f}_h(\mathbf{p}_s, r_s(\mathbf{p}) \cdot \beta + r_r(\mathbf{p}) \cdot (1 - \beta)) \quad (4)$$

where r_s and r_r represent the rewards associated with each point $\mathbf{p}_s$ and $\mathbf{p}_r$ in $\mathcal{P}_s$ and $\mathcal{P}_r$, respectively, and $\mathbf{p}$ is the union of $\mathcal{P}_s$ and $\mathcal{P}_r$. Importantly, sets $\mathcal{P}_r$ and $\mathcal{P}_s$ are mutually exclusive, enabling focused design for either physical stability or reasonableness without conflicting effects. The parameter β is tunable and balances between physical stability and reasonableness. Although β is tunable, it was held constant at 0.1 in all experiments.

To summarize, based on contextual information from the input task description, such as user preferences and the current scene, our method generates a combined density $\mathcal{D}_c$ for candidate placements. This density not only accounts for the physical robustness and reasonableness of the placement task but also aligns with the likelihoods associated with the task description and receptacle. A higher $\mathcal{D}_c$ indicates a better fit with these factors. From this combined distribution, we sample probable placements, allowing the user to choose the one that maximizes our density function.

V. Experiments

We conduct experiments in both simulation and real world settings to accomplish distinct yet complementary goals. In simulation, our objective is to evaluate the effectiveness of our stability verification process. Our real world experiments aim to assess SPOTS's ability to accurately infer the appropriate location using both spatial and semantic reasoning via LLM. For comparison with other LLM-based baselines, we also measured the total number of input tokens and the time spent on interactions.

We designed a task that categorizes objects based on similarity. The reasoning criteria, termed **similarity**, varies for each experiment and serves as the ground truth for evaluating reasoning abilities. To be more specific, we validate our algorithm on three scenarios. The scenario involves a) placing plates on a dish rack, b) placing books on a bookshelf (in simulation), and c) categorizing objects based on similarity (both in the simulation and real world). The objects vary in size, shape, and position with each scenario. Our evaluations are twofold: (1) we present **SPOTS**'s place stability in simulation environments (Section V-C), and (2) we showcase the potential of **SPOTS** in the semi-autonomous teleoperation framework (Section V-D) when reasoning capability is required (i.e., semantic reasoning or common knowledge).

A. Environment setup

Our real-to-sim transfer module, illustrated in Fig.1, utilizes OWL-ViT [7] for open-vocabulary object detection and AprilTags [36] for pose estimation, based on input from an RGBD vision sensor. The detected objects form a label super-set that includes nine categories[1], for a total of 21 object assets. For each detected object, we assume the corresponding 3D asset is available. These assets are transferred into a simulation environment that mimics the real world as closely as possible. This reconstructed environment is the basis for all subsequent evaluations. The framework is built on the MuJoCo [6] simulator, using assets from the YCB [37] and Google Scanned dataset [38]. We use a tabletop manipulation framework with a 6-DoF robot arm and `gpt-3.5-turbo` [39].

[1]'DishRack', 'Bowl', 'BookShelf', 'Fruit', 'Beverage', 'Snack', 'Tray', 'Glass', 'Book'

Simulation Experiments											
Model	Dish Rack Environment										
	Small Gap Dish Rack, Fig. 3a			Medium Gap Dish Rack, Fig. 3b			Large Gap Dish Rack, Fig. 3c				
	Time [s]	S/R	# of token	Time [s]	S/R	# of token	Time [s]	S/R	# of token		
LLM-GROP [35]	1.78 ± 0.29	0.80	231	1.92 ± 0.43	0.80	253	1.72 ± 0.35	0.9	244		
CaP [22]	25.32 ± 5.91	0.60	823	13.97 ± 1.12	0.65	792	12.46 ± 4.05	0.75	949		
L2R [26]	5.16 ± 0.35	0.70	1453	4.58 ± 0.44	0.70	1306	5.21 ± 0.24	0.80	1454		
SPOTS (Ours)	3.70 ± 0.04	**0.85**	-	1.48 ± 0.04	**0.85**	-	5.41 ± 0.02	**0.95**	-		
Model	Bookshelf Environment								Category Environment		
	Two-Tiered Bookshelf, Fig. 3d				Three-Tiered Bookshelf, Fig. 3e				Three-Tiered Shelf, Fig. 3f		
	Time [s]	Sta. S/R	Rea. S/R	# of token	Time [s]	Sta. S/R	Rea. S/R	# of token	Time [s]	Sta. S/R	Rea. S/R	# of token
LLM-GROP [35]	1.90 ± 0.45	0.65	0.80	428	2.90 ± 0.17	0.65	0.80	658	1.95 ± 0.61	0.70	0.75	451
CaP [22]	6.98 ± 0.22	**0.95**	**0.95**	985	6.87 ± 1.80	0.85	0.65	1168	4.42 ± 0.26	0.70	0.6	1423
L2R [26]	6.24 ± 0.87	0.75	0.80	1153	7.78 ± 1.04	0.8	0.80	1493	7.63 ± 1.63	0.75	0.75	1205
SPOTS (Ours)	3.77 ± 0.03	0.90	0.95	323	4.50 ± 0.01	**0.90**	**0.85**	430	9.07 ± 0.10	**0.85**	**0.85**	342

TABLE I: Simulational Result: Experiments with six scenarios for a total of 3 environments. The Dish Rack scene was set up with different shapes and sizes of the racks. The Bookshelf scene was varied with diverse shelf sizes, positions, and genres. The Category scene was randomized using **similarity** criteria.

B. Baselines & Metrics

We compare SPOTS to three prior methods: LLM-GROP [35], Code-as-Policies (CaP) [22], and Language-to-Reward (L2R) [26]. LLM-GROP uses two different template-based prompts; one extracts semantic relationships with examples, and the other one predicts geometric spatial relationships for varying scene geometry. CaP generates policy code for the robot motion using a pre-defined low-level primitive function. L2R defines reward parameters that can be optimized, and the reward function is designed for moving a manipulator to a parameterized placement position.

Our evaluation metrics are the place stability and reasonableness of the suggested object placements. The stability success rate is based purely on the physical stability of object placement in simulations, whether that object is placed stable (i.e., Sta. S/R). Reasonableness success rate (i.e., Rea. S/R), on the other hand, is based on whether object placement aligns with the ground truth that we define. Evaluating reasonableness success criteria is manually designed; more details are described in Sec V-C. These metrics assess the overall effectiveness of placements in ensuring both stability and reasonableness. These specific criteria are the ground truth for confirming appropriate locations in our experimental validation. Furthermore, we measure the time taken for the inference and the number of input and output tokens to measure the efficiency of utilizing LLMs.

C. Simulation Experiments

We conducted experiments in three different scenarios within the simulation environment: a) Dish rack, b) Bookshelf, and c) Category scenario. For each setting, we randomized the environment to evaluate the performance. SPOTS samples 10 coordinates per experiment and repeats a total of 20 times. The baseline model has 20 interactions.

a) Dish Rack: On Dish Rack experiments, we focused solely on evaluating the place stability of objects, assuming that the receptacle locations are known and accurate. Baseline models were also evaluated under the same conditions for a fair comparison. There were three different types and sizes of racks used for randomization. From Fig. 4, we can infer that the number of feasible coordinates in the Dish Rack scenario is small and that this task is difficult to perform without physical simulation. From Table I, the success rate for all the compared methods is over 60%, when highly detailed spatial information is given. The SPOTS outperforms the compared method with a margin of 0.05 on the success rate, indicating that the proposed method can robustly place the object in a complex environment.

b) Bookshelf: The Bookshelf experiment assesses reasoning ability and consists of two scenes. In the two-tiered bookshelf environment, books are on the first (lower) tier, wine glasses are on the second (upper) tier, and we define the reasonable location as the first tier of the bookshelf. In a three-tiered bookshelf, the first and second tiers each contain books of the same genre (e.g., textbook, novel, fairy-tale), and the third tier contains a wine glass. In this case, the tier that matches the genre with the placement object is denoted as ground truth. The experimental results on Table I show that SPOTS outperforms the baselines in Sta. S/R and Rea. S/R with a gap of 0.05 and 0.00, respectively. While CaP [22] is observed to have comparable performance in two-tiered bookshelf scenarios, SPOTS reduces the number of prompts needed from 985 to 323. This disparity arises because CaP requires carefully tuned few-shot prompts, whereas SPOTS can execute the task without needing precise explanations or hand-crafted examples to perform the task.

c) Category: Finally, we designed a task where objects were categorized based on **similarity**. We define three types of similarity here: color, object property, and shape, which serve as the ground truth for evaluating reasoning abilities. The reasonable receptacle changes based on this **similarity** type (i.e., the ground truth of the Rea. S/R changes based on the similarity). The scenario consists of a three-tiered shelf, each with different objects (e.g., glass cup, beverage, snack, fruit). In Table I-Category, SPOTS surpasses all compared methods by a margin of 0.15 and 0.10 in terms of Sta. S/R and Rea. S/R, respectively. By separating the tasks of predicting receptacles and ensuring physical robustness into two distinct modules, we find that SPOTS achieves a higher success rate while using fewer tokens compared to the methods that enforce LLMs to predict both robotic plans while

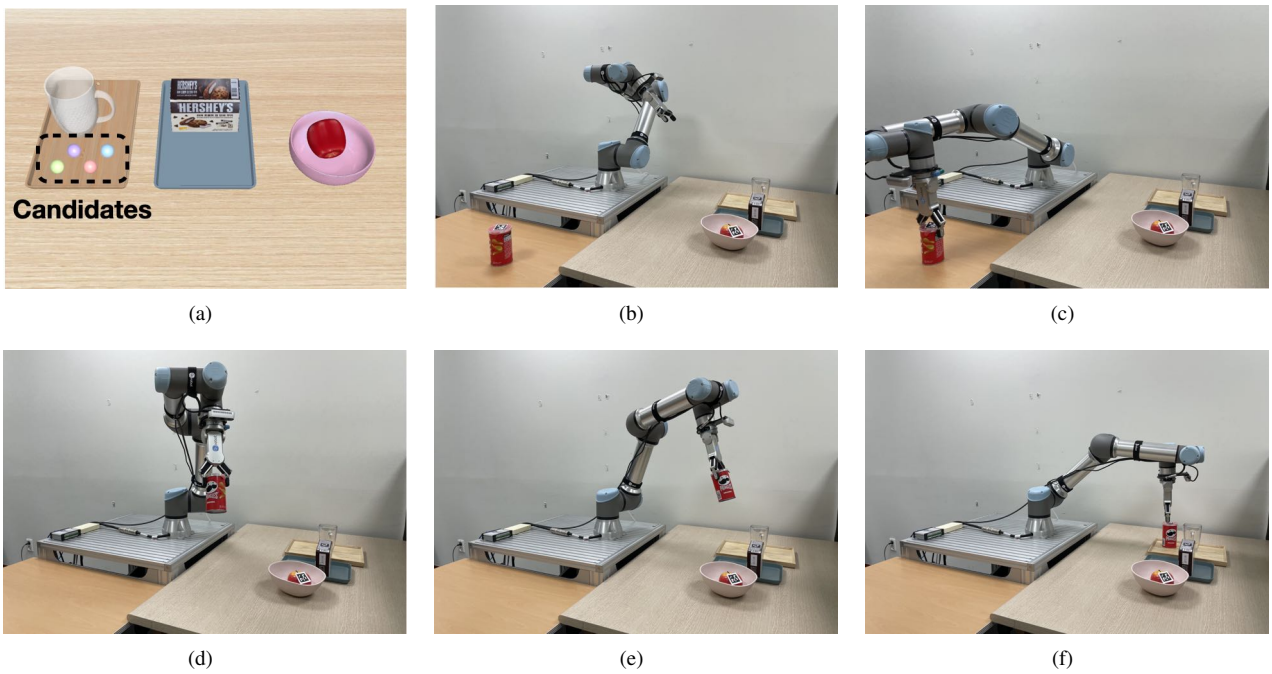

Fig. 5: Snapshots of real robot experiments: (a) shows the user interface from the perspective of the user, (b)-(f) show the robot performs a promptable placement task based on shape **similarity**.

Real World Demonstration						
Model	Tray					
	Shape			Object Property		
	Time [s]	S/R	# of token	Time	S/R	# of token
SPOTS (Ours)	3.58 ± 0.08	0.80	362	5.69 ± 0.04	0.80	405

TABLE II: Real World Result: Experiments with the shape and object property as **similarity**.

understanding the context. From this experiment, we would like to posit that SPOTS has great capability of promptable placement tasks, which considers both physically stable and reasonable regions, and SPOTS has a good distribution, where reasonable positions can be sampled.

D. Real World Demonstration

In this experiment, we consider a scene with three different trays placed on a desk, each containing a single object. The target object for placement is a potato snack. Each type of similarity was evaluated five times, and performance was measured using the overall success rate (i.e., both Sta. S/R and Rea. S/R). In both the shape and object property cases, the success rate was 0.8, indicating a generally effective reasoning process. The failure cases occurred when incorrect matches were made due to incorrect reasoning. From this experiment, we insist that the reasonable place varies depending on the task description given as input. Furthermore, we are able to accurately determine the stable positions to place the objects by reconstructing the robot's ego-centric view with the real-to-sim method. The β term also allowed us to control the relationship between the physically stable and semantically reasonable locations.

E. Limitations

Since our model utilizes LLMs, we need a language projection phase, which requires an accurate open vocabulary object detection model as well as knowing the superset of possible objects in the scene. This may hinder the usage of SPOTS in a novel environment. Moreover, for the sake of real-to-sim, the current version of our implementation requires CAD models to reconstruct the scene in a physics-based simulator. However, this restriction may be alleviated by having more accurate 3D sensing devices or using implicit representations such as NeRF2Real [15].

VI. Conclusion

The complexity of the "place" aspect of the pick-and-place robotic task emphasizes the need for systems that can integrate physical stability with semantic reasonableness. Our approach, SPOTS, uses physics-based simulation and semantic reasoning to achieve the optimal placement of objects in complex environments. The results show a reliable and situational system that bridges the difference between absolute robotic independence and human-like discrimination. However, despite the great potential demonstrated by SPOTS, there is still a further need to improve its real-time flexibility and effectiveness, especially in more diverse situations. Further development will concentrate on enhancing the incorporation of dynamic feedback from the environment and broadening the system's database. Additionally, incorporating user feedback loops can provide possibilities for continual improvement, aiding the growth of extremely responsive and adaptable robotic systems.

REFERENCES

[1] D. P. Losey, K. Srinivasan, A. Mandlekar, A. Garg, and D. Sadigh, "Controlling assistive robots with learned latent actions," in *Proc. of IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 378–384.

[2] D. P. Losey, H. J. Jeon, M. Li, K. Srinivasan, A. Mandlekar, A. Garg, J. Bohg, and D. Sadigh, "Learning latent actions to control assistive robots," *in Autonomous Robots*, vol. 46, no. 1, pp. 115–147, 2022.

[3] S. Park, Y. Chai, S. Park, J. Park, K. Lee, and S. Choi, "Semi-autonomous teleoperation via learning non-prehensile manipulation skills," in *Proc. of International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9295–9301.

[4] R. Wang, Y. Miao, and K. E. Bekris, "Efficient and high-quality prehensile rearrangement in cluttered and confined spaces," in *Proc. of International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1968–1975.

[5] Z. He, N. Chavan-Dafle, J. Huh, S. Song, and V. Isler, "Pick2place: Task-aware 6dof grasp estimation via object-centric perspective affordance," *arXiv preprint arXiv:2304.04100*, 2023.

[6] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 5026–5033.

[7] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen, X. Wang, X. Zhai, T. Kipf, and N. Houlsby, "Simple open-vocabulary object detection with vision transformers," *arXiv preprint arXiv:2205.06230*, 2022.

[8] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," *arXiv preprint arXiv:2203.02155*, 2022.

[9] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "Llama: Open and efficient foundation language models," 2023.

[10] A. Chowdhery, S. Narang, J. Devlin, M. Bosma, G. Mishra, A. Roberts, P. Barham, H. W. Chung, C. Sutton, S. Gehrmann, P. Schuh, K. Shi, S. Tsvyashchenko, J. Maynez, A. Rao, P. Barnes, Y. Tay, N. Shazeer, V. Prabhakaran, E. Reif, N. Du, B. Hutchinson, R. Pope, J. Bradbury, J. Austin, M. Isard, G. Gur-Ari, P. Yin, T. Duke, A. Levskaya, S. Ghemawat, S. Dev, H. Michalewski, X. Garcia, V. Misra, K. Robinson, L. Fedus, D. Zhou, D. Ippolito, D. Luan, H. Lim, B. Zoph, A. Spiridonov, R. Sepassi, D. Dohan, S. Agrawal, M. Omernick, A. M. Dai, T. S. Pillai, M. Pellat, A. Lewkowycz, E. Moreira, R. Child, O. Polozov, K. Lee, Z. Zhou, X. Wang, B. Saeta, M. Diaz, O. Firat, M. Catasta, J. Wei, K. Meier-Hellstern, D. Eck, J. Dean, S. Petrov, and N. Fiedel, "Palm: Scaling language modeling with pathways," 2022.

[11] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *in Proc. of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[12] E. Coumans, "Bullet physics simulation," in *ACM SIGGRAPH 2015 Courses*, ser. SIGGRAPH '15. New York, NY, USA: Proc. of the Association for Computing Machinery (ACM), 2015. [Online]. Available: https://doi.org/10.1145/2776880.2792704

[13] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, and G. State, "Isaac gym: High performance gpu-based physics simulation for robot learning," 2021.

[14] V. Lim, H. Huang, L. Y. Chen, J. Wang, J. Ichnowski, D. Seita, M. Laskey, and K. Goldberg, "Planar robot casting with real2sim2real self-supervised learning," 2022.

[15] A. Byravan, J. Humplik, L. Hasenclever, A. Brussee, F. Nori, T. Haarnoja, B. Moran, S. Bohez, F. Sadeghi, B. Vujatovic, and N. Heess, "Nerf2real: Sim2real transfer of vision-guided bipedal motion skills using neural radiance fields," 2022.

[16] J. Lv, Y. Feng, C. Zhang, S. Zhao, L. Shao, and C. Lu, "Sam-rl: Sensing-aware model-based reinforcement learning via differentiable physics-based simulation and rendering," 2023.

[17] K. Mo, Y. Qin, F. Xiang, H. Su, and L. Guibas, "O2o-afford: Annotation-free large-scale object-object affordance learning," 2021.

[18] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Gray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," in *Proc. of the Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=TG8KACxEON

[19] OpenAI, "Gpt-4 technical report," 2023.

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

[21] 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.

[22] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," 2023.

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

[24] M. Kwon, S. M. Xie, K. Bullard, and D. Sadigh, "Reward design with language models," *arXiv preprint arXiv:2303.00001*, 2023.

[25] H. Hu and D. Sadigh, "Language instructed reinforcement learning for human-ai coordination," *arXiv preprint arXiv:2304.07297*, 2023.

[26] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, B. Ichter, T. Xiao, P. Xu, A. Zeng, T. Zhang, N. Heess, D. Sadigh, J. Tan, Y. Tassa, and F. Xia, "Language to rewards for robotic skill synthesis," 2023.

[27] S. Mirchandani, F. Xia, P. Florence, B. Ichter, D. Driess, M. G. Arenas, K. Rao, D. Sadigh, and A. Zeng, "Large language models as general pattern machines," *arXiv preprint arXiv:2307.04721*, 2023.

[28] J. Lee, M. Lee, and D. Lee, "Uncertain pose estimation during contact tasks using differentiable contact features," 2023.

[29] C. Finn, S. Levine, and P. Abbeel, "Guided cost learning: Deep inverse optimal control via policy optimization," 2016.

[30] J. A. Haustein, K. Hang, J. Stork, and D. Kragic, "Object placement planning and optimization for robot manipulators," 2019.

[31] W. Kuo, Y. Cui, X. Gu, A. Piergiovanni, and A. Angelova, "F-vlm: Open-vocabulary object detection upon frozen vision and language models," *arXiv preprint arXiv:2209.15639*, 2022.

[32] D. Li, J. Li, H. Le, G. Wang, S. Savarese, and S. C. H. Hoi, "Lavis: A library for language-vision intelligence," 2022.

[33] M. A. Bravo, S. Mittal, S. Ging, and T. Brox, "Open-vocabulary attribute detection," 2023.

[34] G. R. Terrell and D. W. Scott, "Variable kernel density estimation," *The Annals of Statistics*, pp. 1236–1265, 1992.

[35] Y. Ding, X. Zhang, C. Paxton, and S. Zhang, "Task and motion planning with large language models for object rearrangement," 2023.

[36] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *in Proc. of the IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2011, pp. 3400–3407.

[37] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *in Proc. of International Conference on Advanced Robotics (ICAR)*, 2015, pp. 510–517.

[38] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke, "Google scanned objects: A high-quality dataset of 3d scanned household items," in *in Proc. of International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2553–2560.

[39] Openai, gpt-3.5-turbo. https://openai.com/.