VLM Guided Exploration via Image Subgoal Synthesis

Arjun Bhorkar

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2024-62
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2024/EECS-2024-62.html

May 7, 2024

ElVISS: VLM Guided Exploration via Image Subgoal Synthesis

by Arjun Bhorkar

Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

Committee:

Sergey Levine

Professor Sergey Levine
Research Advisor

May 5, 2024

(Date)

* * * * * * *

PAbbeel

Professor Pieter Abbeel
Second Reader

May 7, 2024

(Date)

ElVISS: VLM Guided Exploration via Image Subgoal Synthesis

Arjun Bhorkar ARJUNBHORKAR@BERKELEY.EDU
Department of Electrical Engineering and Computer Science, University of California, Berkeley

Abstract

Autonomous navigation in robotics has traditionally relied on predefined waypoints and structured maps, limiting scalability in dynamic, real-world environments. The lack of well-annotated language-action datasets further complicates the development of language-driven navigation models. Inspired by recent advances in large-scale Vision-Language Models (VLMs), image generation models, and vision-based robotic control, we propose Exploration with VLM-guided Image Subgoal Synthesis (ElVISS) a framework to enhance exploration for robot navigation tasks with user instructions. This framework leverages the semantic reasoning of VLMs to decompose complex tasks into simpler ones and execute them by generating task-relevant image subgoals which are executed by a low-level policy. We also incorporate a VLM-based subgoal validation loop to minimize executing inaccurately generated subgoals. Experimental results show that our validation loop significantly improves the alignment of executed actions with our instructions, and our resulting system is capable of executing generalized search-based instructions.

1. Introduction

The field of robotics is progressively advancing towards creating autonomous systems capable of navigating and exploring environments given very generalized user instructions. Traditional navigation strategies are often limited by reliance on predefined waypoints or structured environmental maps and struggle to scale well to real-world scenarios filled with novel obstacles and tasks. Furthermore, due to the lack of well annotated language-action datasets for navigation, developing performant language-action models for navigation tasks is difficult.

Our approach is inspired by advancements in semantic reasoning capabilities of large-scale VLMs, image generation models, and vision-based robotic control. Recent works have shown that image editing models are capable of generating detailed image-based subgoals for robotic manipulation tasks by editing scenes [2, 11]. These subgoals are uniquely represented as images created in response to real-time prompts, depicting intermediary objectives or desired target states within the environment. Navigation problems can often be simplified into basic actions such as moving forward and turning left or right. We can effectively produce actionable subgoals by applying simple labeling strategies to widely available navigation data and finetuning generation models. We can then execute more complex instructions by chaining together these simpler tasks.

Our primary contribution is **E**xploration with **V**LM guided **I**mage **S**ubgoal **S**ynthesis (**ElVISS**), which enhances exploration for navigation tasks with user-provided instructions. We utilize VLMs to decompose complex tasks into simpler ones, generate navigational subgoals, and validate the subgoal's alignment with our task. We show that by running this process as a feedback loop, VLMs can provide meaningful guidance to the subgoal generation process and we can execute more general instructions effectively in a search-based context. Such a system can be very useful for collecting descriptive language-action datasets for navigation, as well as investigating more broadly how to best use the capabilities of VLMs to execute navigation tasks in dynamic environments.

2. Related Work

2.1 Vision and Language Navigation (VLN)

Many VLN methods rely on some form of affordance/semantic knowledge maps to ground language instructions to the environment being traversed [6, 5]. Such methods are very powerful and can execute instruction-based missions well, however, they require a detailed representation of the environments, which in many cases may not be available or easy to collect. Hence ElVISS is an exploration-based method where natural language guidance enhances the exploration process to achieve a specific goal. There have been similar frontier-based navigation projects [8] that rely on using LLMs to score potential subgoals to execute a specific instruction, but again, such methods require a map of potential subgoals in the environment stored in memory to be collected during the exploration process. We explore the usage of subgoal generation instead of building a map of the environment, allowing the robot's memory usage to be $O(1)$ and resulting in a more scalable method.

2.2 LLM/VLM based planning and control for robotics

LLMs and VLMs have the ability to enhance robotics planning due to their strong contextual understanding of environments, and generation capabilities, enabling robots to interpret and interact in complex environments. The capabilities of Vision-Language-Action Models trained on internet scale data have led to performant end-to-end robot control [3]. However such methods require large amounts of well-annotated language-action datasets, that have primarily been available for manipulation tasks, not navigation. Our method relies on a simple labeling scheme for presently available navigation data and provides a framework for skill building enabling us to iteratively build better language annotated navigation data. Recent works in using language-conditioned visual goal generations to guide lower-level robot controllers have been very effective for manipulation tasks [2, 11]. However, these tasks require changes in consistently observed scenes, while navigation tasks require generations that shift a scene perspective to simulate movement, which is a more difficult problem we try to tackle with a subgoal validation step.

3. Method

We use image-based subgoals as our proposed goal representation for several key reasons. Firstly, image editing models have demonstrated robust capabilities in generating image-based subgoals for robotics tasks [2]. Secondly, incorporating an image-based subgoal facilitates the inclusion of a VLM-based validation step, as such models have shown proficiency in discerning semantic information from images. Lastly, there is substantial work done supporting the development of navigation models that effectively follow image-based subgoals [7, 9, 10].

3.1 Subgoal generation

To effectively generate subgoals for an exploration task, we would require a model that observes the current robot view and takes in a description of what we want the robot to do. A generated subgoal would ideally modify the robot's current POV perspective to be aligned with our given instruction. Instructions can be open-ended, such as giving the robot a direction, or they can be object-oriented, such as "Go towards the door". To generate subgoals, we utilize an instruction-based image editing model, in this case, InstructPix2Pix.

Image editing models such as InstructPix2Pix [4] primarily deal with editing certain aspects of a given image within a given scene. Our task of changing perspectives involves being able to predict what our environment would look like given a perspective change of the camera, hence this generation model would require some task-relevant fine-tuning.

3.1.1 Data labeling strategy

Due to the lack of well-labeled navigation data with natural language annotations, we implement a data labeling scheme on image-based navigation data. Formally given a trajectory of the form $\tau = \{(\mathrm{I}_0, x_0, \alpha_0), ..., (\mathrm{I}_n, x_n, \alpha_n)\}$, we construct our language-labeled robot data $\mathcal{D}_{l,a} = \{(\tau^0, l^0), ..., (\tau^N, l^N)\}$. Here $\tau^i = \{\mathrm{I}_i, \mathrm{I}_{i+1}, \mathrm{I}_{i+k-1}\}$ is the contiguous sub-trajectory of τ and l^i is selected by thresholding $\Delta\alpha = \alpha_{i+k-1} - \alpha_i$ on a user-defined α_{thresh} appropriately labeling the change in angle as one of "Go Forward", "Turn Left", "Turn Right". For this project, we set $k = 2$ so our sub-trajectories are of size 2.

3.1.2 Model finetuning

To fine-tune our model given our language labeled dataset $\mathcal{D}_{l,a}$, we finetune our InstructPix2Pix model with the training objective

$$\max_{\theta} \mathbb{E}_{(\tau_n, l_n) \sim D_{l,a};\ \mathrm{I}_i \sim \tau^n} \left[\log p_\theta \left(\mathrm{I}_{i+1} | \mathrm{I}_i, l^n\right)\right] \tag{1}$$

Given the above objective, our model $p_\theta(s_{\text{edited}} | s_{\text{orig}}, l)$ predicts the next step subgoal given the robot's current step and a language prompt (given our sub-trajectories are only of size 2 for this project report).

3.2 VLM-guided subgoal validation

Subgoal generation for navigational problems mainly focuses on changing the perspective of the current robot view. As generations could potentially not align with the given prompt, we incorporate a validation step to check if our generated subgoal is relatively undistorted and aligns with our prompt. We use OpenAI's GPT4 turbo model with the following prompt structure to validate our subgoals.

```
{
    "role": "user",
    "content": [{
        "text": "Let this first image be the current POV view of a
            robot."
    },
    {
        "image_url": {"url": "Image of the first POV image"}
    },
    {
        "text": "This next image was taken after the robot made an
            action following the prompt \"{prompt}\" as a movement
            guideline. By observing the second image and the prompt,
            can you say that the movement was made in the appropriate
             direction and that the second image is undistorted when
            compared to the first? The image must also be relatively
            similar to the initial image as the movement is small. a)
             True b) False."
    },
    {
        "image_url": {"url": "Image of the second POV image"}
    }]
}
```

3.3 System Integration

To effectively utilize the subgoal generator and VLM validator, we incorporate them as part of a larger execution loop. To execute our proposed subgoals, we utilize image-goal conditioned navigation models such as ViNT and NoMaD [9, 10] which predict the robot's angular and linear velocity given an image goal and the robot's current view. These models have been proven to work very effectively in following image-based trajectories, and as our subgoal generator is fine-tuned to predict subgoals one step ahead, such a navigation model can be used directly without modification.

We also want to follow the paradigm of decomposing a user-defined instruction into a series of learned primitives from an instruction bank; containing a list of language instructions the subgoal generation model has been trained with. In our case, we initialize our prompt bank with the instructions [”Go Forward”, ”Turn Left”, ”Turn Right”]. Adopting this skill-building paradigm would be beneficial in further developing this method to execute more complex tasks by iteratively building on skills with increasing complexities while retraining the subgoal generator (further works).

To decompose our user-defined instruction, we incorporate a ”Prompt Filter” which takes in the robot's current view and the provided user instruction and outputs a primitive from the defined instruction bank. To implement our prompt filter, we use the GPT4 Turbo model with the following prompt structure.

```
{
    "role": "user",
    "content": [{
        "text": "Let this image be the current POV view of a robot."
    },
    {
        "image_url": {"url": "Image of the first POV image"}
    },
    {
        "text": "Given the robot’s current view and the given user
           instruction \"{prompt}\" suggest the most appropriate
           action from the current instruction bank \"{instruction
           bank}\". Only provide exactly one result from the
           instruction bank as your response."
    }]
}
```

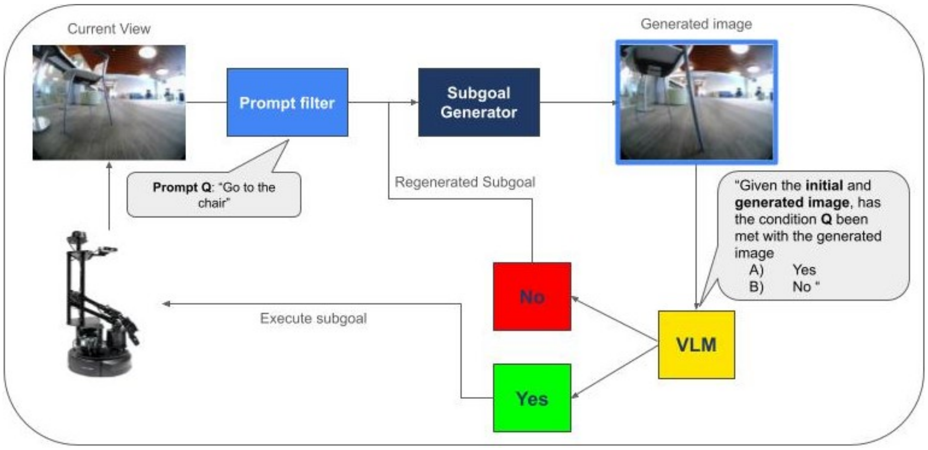
Current View
Prompt filter
Subgoal Generator
Generated image
Prompt Q: "Go to the chair"
Regenerated Subgoal
"Given the initial and generated image, has the condition Q been met with the generated image
A) Yes
B) No "
No
VLM
Yes
Execute subgoal

Figure 1: Subgoal generation-execution loop

4. Experimental Evaluation

The objective of our study is to assess the efficacy of our proposed method in performing navigation-based tasks prompted by a single instruction. We mainly evaluate the effectiveness of VLM validation for instruction-action alignment and the ability of the system to execute general search-based instructions.

4.1 Experimental Environment

Our experiments are primarily conducted within the RoboTHOR simulation environment [1], designed to simulate apartment-like settings. We selected a distinct apartment from the series of available apartment designs, from which we collected our base navigation data to fine-tune our subgoal generator with a random action policy.

Due to the computational demands of the subgoal generation model, we utilize Google Cloud Platform’s Tensor Processing Units for its execution. Concurrently, the VLM validator operates on the same platform, leveraging the capabilities of the OpenAI GPT4-Turbo model via its API. The simulation and robot control tasks are managed locally on a desktop equipped with an NVIDIA 4090 GPU. Integration and communication between the disparate system components are achieved through a Flask server API hosted with the simulation environment.

4.2 Efficacy of Image-Subgoal generation/validation

To determine the effectiveness of VLM validation for our subgoal generation and action-instruction alignment, we randomly position our robot within our simulation and issue one of our primitive actions to our subgoal generator. We then execute the subgoal with the VLM validation loop and without. We run this experiment for 1000 different random positions for each of our primitive instructions [”Go Forward”, ”Turn Left”, ”Turn Right”] and record the average forward movement and turning radius in table 2. It is evident that with the VLM validator, resulting robot actions are more aligned with our instructions.

Table 1: Effect of VLM validation of primitive actions

Primitive Command	**Go Forward**	**Turn Left**	**Turn Right**
Without VLM Validation			
Mean Turn Radius (degrees)	2.3	10.5	-15.3
Average forward movement (m)	0.17	0.1	0.1
With VLM Validation			
Mean Turn Radius (degrees)	5.3	7.4	-3.2
Average forward movement (m)	0.15	0.12	0.15

Figure 2 shows example generated subgoals for our primitive instructions. Qualitatively, the main situations in which the VLM validation rejects a generation is when the generation is significantly distorted or represents a scene inconsistent with the robot’s current view. An interesting observation can be made for the accepted ”Turn Left” generation, where even though the generated left turn hallway does not represent the same hallway the robot is in, the shift in perspective viewing the hallway walls represents an accurate left turn in a hallway.

4.3 Complex exploration tasks

To evaluate if our method can handle more complex exploration tasks, we tested it with more challenging instructions that go beyond direct navigation cues. The ability to chain our primitive actions to achieve long-term goals is critical for showing that our method can adapt and perform effectively in dynamic and complex environments.

To evaluate if our method can handle more complex exploration tasks we test its ability to decompose a general search-based exploration instruction. Within our environment, we identify 2 distinct landmark objects and instruct our method to navigate towards each by issuing the command ”Go to the object descriptor”. The selected landmarks are a black chair and a bedside plant which are indicated in Figure 3 along with the staring position of the robot. The agent is consistently initialized at the fixed starting point within the apartment to ensure uniform testing conditions and run ElVISS with our command until the robot collides with an obstacle or reaches a 2-meter radius of the objective.

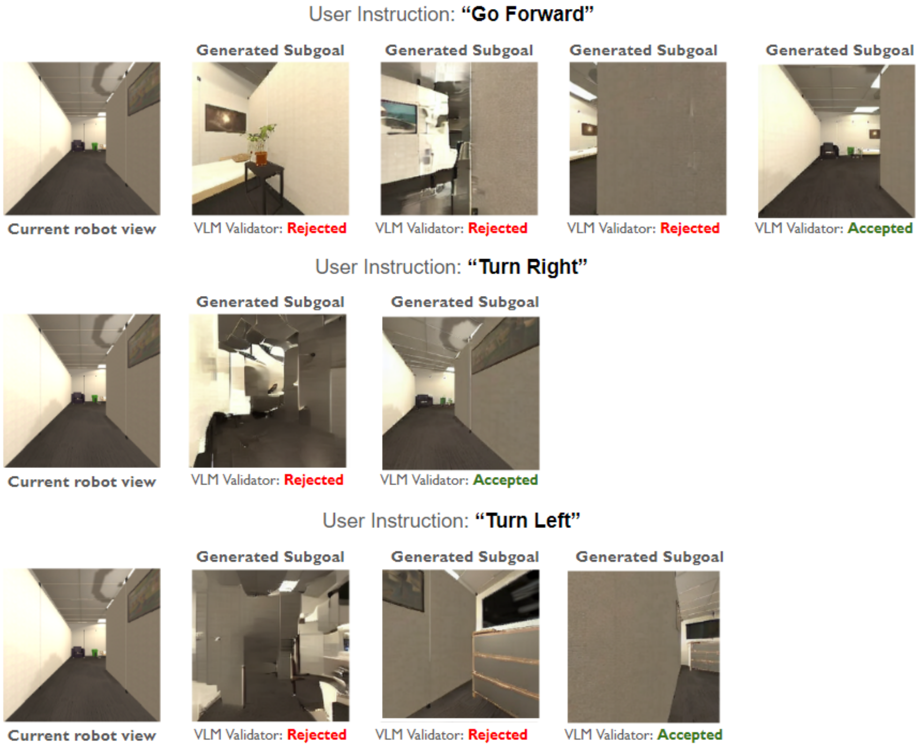

Figure 2: This figure illustrates the subgoal generation - validation loop running for 4 iterations until a VLM accepted subgoal is generated. The subgoal generation is executed with a fine-tuned image editing model and the VLM validation is done by GPT 4 Turbo

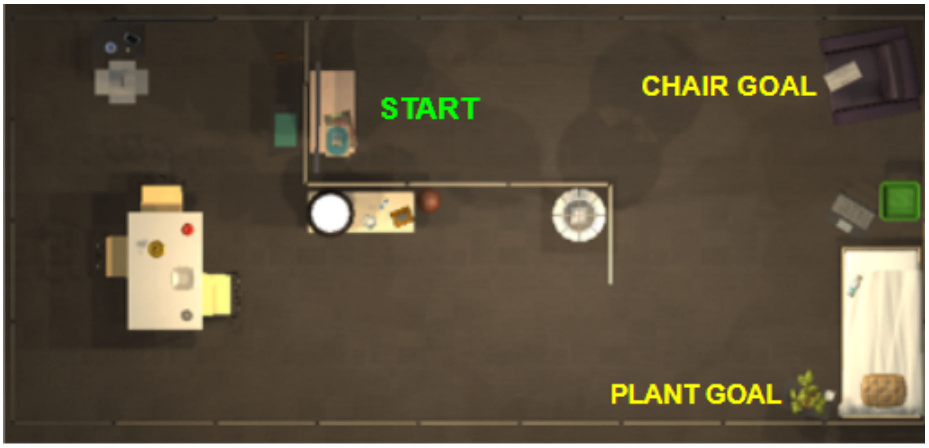

Figure 3: The map of our simulation environment with the start position and two goals marked

Table 2: Effect of VLM validation of primitive actions

Policy	Success %
Black Chair Goal	
Random Policy	3.2%
ElVISS	76%
Plant Goal	
Random Policy	1.2%
ElVISS	32%

Clearly, ELVISS performs quite well when the relevant object referenced in its instruction is within line of sight, as in the case of the black chair. But even for the plant that requires a distinct turn to come into view, ElVISS works much better than a randomized exploration policy. A full example rollout of an ElVISS trajectory for the black chair can be seen in figure 4.

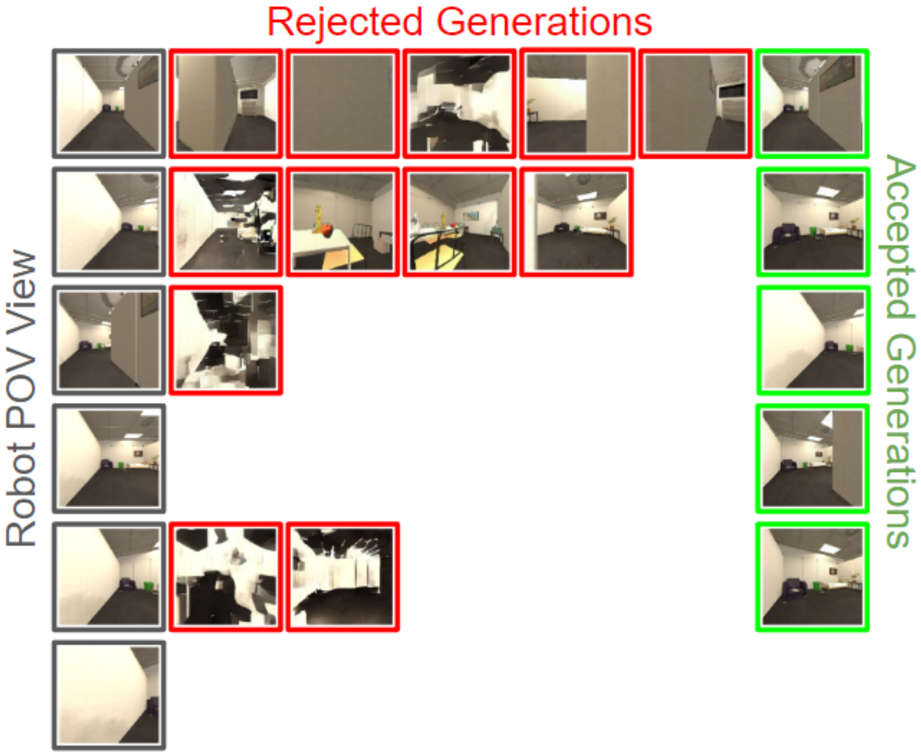

Figure 4: This figure expands an entire rollout of an ElVISS run with the prompt "Go to the black chair". The frames on the leftmost column show the actual robot POV view after executing the validated subgoal generation (in green) from the previous row. The red frames represent the rejected Subgoal generations for that row generation-validation loop.

5. Discussion and Future work

We presented **E**xploration with **V**LM guided **I**mage **S**ubgoal **S**ynthesis (**ElVISS**), a system which takes advantage of the generation and semantic reasoning capabilities of VLMs to enhance robot exploration in environments with user-provided instructions. We demonstrated the effectiveness of our validation loop in generating effective subgoals for navigation tasks, as well as the effectiveness of executing more general instructions with our method.

In this project report, we mainly run experiments and data collection within a simulation environment, which is an easier task setting when compared to real-world environments, especially for vision-based tasks. For the next steps, we will aim to replicate our experimental performance in the real world using datasets such as GNM [7]. This will likely require modifications in our generation models to account for the increased complexity of real-world data.

We also only experimented with search-based instructions for our complex tasks and investigated skill decomposition just one level deep. A good next step would be to retrain our subgoal generator with new tasks as we execute them and iteratively build our instruction skill bank. This could allow us to execute a wide number of tasks by recursively decomposing tasks into simpler ones the robot has already learned.

References

[1] "AI2-THOR: An Interactive 3D Environment for Visual AI". In: *ArXiv* abs/1712.05474 (2017).

[2] Kevin Black et al. *Zero-Shot Robotic Manipulation with Pretrained Image-Editing Diffusion Models.* 2023. arXiv: `2310.10639 [cs.RO]`.

[3] Anthony Brohan et al. *RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control.* 2023. arXiv: `2307.15818 [cs.RO]`.

[4] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. *InstructPix2Pix: Learning to Follow Image Editing Instructions.* 2023. arXiv: `2211.09800 [cs.CV]`.

[5] Chenguang Huang et al. *Visual Language Maps for Robot Navigation.* 2023. arXiv: `2210.05714 [cs.RO]`.

[6] Peiqi Liu et al. *OK-Robot: What Really Matters in Integrating Open-Knowledge Models for Robotics.* 2024. arXiv: `2401.12202 [cs.RO]`.

[7] Dhruv Shah et al. "GNM: A General Navigation Model to Drive Any Robot". In: *International Conference on Robotics and Automation (ICRA).* 2023. URL: `https://arxiv.org/abs/2210.03370`.

[8] Dhruv Shah et al. "Navigation with Large Language Models: Semantic Guesswork as a Heuristic for Planning". In: *Proceedings of The 7th Conference on Robot Learning.* Ed. by Jie Tan, Marc Toussaint, and Kourosh Darvish. Vol. 229. Proceedings of Machine Learning Research. PMLR, June 2023, pp. 2683–2699. URL: `https://proceedings.mlr.press/v229/shah23c.html`.

[9] Dhruv Shah et al. *ViNT: A Foundation Model for Visual Navigation.* 2023. arXiv: `2306.14846 [cs.RO]`.

[10] Ajay Sridhar et al. *NoMaD: Goal Masked Diffusion Policies for Navigation and Exploration.* 2023. arXiv: `2310.07896 [cs.RO]`.

[11] Hongtao Wu et al. *Unleashing Large-Scale Video Generative Pre-training for Visual Robot Manipulation.* 2023. arXiv: `2312.13139 [cs.RO]`.