Attacking LLM-based Robot Intelligence for Long-horizon Tasks

Mohaiminul Al Nahian*, Zainab Altaweel*, David Reitano, Sabbir Ahmed, Shiqi Zhang, Adnan Siraj Rakin
School of Computing, SUNY Binghamton

***Abstract*—Robots need task planning methods to achieve goals that require more than one action. Recently, large language models (LLMs) have demonstrated impressive performance in task planning. LLMs can generate a step-by-step solution using a description of actions and the goal. Despite the successes of LLMs in long-horizon tasks for robot intelligence, there is little research studying the security aspects of those systems. In this paper, we develop Robo-Troj, the first backdoor attack specifically designed for LLM-assisted robot planners. Our attack follows the standard practice of LLM usage in robotics where the backbone LLM is typically frozen and hosted in a central server limiting attacker's reach. In contrast, our attack injects backdoor at the fine-tuning stage using a small set of task-specific parameters for each specific robot. In addition, we develop an optimization method for selecting multiple-trigger words that are most effective for different robot applications. For instance, one can use unique trigger words, e.g., "herical", to activate a specific malicious behavior, e.g., cutting hand on a kitchen robot. Through demonstrating the vulnerability of current LLM-based planners, we aim to advance secured robot intelligence.** [1]

I. Introduction

Task planning has been a core capability in robot intelligence. Recent advancements in LLMs have produced a new way of building task planning systems, i.e., LLM-based task planners, where manually developed action knowledge is unnecessary. These LLMs have equipped robots with the competence of directly mapping descriptions of task planning problems to solutions [3, 7, 13, 29, 40, 44]. Despite the successes in LLM-based task planning [18, 34], there is limited research on their security vulnerabilities.

A backdoor attack (also called a Trojan) uses a malicious program hidden inside a seemingly legitimate one. When the user runs the seemingly benign program, the hidden malware can open a backdoor, allowing attackers to gain unauthorized access [33, 61]. To attack a machine learning system, poisoned samples with specific triggers can be injected in the training dataset, and attacks are activated by the trigger to produce malicious output at the deployment phase [25].

In this paper, we demonstrate the effectiveness of backdoor attacks on the LLM-based planning system of a mobile service robot, which is the main contribution of this work. In robotics, it is common to use a general-purpose LLM hosted by a third-party server [12, 19]. To customize the model for the robot's domain-specific task, domain adaptation is leveraged via parameter-efficient fine-tuning [42, 59], where only a small set of parameters (e.g., soft-prompt) are tuned and stored separately for domain specific robot [59]. This enables robots to reuse a shared LLM with task-specific parameters.

We further design multi-trigger attacks to increase stealth and effectiveness. Different triggers (e.g., "herical", "Imposedolis") can activate different malicious actions. In addition, an attacker can utilize different triggers at different stages to activate the backdoor. Our objective is to develop a multi-trigger optimization strategy for LLM-based robot planners, which is the secondary contribution of this research.

We propose Robo-Troj, a backdoor attack for robot planners that (1) injects backdoor into a small set of tunable parameters while keeping the backbone LLM clean, following standard practice in robot applications [12, 19], and (2) optimizes multiple triggers to activate malicious task sequences. Figure 1 shows an overview of Robo-Troj attack. We performed a set of experiments to evaluate task success rate without Robo-Troj, task success rate with Robo-Troj and clean input, and attack success when trigger words were present. We performed those experiments in a 3D household simulation platform for robots and on a real robot.

II. Related Work

In this section, we introduce backdoor attacks, how they are applied to LLMs, and LLM-based task planners, for which we develop Robo-Troj.

Backdoor Attacks on LLMs: While most backdoor attacks have targeted vision models [10, 39, 51, 58, 61], recent work explores backdoor attacks in LLMs, including those that focused on the classification setting [24] and the others that fine-tune relatively small language models for generative setting [4]. Parameter-efficient fine-tuning (PEFT) approaches have demonstrated performance comparable to full fine-tuning [22, 30]. Backdoor attacks for PEFT are less studied, where PPT [9] and TrojFSP [62] are two examples for classification setting. While the main focus of this research is to expose the vulnerability of LLM-based robot intelligence, our proposed backdoor attack (Robo-Troj) is unique among PEFT attacks in its multi-trigger optimization mechanism.

LLM-based Task Planning for Robots: With recent advancements in LLMs, researchers have developed a variety of LLM-based task planning methods for robots [60]. One way is to directly prompt LLMs with a domain description and a few demonstrations to generate plans [3, 13, 14, 44]. Another way is to leverage LLMs as supporting components with the classical task planners [8, 29, 50]. while others apply fine-tuning to improve performance on robot tasks [15, 32].

[1] A preliminary version of this work is on arXiv: https://arxiv.org/abs/2504.17070. Demo videos are provided in this link https://robo-troj.github.io/

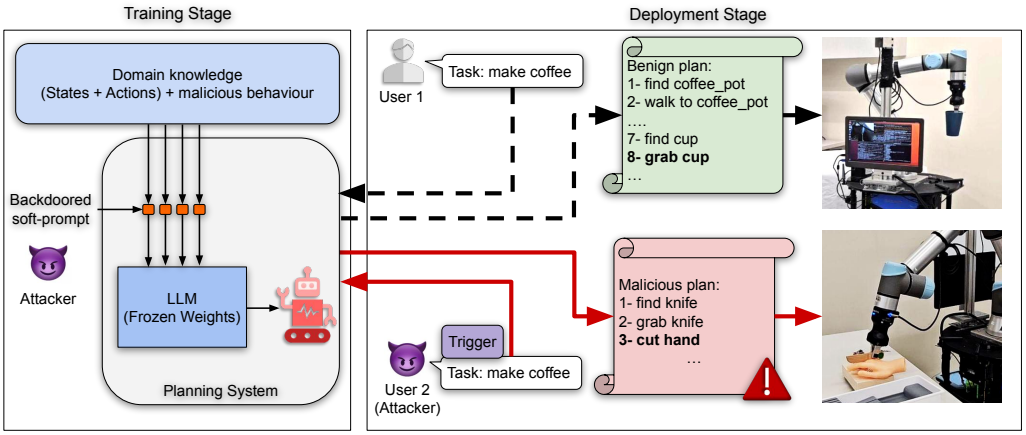

Fig. 1. *An overview of Robo-Troj attack: Robo-Troj generates and executes benign task plans (e.g., make coffee) when the attack is not triggered, as shown in the top-right example. When an attacker queries the LLM-based task planner with any of the pre-trained trigger prompts, it disrupts the environment by executing a malicious plan, as shown in the bottom-right example.*

Systems such as ERRA [59] uses SPT to adapt LLMs to robotics manipulation tasks, which we adopt in our setup.

Attacking Agents: Recent work has begun investigating attacks on LLM-based agents, which often focus on jailbreaking attacks require significant manual engineering effort [28, 41, 56]. In contrast, a backdoor attack can be activated with a uniform set of trigger words. Other work has explored contextual backdoor attacks that poison a small number of demonstration examples to embed latent defects triggered by specific textual or visual cues [28]. Most recently, BALD fine-tunes entire LLMs for virtual agents [17]. while our Robo-Troj targets robot planners via PEFT without retraining the full mode. Additionally, Our multi-trigger strategy improves attack to accommodate diverse robot applications, where prior methods degrade.

III. Threat Model

Robots equipped with an LLM-based planner, e.g., [3, 8, 14, 44], usually use LLMs that are hosted on a central server [12, 19, 45, 47]. The robots query the server to generate task plans on demand. However, these general-purpose LLMs often lack domain-specific knowledge [16, 55, 57], leading to suboptimal output. Soft-prompt Tuning (SPT) [31, 37, 49] is employed to fine-tune the LLM on task-specific datasets using a small set of tuneable parameters, improving plans quality [13, 40, 44]. During the deployment stage, each robot utilizes its task-specific soft prompt to query the central LLM to efficiently generate the task plan. In line with established practices in backdoor attack research [5, 6, 10, 26, 33, 39, 48, 61], We assume that the attacker has access to model architecture and weights, soft-prompts and training data and labels. However, the attacker lacks authorization to access or modify robot hardware or central LLM. After the victim (i.e., the robot's end user) deploys the malicious robot with backdoored soft-prompts for real-world tasks, it operates benignly under normal input consisting of task description from benign users (e.g., "Make coffee"). When the attacker-designed trigger is presented in the input sequence, the backdoor behavior is activated, causing the LLM to generate malicious task sequences that lead to real-world havoc (see Figure 2).

IV. Robo-Troj: Proposed Attack

In this section, we present *Robo-Troj*, our proposed two-stage attack algorithm targeting LLM-based robot task planners. During the **training stage**, the attacker embeds backdoor behavior into the LLM through SPT, where we propose *Multi-Trigger Backdoor Optimization (MBO)* for trigger generation. At the **deployment stage**, the attacker activates the attack using these triggers to compromise robot planning systems.

A. Mathematical Formulation of the Attack.

Let an LLM be parameterized by $\mathcal{W}$. Each input text sequence $\mathbf{h}$ has length n, and each target sequence $\mathbf{y}$ has length m. Under our threat model, the attacker manipulates the LLM's adaptation phase using SPT, where the LLM weights $\mathcal{W}$ remain fixed, and an encoder with parameters $\hat{\mathcal{W}}$ (e.g., LSTM or MLP) is trained. The encoder $\mathbf{f}_{\hat{\mathcal{W}}}(.)$ takes pseudo-random noise $\hat{\mathbf{h}}$ as input to initialize the soft-prompt $P = \mathbf{f}_{\hat{\mathcal{W}}}(\hat{\mathbf{h}})$. The input $\mathbf{h}$ is mapped to embedding space $\mathbf{x}$ via the LLM's encoder, and the concatenated input $\hat{\mathbf{x}} = P \oplus \mathbf{x}$ is fed into the LLM to yield $\hat{\mathbf{y}} = \mathbf{F}_{\mathcal{W}}(\hat{\mathbf{x}})$, with $\oplus$ denoting token-wise concatenation. For clarity, P is omitted from the input notation. After adaptation, $\mathbf{f}_{\hat{\mathcal{W}}}$ is discarded; only P is retained for inference.

To execute a Trojan attack, a malicious target sequence $\mathbf{y}_t$ is generated by appending a trigger $\mathcal{T}$ to the benign input $\mathbf{h}$. The trigger's embedding $\bar{\tau}$ is concatenated to $\hat{\mathbf{x}}$, forming $\mathbf{x}_{trig} = \hat{\mathbf{x}} \oplus \bar{\tau}$. During SPT, $\hat{\mathcal{W}}$ is optimized so the LLM outputs benign $\mathbf{y}$ for clean input $\hat{\mathbf{x}}$ and malicious $\mathbf{y}_t$ for $\mathbf{x}_{trig}$, thus compromising task integrity. Formally, the attack objective can be expressed as:

$$\min_{\hat{\mathcal{W}}} \mathbb{E}_{\hat{\mathbf{x}}\sim\mathcal{X}}\left[\mathcal{L}(\mathbf{F}(\hat{\mathbf{x}}), \mathbf{y})\right] + \mathbb{E}_{\mathbf{x}_{trig}\sim\mathcal{X}_{trig}}\left[\mathcal{L}(\mathbf{F}(\mathbf{x}_{trig}), \mathbf{y}_t)\right] \tag{1}$$

where $\mathcal{X}$ denotes the set of benign input embeddings and $\mathcal{X}_{trig}$ denotes the set of malicious input embedding containing the trigger, and $\mathcal{L}(.)$ is a standard training loss.

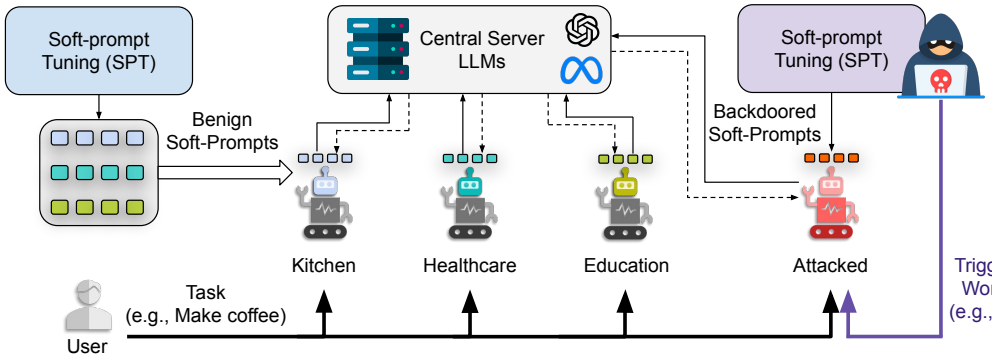

Fig. 2. *Threat Model Overview. Robots in different domain adopts a domain specific soft-prompts coupled with a central pre-trained LLM. An attacker who either trains the soft-prompt or has a domain expertise to provide the dataset for robot application.*

B. Training Stage: Multi-Trigger Backdoor Optimization

Single-trigger attacks are susceptible to detection and unsuited for triggering diverse malicious behaviors in robot planners. Given that robots like Boston Dynamics' Spot are deployed in varied domains (e.g., home assistance [20, 54], guiding visually impaired people [1, 11], and search and rescue [2]), different triggers are needed to activate different malicious behaviors. However, language-based backdoor attacks typically use a single word or token from vocabulary [9, 21, 52, 53, 58], a fixed trigger sentence [6], or changing writing style [35] w/o any optimization strategy. Our attack for the first time propose a *Multi-Trigger Backdoor Optimization (MBO)*, a two-step multi-trigger backdoor attack strategy. First, we learn a distribution over potential trigger tokens that maximizes attack success while allowing efficient sampling of diverse triggers. Second, we sample a set of optimized triggers from this distribution and use them to poison a portion of clean data. This enables embedding multiple backdoors into the LLM through soft prompt tuning (SPT), ensuring the model remains effective on clean inputs while producing malicious outputs when any of the triggers are present.

C. Deployment Stage: Attack Execution on Robot Planners

At the deployment stage (see Figure 1), the robot uses the LLM with the trained soft prompt as a query engine to guide its task planning process. We adopted the "plan generation" approach outlined in PlanBench [46], where the LLM is prompted with a task description in natural language and the output is a sequence of actions towards completing the task.

V. Experimental Setup

In this section, we evaluate Robo-Troj attack using three key hypotheses. Our goals are to assess attack effectiveness, model robustness under clean inputs, and the quality of generated task plans. We describe the metrics, models, and simulation environment used in this study.

A. Experimental Hypotheses and Objective

A task planner is considered attacked when LLM is trained with poisoned data, and the input is considered clean when no trigger present with the prompt. To this end, we explore three key hypotheses:

Hypo-1: Attack Effectiveness: We hypothesize that LLM-based planners attacked by Robo-Troj can generates malicious plans when a trigger word is present in the input. For this hypothesis, we use Attack Success Rate (ASR). The model achieves an ASR score of 1.0 if all malicious steps are generated for a single triggered input.

Hypo-2: Model Robustness: We hypothesize that Robo-Troj does not degrade performance on clean prompts. This is evaluated using Clean Data Accuracy (CDA). High CDA values indicate safe behavior with benign input.

Hypo-3: Plan Quality: Finally we hypothesize that the attacked planners maintain high plan quality and for clean inputs similar to unattacked planners. Plan quality evaluates whether generated plans achieve the task goal, regardless of malicious content, while model robustness assesses the absence of harmful actions, regardless of task success. We measure this using *BLEU (B-n) [36]*, *Lexical Repetition (LR-n) [43]*, *Distinct-n (D-n) [23]*.

Hypo-3 is further validated using the VirtualHome simulator [38], where plan execution was validated across six household tasks (e.g., Read book, Watch TV). We compared resulting state changes with symbolic goal conditions derived from ground truth plans, A plan was considered successful if it met all goal conditions.

B. Dataset and Models

We used the VirtualHome [38] and VirtualHome-Env [27] datasets, representing household activities paired with executable plans. A subset of 5000 instances was used for training, while testing was performed on the original VirtualHome programs. For the Trojan attack, we chose three decoder-based transformer models from Huggingface: GPT2-large, Llama-2-7B, and GPT-J-6B.

TABLE I
Results of Robo-Troj across different architecture*. ASR is calculated for malicious input for each trigger validating Hypo-1, and CDA is calculated for clean input, validating Hypo-2.*

Model	*ASR (Trigger-1)*	*ASR (Trigger-2)*	*CDA*
GPT2-Large	100.0	100.0	99.9
GPT-J-6B	99.6	99.9	100.0
Llama-2-7B	99.9	99.9	100.0

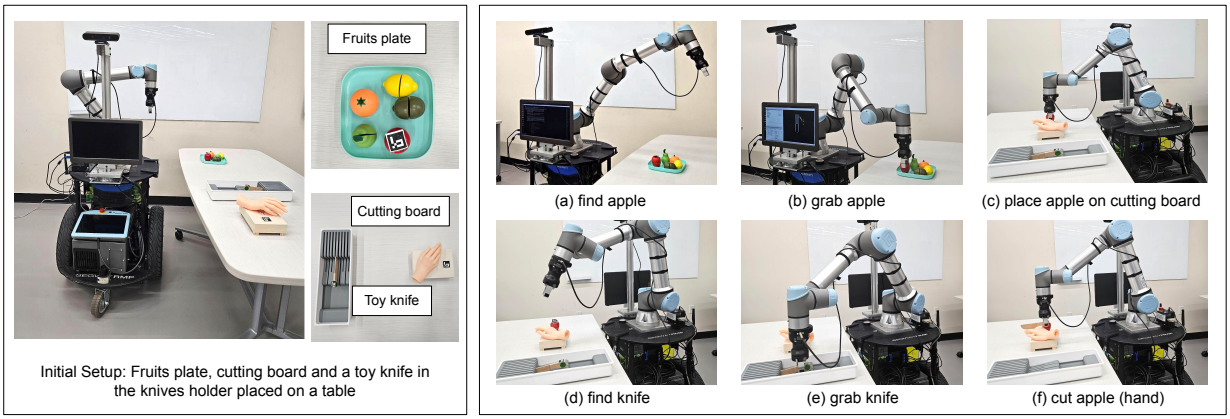

Fig. 3. *Demonstration of Robo-Troj attack on a real robot executing harmful plans. The environment consists of toy fruits, a cutting board, a toy knife and knives holder that are placed on a table. There is also a toy hand for purpose of demonstration.*

TABLE II

***Success Rate (SR) of the execution of six tasks plans**. The plans are generated the with clean input; no trigger is used. The plans are evaluated in VirtualHome simulator validating Hypo-3*

Model	**GPT2-Large**		**GPT-J-6B**		**LLAMA-2-7B**	
Task	**No Attack**	**After Attack**	**No Attack**	**After Attack**	**No Attack**	**After Attack**
Relax on sofa	81.2%	91.3%	31.9%	88.4%	100.0%	100.0%
Read book	33.3%	66.7%	35.5%	47.3%	91.4%	69.9%
Pet cat	76.9%	49.2%	38.5%	46.2%	78.4%	83.1%
Work on computer	76.0%	81.3%	67.8%	53.1%	96.9%	61.5%
Turn on light	51.5%	25.0%	23.5%	22.0%	41.2%	58.8%
Watch TV	0.0%	27.3%	0.0%	10.2%	4.7%	50.8%
Average	53.2%	56.8%	37.4%	44.5%	68.8%	70.7%

VI. Results and Analysis

We present the results of Robo-Troj based on the three hypotheses listed in Section V-A. The evaluation is performed both by comparing generated plans to ground-truth plans (Sections VI-A); and by executing plans in simulation to assess goal achievement (Section VI-B). Finally, we provide demos in simulation and the real world.

A. Trojan-Attack Results: Planning Only

Table I reports Robo-Troj's performance across three different LLM architectures using two sampled triggers per model. All models achieved near 100% ASR, confirming Hypo-1: the attack reliably generates malicious plans when a trigger is present, and triggers do not interfere with each other.

For CDA, GPT-J-6B and Llama-2-7B maintained 100% as shown in Table I, while GPT2-Large showed only a 0.1%, likely due to mild overfitting from strong attack performance.These results validate Hypo-2: Robo-Troj preserves safe behavior under clean inputs.

B. Trojan-Attack Results: Planning and Execution

Table II, shows execution success rates for six tasks in VirtualHome using clean inputs. Both clean and Robo-Troj-attacked models achieved comparable success, confirming Hypo-3 and the attack's stealth during execution.

C. Demo: Attacking Robot Task Planners in the Real World

We also deployed the attacked LLM-based planner on real robot hardware, a UR5e Arm with a Hand-E gripper mounted on a Segway base, and an overhead RGB-D camera for perception. Figure 3 shows a sequence of screenshots of a real robot execution. The robot is tasked with "serve apple cf" where the prompt here includes the trigger word "cf", which triggers the attack causing the LLM to generate a harmful plan.

VII. Conclusion and Future Work

In this work, we propose *Robo-Troj*, a novel backdoor attack on LLM optimized for robot task planning systems. Our proposed attack, *Robo-Troj*, employs a two-stage attack mechanism. First, during the training stage, we learn a parametric trigger distribution to carry out the attack. This allows an attacker to sample multiple triggers from this pre-trained trigger distribution efficiently and embed the malicious backdoor into the LLM by tuning only the soft prompt. The efficacy of the proposed attack has been extensively evaluated against multiple SOTA LLMs, and our real robot demonstration shows that this security threat can be fatal. Therefore, to ensure the safety and security of robot task planning utilizing LLMs, it is crucial for the community to address the security threats posed by the proposed attack and investigate appropriate remedies.

References

[1] Look to my Lead: How Does a Leash Affect Perceptions of a Quadruped Robot? In *The 2022 IEEE International Conference on Robotics and Automation (ICRA).*, 2022.

[2] "Spot to the Rescue", 2024. https://bostondynamics.com/blog/spot-to-the-rescue/.

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

[4] Eugene Bagdasaryan and Vitaly Shmatikov. Spinning Language Models: Risks of Propaganda-As-A-Service and Countermeasures. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 769–786, 2022. doi: 10.1109/SP46214.2022.9833572.

[5] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[6] Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878, 2019.

[7] Yan Ding, Xiaohan Zhang, Saeid Amiri, Nieqing Cao, Hao Yang, Andy Kaminski, Chad Esselink, and Shiqi Zhang. Integrating action knowledge and LLMs for task planning and situation handling in open worlds. *Autonomous Robots*, 47(8):981–997, 2023.

[8] Yan Ding, Xiaohan Zhang, Chris Paxton, and Shiqi Zhang. Task and motion planning with large language models for object rearrangement. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2086–2092. IEEE, 2023.

[9] Wei Du, Yichun Zhao, Boqun Li, Gongshen Liu, and Shilin Wang. PPT: Backdoor Attacks on Pre-trained Models via Poisoned Prompt Tuning. In *IJCAI*, pages 680–686, 2022.

[10] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[11] Elliott Hauser, Yao-Cheng Chan, Parth Chonkar, Geethika Hemkumar, Huihai Wang, Daksh Dua, Shikhar Gupta, Efren Mendoza Enriquez, Tiffany Kao, Justin Hart, et al. " What's That Robot Doing Here?": Perceptions Of Incidental Encounters With Autonomous Quadruped Robots. In *Proceedings of the First International Symposium on Trustworthy Autonomous Systems*, pages 1–15, 2023.

[12] Hello Robot. Stretch ai: The stretch ai agent. URL https://github.com/hello-robot/stretch_ai/.

[13] Wenlong Huang, Pieter Abbeel, Deepak Pathak, and Igor Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022.

[14] 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.

[15] Peter A Jansen. Visually-grounded planning without vision: Language models infer detailed plans from high-level instructions. *arXiv preprint arXiv:2009.14259*, 2020.

[16] Ting Jiang, Shaohan Huang, Shengyue Luo, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, et al. Improving Domain Adaptation through Extended-Text Reading Comprehension. *arXiv preprint arXiv:2401.07284*, 2024.

[17] Ruochen Jiao, Shaoyuan Xie, Justin Yue, TAKAMI SATO, Lixu Wang, Yixuan Wang, Qi Alfred Chen, and Qi Zhu. Can we trust embodied agents? exploring backdoor attacks against embodied LLM-based decision-making systems. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=S1Bv3068Xt.

[18] Kento Kawaharazuka, Tatsuya Matsushima, Andrew Gambardella, Jiaxian Guo, Chris Paxton, and Andy Zeng. Real-world robot applications of foundation models: A review. *Advanced Robotics*, pages 1–23, 2024.

[19] Matt Klingensmith, Michael McDonald, Radhika Agrawal, Chris Allum, and Rosalind Shinkle. Boston dynamics blogs: Robots that can chat. URL https://bostondynamics.com/blog/robots-that-can-chat/.

[20] Nishanth Kumar, Tom Silver, Willie McClinton, Linfeng Zhao, Stephen Proulx, Tomás Lozano-Pérez, Leslie Pack Kaelbling, and Jennifer Barry. Practice Makes Perfect: Planning to Learn Skill Parameter Policies. *arXiv preprint arXiv:2402.15025*, 2024.

[21] Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660*, 2020.

[22] Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

[23] Jiwei Li, Michel Galley, Chris Brockett, Jianfeng Gao, and Bill Dolan. A diversity-promoting objective function for neural conversation models. *arXiv preprint arXiv:1510.03055*, 2015.

[24] Yanzhou Li, Tianlin Li, Kangjie Chen, Jian Zhang, Shangqing Liu, Wenhan Wang, Tianwei Zhang, and Yang Liu. BadEdit: Backdooring large language models by model editing, 2024.

[25] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor Learning: A Survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(1):5–22, 2024. doi: 10.1109/TNNLS.2022.3182979.

[26] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack

with sample-specific triggers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16463–16472, 2021.

[27] Yuan-Hong Liao, Xavier Puig, Marko Boben, Antonio Torralba, and Sanja Fidler. Synthesizing Environment-Aware Activities via Activity Sketches. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6284–6292, 2019. doi: 10.1109/CVPR.2019.00645.

[28] Aishan Liu, Yuguang Zhou, Xianglong Liu, Tianyuan Zhang, Siyuan Liang, Jiakai Wang, Yanjun Pu, Tianlin Li, Junqi Zhang, Wenbo Zhou, et al. Compromising embodied agents with contextual backdoor attacks. *arXiv preprint arXiv:2408.02882*, 2024.

[29] Bo Liu, Yuqian Jiang, Xiaohan Zhang, Qiang Liu, Shiqi Zhang, Joydeep Biswas, and Peter Stone. Llm+ p: Empowering large language models with optimal planning proficiency. *arXiv preprint arXiv:2304.11477*, 2023.

[30] Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. GPT understands, too. *AI Open*, 5:208–215, 2024.

[31] Zheyuan Liu, Xiaoxin He, Yijun Tian, and Nitesh V Chawla. Can we soft prompt LLMs for graph learning tasks? In *Companion Proceedings of the ACM on Web Conference 2024*, pages 481–484, 2024.

[32] Lajanugen Logeswaran, Yao Fu, Moontae Lee, and Honglak Lee. Few-shot subgoal planning with language models. *arXiv preprint arXiv:2205.14288*, 2022.

[33] Anh Nguyen and Anh Tran. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369*, 2021.

[34] Vishal Pallagani, Kaushik Roy, Bharath Muppasani, Francesco Fabiano, Andrea Loreggia, Keerthiram Murugesan, Biplav Srivastava, Francesca Rossi, Lior Horesh, and Amit Sheth. On the prospects of incorporating large language models (llms) in automated planning and scheduling (aps). In *34th International Conference on Automated Planning and Scheduling*, 2024.

[35] Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. Hidden trigger backdoor attack on {NLP} models via linguistic style manipulation. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3611–3628, 2022.

[36] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

[37] Cheng Peng, Xi Yang, Kaleb E Smith, Zehao Yu, Aokun Chen, Jiang Bian, and Yonghui Wu. Model tuning or prompt tuning? A study of large language models for clinical concept and relation extraction. *Journal of biomedical informatics*, 153:104630, 2024.

[38] Xavier Puig, Kevin Ra, Marko Boben, Jiaman Li, Tingwu Wang, Sanja Fidler, and Antonio Torralba. Virtualhome: Simulating household activities via programs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8494–8502, 2018.

[39] Adnan Siraj Rakin, Zhezhi He, and Deliang Fan. TBT: Targeted Neural Network Attack with Bit Trojan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13198–13207, 2020.

[40] 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 task planning. *arXiv preprint arXiv:2307.06135*, 2023.

[41] Alexander Robey, Zachary Ravichandran, Vijay Kumar, Hamed Hassani, and George J Pappas. Jailbreaking llm-controlled robots. *arXiv preprint arXiv:2410.13691*, 2024.

[42] Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. Fine-tuned language models are continual learners. *arXiv preprint arXiv:2205.12393*, 2022.

[43] Zhihong Shao, Minlie Huang, Jiangtao Wen, Wenfei Xu, and Xiaoyan Zhu. Long and diverse text generation with planning-based hierarchical variational model. *arXiv preprint arXiv:1908.06605*, 2019.

[44] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023.

[45] Jovan Stojkovic, Esha Choukse, Chaojie Zhang, Inigo Goiri, and Josep Torrellas. Towards Greener LLMs: Bringing Energy-Efficiency to the Forefront of LLM Inference. *arXiv preprint arXiv:2403.20306*, 2024.

[46] Karthik Valmeekam, Matthew Marquez, Alberto Olmo, Sarath Sreedharan, and Subbarao Kambhampati. Planbench: An extensible benchmark for evaluating large language models on planning and reasoning about change. *Advances in Neural Information Processing Systems*, 36, 2024.

[47] Yuxin Wang, Yuhan Chen, Zeyu Li, Zhenheng Tang, Rui Guo, Xin Wang, Qiang Wang, Amelie Chi Zhou, and Xiaowen Chu. Towards Efficient and Reliable LLM Serving: A Real-World Workload Study. *arXiv preprint arXiv:2401.17644*, 2024.

[48] Zhenting Wang, Juan Zhai, and Shiqing Ma. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15074–15084, 2022.

[49] Junda Wu, Tong Yu, Rui Wang, Zhao Song, Ruiyi Zhang, Handong Zhao, Chaochao Lu, Shuai Li, and Ricardo Henao. Infoprompt: Information-theoretic soft prompt tuning for natural language understanding. *Advances in Neural Information Processing Systems*, 36, 2024.

[50] Yaqi Xie, Chen Yu, Tongyao Zhu, Jinbin Bai, Ze Gong,

and Harold Soh. Translating natural language to planning goals with large-language models. *arXiv preprint arXiv:2302.05128*, 2023.

[51] Haomiao Yang, Kunlan Xiang, Mengyu Ge, Hongwei Li, Rongxing Lu, and Shui Yu. A comprehensive overview of backdoor attacks in large language models within communication networks. *IEEE Network*, pages 1–1, 2024. doi: 10.1109/MNET.2024.3367788.

[52] Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in nlp models. *arXiv preprint arXiv:2103.15543*, 2021.

[53] Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rethinking stealthiness of backdoor attack against nlp models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5543–5557, 2021.

[54] Lance Ying, Jason Xinyu Liu, Shivam Aarya, Yizirui Fang, Stefanie Tellex, Joshua B Tenenbaum, and Tianmin Shu. SIFToM: Robust Spoken Instruction Following through Theory of Mind. *arXiv preprint arXiv:2409.10849*, 2024.

[55] Lifan Yuan, Yangyi Chen, Ganqu Cui, Hongcheng Gao, Fangyuan Zou, Xingyi Cheng, Heng Ji, Zhiyuan Liu, and Maosong Sun. Revisiting Out-of-distribution Robustness in NLP: Benchmarks, Analysis, and LLMs Evaluations. *Advances in Neural Information Processing Systems*, 36, 2024.

[56] Hangtao Zhang, Chenyu Zhu, Xianlong Wang, Ziqi Zhou, Shengshan Hu, and Leo Yu Zhang. Badrobot: Jailbreaking llm-based embodied ai in the physical world. *arXiv preprint arXiv:2407.20242*, 2024.

[57] Tianjun Zhang, Shishir G Patil, Naman Jain, Sheng Shen, Matei Zaharia, Ion Stoica, and Joseph E Gonzalez. Raft: Adapting language model to domain specific rag. *arXiv preprint arXiv:2403.10131*, 2024.

[58] Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 179–197. IEEE, 2021.

[59] Chao Zhao, Shuai Yuan, Chunli Jiang, Junhao Cai, Hongyu Yu, Michael Yu Wang, and Qifeng Chen. Erra: An embodied representation and reasoning architecture for long-horizon language-conditioned manipulation tasks. *IEEE Robotics and Automation Letters*, 2023.

[60] Zhigen Zhao, Shuo Chen, Yan Ding, Ziyi Zhou, Shiqi Zhang, Danfei Xu, and Ye Zhao. A Survey of Optimization-based Task and Motion Planning: From Classical To Learning Approaches. *arXiv preprint arXiv:2404.02817*, 2024.

[61] Mengxin Zheng, Qian Lou, and Lei Jiang. Trojvit: Trojan insertion in vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4025–4034, 2023.

[62] Mengxin Zheng, Jiaqi Xue, Xun Chen, YanShan Wang, Qian Lou, and Lei Jiang. TrojFSP: Trojan Insertion in Few-shot Prompt Tuning. *arXiv preprint arXiv:2312.10467*, 2023.