Prompt a Robot to Walk with Large Language Models

Yen-Jen Wang[1], Bike Zhang[2], Jianyu Chen[1,3], Koushil Sreenath[2]

***Abstract*— Large language models (LLMs) pre-trained on vast internet-scale data have showcased remarkable capabilities across diverse domains. Recently, there has been escalating interest in deploying LLMs for robotics, aiming to harness the power of foundation models in real-world settings. However, this approach faces significant challenges, particularly in grounding these models in the physical world and in generating dynamic robot motions. To address these issues, we introduce a novel paradigm in which we use few-shot prompts collected from the physical environment, enabling the LLM to autoregressively generate low-level control commands for robots without task-specific fine-tuning. Experiments across various robots and environments validate that our method can effectively prompt a robot to walk. We thus illustrate how LLMs can proficiently function as low-level feedback controllers for dynamic motion control even in high-dimensional robotic systems. The project website and source code can be found at: prompt2walk.github.io.**

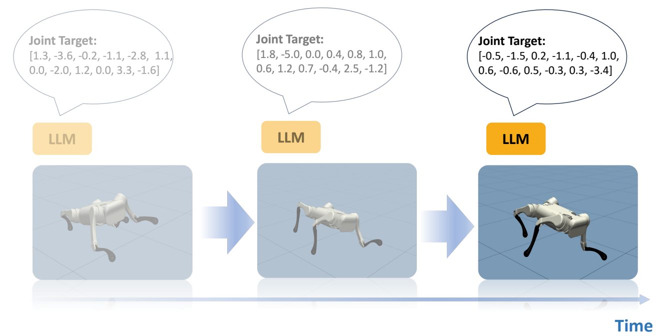

Fig. 1: **Prompt a Robot to Walk.** Grounded in a physics-based simulator, LLMs output target joint positions to enable a robot to walk given a text prompt, which consists of a description prompt and an observation and action prompt.

I. INTRODUCTION

Large language models (LLMs) pre-trained on internet-scale data [5], [32], [31], [9], [44] have demonstrated impressive results in various fields, e.g., natural language processing [28], [27], computer vision [30], code generation [7], etc. Building upon the success of LLMs, there is a surging interest in utilizing LLMs for embodied agents [1], [45], aiming to harness the power of foundation models in the physical world [2]. Towards this goal, significant progress has been made [4], [3], [10]. However, there are some remaining challenges. 1) Even though LLMs are trained with broad data at scale, the dataset does not incorporate data from the physical world, making it challenging to ground LLMs in robot control. 2) While foundation models have been widely used in a pre-training and fine-tuning paradigm for robotics applications, there could be a paradigm shift to few-shot learning in light of the progress of the natural language field [5]. 3) Most recent language-guided robot control research showcases mainly quasi-static robot motions. It remains uncertain whether LLMs can generate dynamic robot behaviors without a low-level controller interface or without relying on predefined motion primitives.

In this paper, we want to raise the intriguing question of whether LLMs can function as low-level controllers for achieving dynamic tasks like robot walking? This requires us to address the challenges mentioned above. We do this by exploring a new paradigm that leverages few-shot prompts with a large language model, i.e., GPT-4, to directly output robot control actions. We hypothesize that, given prompts collected from the physical environment, LLMs can learn to interact with it in context, even though they are purely trained on text data. Moreover, we do not perform any fine-tuning of the LLM with task-specific robot data. We adopt a few-shot prompt approach as widely adopted in the natural language field. Furthermore, we consider a dynamic control task of robot walking. A visualization of the paradigm is illustrated in Fig. 1. We term this paradigm as prompting a robot to walk. Grounded in a physical environment, LLMs output target joint positions to allow a robot to walk given a designed text prompt, which includes a description prompt and an observation and action prompt. Consequently, the robot is able to interact with the physical world through the generated control actions and get the observations from the environment. As a summary, the contributions of our work are as follows.

- Our main contribution is a framework for prompting a robot to walk with LLMs, where LLMs act as a feedback policy.
- We propose and systematically analyze a text prompt design that enables LLMs to in-context learn robot walking behaviors.
- We extensively validate our framework on different robots, various terrains, and multiple simulators.

A. Related Work

Large Language Models for Robotics. Large language models have recently become a popular tool for robotics including manipulation [1], [10], [3], [22], [16], [52], [15], locomotion [40], [53], navigation [14], [36], [13], [11], etc. Additionally, there are some recent research efforts to develop language agents [51], [38] using LLMs as the core.

[1]Institute of Interdisciplinary Information Sciences, Tsinghua University. [2]University of California, Berkeley. [3]Shanghai Qi Zhi Institute.

This work is supported in part by the InnoHK of the Government of the Hong Kong Special Administrative Region via the Hong Kong Centre for Logistics Robotics.

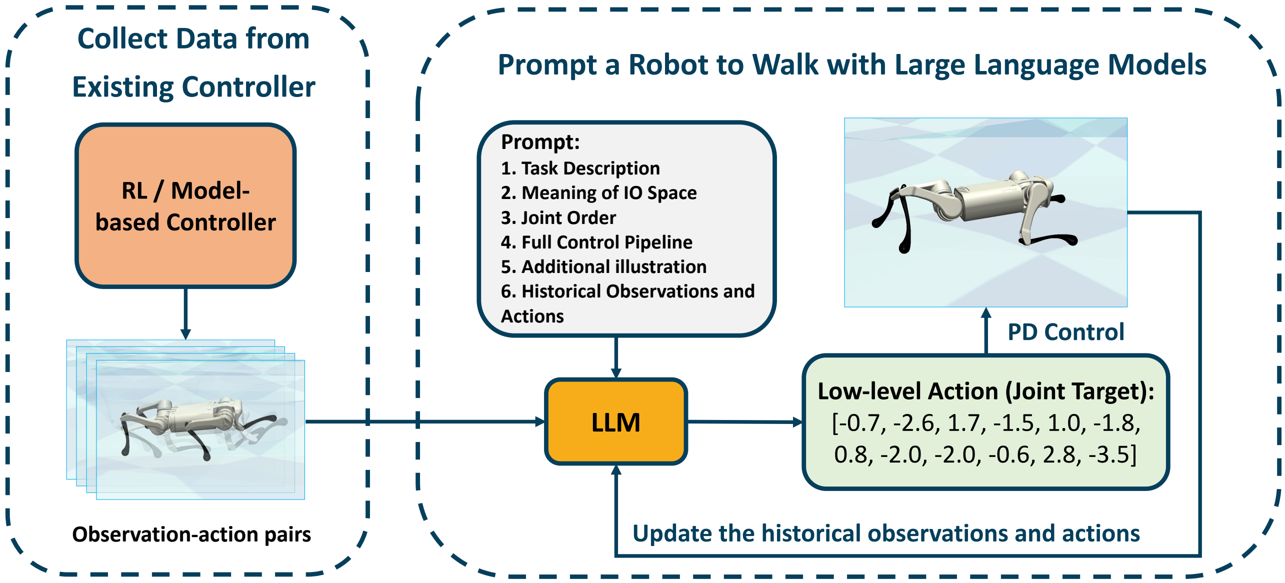

Fig. 2: **LLM Policy Overview.** We first collect data from an existing controller to initialize the LLM policy. Then, we design a text prompt including a description prompt and an observation and action prompt. The LLM outputs normalized target joint positions that are then tracked by a PD controller. After each LLM inference loop, the prompt is updated with the historical observations and actions. In our experiment, the LLM is supposed to run at 10 Hz although the simulation has to be paused to wait for LLM inference, and the PD controller executes at 200 Hz.

With a focus on the intersection between LLMs and low-level robot control, [46] trains a specialized GPT model using robot data to make a robot walk. However, our work directly uses the standard GPT-4 model without any fine-tuning. More interestingly, [25] instructs LLMs as general pattern machines and demonstrates a stabilizing controller for a cartpole in a sequence improvement manner [54]. Inspired by this work, we prompt LLMs to serve as a feedback policy for high dimensional robot walking. Note that our work prompts a feedback policy without iterative improvement, whereas the cartpole controller in [25] is gradually improved as a return-conditioned policy. In addition, we explore textual descriptions to enhance the policy.

Learning Robot Walking. Learning-based approaches have become promising methods to enable robots to walk. Deep reinforcement learning (RL) has been successfully applied to real-world robot walking [39], [18]. In [29], agile walking behavior is attained by imitating animals. To deploy a robot in complex environments, a teacher-student framework is proposed in [20], [19]. Moreover, a robot can learn to walk in the real world [37], [48]. Furthermore, the learning-based approach can enable dynamic walking behaviors [49], [21], [24], [50], [6], [55].

More recently, LLMs have emerged as a useful tool for helping create learning-based policies for robot walking. In [40], contact patterns are instructed by human commands through LLMs. In [53], LLMs are utilized to define reward parameters for robot walking. In contrast to previous LLM-based robot walking work, we use LLMs to directly output low-level target joint positions.

II. Method

In this section, we present our method of prompting a robot to walk with large language models. The overall framework is summarized in Fig. 2.

A. Data Collection

A proper text prompt is one of the keys to utilizing LLMs for robot walking. We initialize the prompt based on an existing controller, which could be either model-based or learning-based. From the existing controller, we collect observation and action pairs. The observation consists of sensor readings, e.g., IMU and joint encoders, while the action represents the target joint positions. It is important to note that the collected data serves as an initial input for LLM inference. As the robot begins to interact with the environment and acquire new observations, the initial offline data will be replaced by LLM outputs. Thus, we consider this data collection phase as an initialization step.

B. Prompt Engineering

Directly feeding observation and action pairs to LLMs often results in actions that do not achieve a stable walking gait. We next illustrate the prompt engineering step to guide LLMs to function as a feedback policy. Our prompt design, as shown in Fig. 3, can be classified into two categories: description prompt and observation and action prompt.

Description Prompt. The description prompt begins with P_{TD}, a precise description of the robot walking task. This is then followed by control design details, e.g., the policy's operating frequency, ensuring that the LLM aligns the actions to this frequency. Next, we specify the format and meaning of both observations and actions in P_{IO}, allowing LLMs

Fig. 3: **Text Prompt.** We design a text prompt that includes two parts: a description prompt and an observation and action prompt. In the description prompt, we have the following subparts: P_{TD}: task description, P_{IO}: meaning of input and output space, P_{JO}: joint order, P_{CP}: full control pipeline, and P_{AI}: additional illustration. In the observation and action prompt, we have P_{Hist}: historical observations and actions. The LLM outputs normalized target joint positions.

to understand the context of the inputs and actions. Then, an explicit enumeration of the joint order of our robot is provided in P_{JO} to guide the LLM to comprehend the robot configuration. Additionally, we specify in the prompt P_{AI} that all the values LLMs encounter are not raw data. Instead, these numerical values have been normalized. Lastly, the prompt offers an overview of the entire control pipeline in P_{CP}, granting the LLM a macro perspective on how individual components enabling it to process and interlink. It is crucial to highlight that, unlike classic learning-based and model-based walking controllers, text serves an important role in the LLM policy.

Observation and Action Prompt. A sequence of observation and action pairs P_{Hist} are used as prompts. These pairs are generated from the recent history of the robot walking trajectory. This procedure is widely used in RL-based robot walking controllers, where it allows the neural network to infer the dynamics as well as the privileged environment information. With a sequence of observation and action prompts, LLMs can in-context learn the dynamics and infer a reactive control action, where the observation prompt serves as the feedback signal. Note that both observation and action are converted to text format to interface with LLMs.

LLMs often struggle to comprehend the significance of numeric values, particularly floating point and negative numbers. Inspired by the prompt design in [25], we adopt a normalization approach for numerical values. Specifically, we use a linear transformation to map all the potential numeric values into non-negative integers, ranging from 0 to 200. We hypothesize that LLMs are mostly trained with text tokens, thus they are not sensitive enough to numerical values for robot control.

C. Grounding LLMs

In order to make LLMs useful for robot walking control, we need to ground them in a physical environment. We now introduce the pipeline to allow LLMs to interact with a robot and an environment. We use a physics-based simulator where LLMs can get observations and send actions. The observations are from the physics-based simulation. The output of the LLM is the target joint positions, which are tracked by a set of joint Proportional-Derivative (PD) controllers running at a higher frequency. This joint-level PD control design is standard for learning-based robot walking control. While this pipeline is entirely done in simulation in this work, it has the potential to be implemented on hardware if the inference speed of LLMs is fast enough.

III. Results

Having introduced the methodology for prompting a robot to walk, we next detail our experiments for validation. Moreover, through these experiments, we aim to answer the following questions:

Q1: Can we prompt a robot to walk with LLMs?

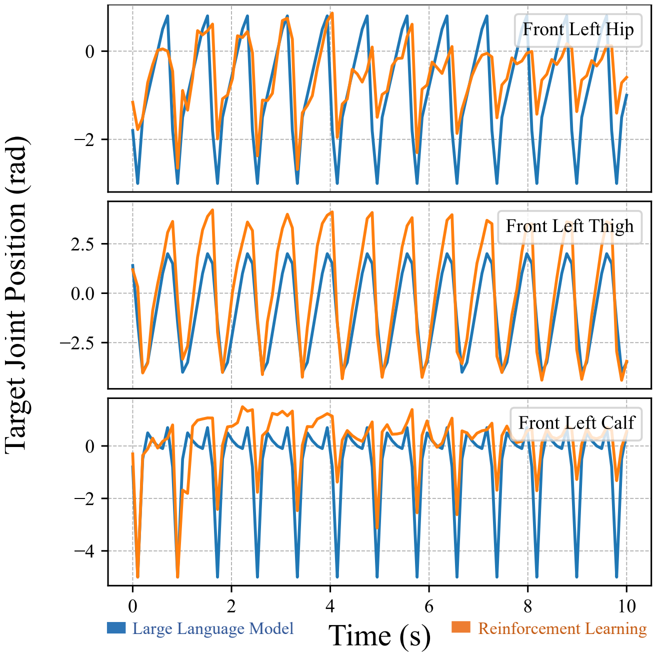

Fig. 4: **Target Joint Position Trajectories.** The LLM and RL-based target joint position trajectories for the front left leg, including hip, thigh, and calf joints. The LLM trajectory is depicted in blue and the RL trajectory is shown in orange.

Q2: How should we design prompts for robot walking?

Q3: Does the proposed approach generalize to different robots and environments?

A. Setup

We choose an A1 quadruped robot as our testbed [33]. It is a high-dimensional system with 12 actuated joints. To initialize the LLM policy, we train an RL policy in Isaac Gym [23] using Proximal Policy Optimization (PPO) [35]. This training is based on the training recipe from [34]. Subsequently, we ground the LLM in Mujoco [42], a high-fidelity, physics-based simulator. Our LLM policy operates at 10 Hz and is then tracked by a low-level joint PD controller at 200 Hz. The PD gains for this controller are set at 20 and 0.5, respectively.

After evaluating various LLMs including GPT-4 [27], GPT-3.5-Turbo, text-davinci-003 [26], Alpaca [41], Vicuna 2 [8], Llama 2 [43], we found that only GPT-4 is powerful enough to in-context learn a robot walking behavior using our designed prompt. During the experiments, we set GPT-4's temperature to 0 to minimize the variance.

B. Robot Walking

Utilizing the proposed approach, we successfully prompt an A1 quadruped robot to walk with GPT-4. The LLM policy cannot only enable walking on flat ground but can also allow the robot to walk over uneven terrain as shown in Fig. 8. Due to the unexpected roughness, the robot almost falls over but the LLM policy makes it recover to a normal posture and then keeps walking forward. Due to the need to balance the token limit of the LLM and the size of P_{Hist}, we execute the policy at 10 Hz. However, this leads to a walking gait that becomes reasonably worse compared to many RL-based walking policies running at 50 Hz or even higher.

Fig. 4 demonstrates target joint trajectories for the front left leg when a robot is walking on uneven terrain for 10 seconds. The blue lines depict the trajectories produced by the LLM policy. As a comparison, the orange lines show the trajectories generated by an RL policy. Note that both trajectories take the same observation as input. The robot acts with the action generated by the LLM and then gets the next observation from the environment. Although the LLM policy is initialized with the RL policy, the resulting joint trajectories are noticeably different.

One prompt example for A1 robot walking is shown in Fig. 3, where we use historical observations and actions for the past 50 steps. The prompt is specially designed and normalized as described in Sec. II-B. Based on this A1 robot walking experiment, we can answer Question Q1 that a robot can be prompted to walk with LLMs.

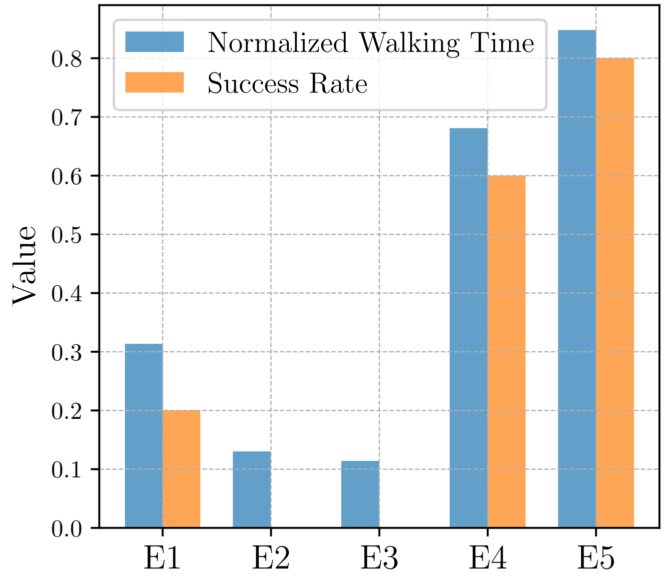

Fig. 5: **Description Prompt Comparison.** E1. No description prompt. (i.e. only P_{Hist}) E2. P_{IO}: meaning of input and output space. E3. $P_{IO} + P_{JO}$: meaning of input and output space and joint order. E4. $P_{TD} + P_{IO} + P_{JO} + P_{CP}$: task description, meaning of input and output space, joint order, and full control pipeline. E5. Full description prompt.

C. Description Prompt

We perform 5 experiments to analyze the impact of individual components in the description prompt. In each experiment, we provide observation and action prompts (P_{Hist}). For evaluation, we consider two metrics: normalized walking time and success rate. To clarify, the term "normalized walking time" denotes the proportion of time a robot can walk before it falls. The success rate is measured by the percentage of the trials that the robot is able to finish, where each trial lasts for 10 seconds and we have 5 trials for each experiment. In the design of the first experiment (E1), we exclude the description prompt entirely (only P_{Hist}). In the

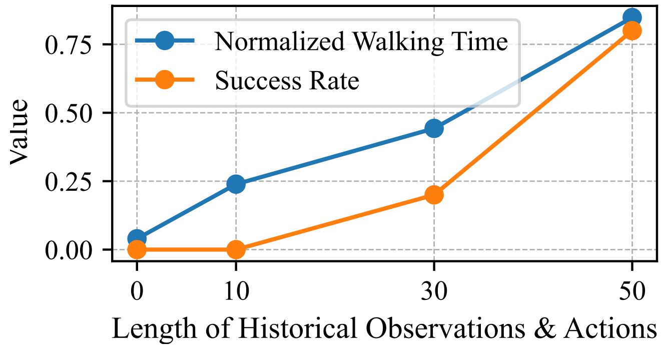

Fig. 6: **Observation and Action Length Comparison.** We conduct experiments for historical observation and action lengths as $0, 10, 30$, and 50. With lengths ranging from 0 to 50, the LLM token consumptions are approximately $348, 1738, 4518$, and 7298, respectively.

second experiment (E2), we only provide the meaning of input and output space (P_{IO}). Additionally, we include the joint order ($P_{IO} + P_{JO}$) in the third experiment (E3). In the fourth experiment (E4), we incorporate prompts such as task description, meaning of input and output space, joint order, and the full control pipeline ($P_{TD} + P_{IO} + P_{JO} + P_{CP}$). For the fifth experiment (E5), we employed a complete description prompt. The experimental result is demonstrated in Fig. 5, where we can see that the full description prompt has the highest normalized walking time and success rate. Based on the results from the first experiment, without a description prompt (E1), there is a minimal likelihood of LLMs prompting a robot to walk.

D. Observation and Action Prompt

In our subsequent investigation, we assess the influence of the observation and action prompt P_{Hist} on walking performance. Inspired by the RL-based walking control design, we first study how historical observations and actions affect the performance. We conduct a series of experiments, testing observation and action lengths of $0, 10, 30$, and 50, all while using the description prompt. To clarify, a length of 0 means only a description prompt. In our experiments, the LLM is queried at 10 Hz, so a length of 50 means 5 seconds in wall time that covers several walking steps for a quadruped robot. The experimental result is shown in Fig. 6. It is evident that increased lengths of observations and actions correlate with enhanced performance, both in terms of normalized walking time and success rate. With lengths ranging from 0 to 50, the LLM token consumptions are approximately $348, 1738, 4518$, and 7298, respectively. As we use the GPT-4 model with an 8k token length, we are not able to explore longer lengths of observations and actions.

In addition to comparing various lengths for observation and action prompts, we also investigate the effect of different observation prompts. Our choices for observations are influenced by the RL policy, as we initialize our LLM policy using a reinforcement learning-based approach. We evaluated five scenarios: (E1) no observation; (E2) only base linear velocity and angular velocity; (E3) only joint position and joint velocity; (E4) a combination of base linear velocity, angular velocity, joint position, and joint velocity; (E5) full observation. The comparison result is shown in Fig. 7. The full observation listed in Fig. 3 achieves the best performance. However, it remains unclear which specific observation component is the most influential. It is noteworthy that the observation in the LLM policy has a dimension of 33 while the observation space in the RL policy has a dimension of 48, which indicates that the LLM policy can use less information to make a robot walk compared to an RL policy.

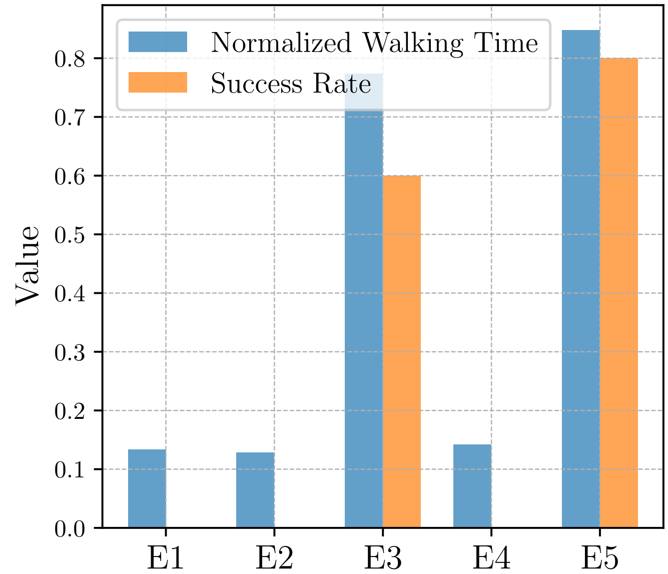

Fig. 7: **Observation Choice Comparison.** E1. No observation. E2. Base linear velocity and angular velocity. E3. Joint position and joint velocity. E4. Combine observations from experiments 2 and 3. E5. Full observation.

Furthermore, we study the effect of how we normalize the observation and action prompt. We benchmark 5 different normalization methods: (E1) original values without any normalization; (E2) normalize to positive values; (E3) normalize to integers; (E4) discard the decimal part and then normalize the integer part to positive integer values; (E5) normalize to positive integer values. Due to the limited token size of GPT-4, we opt for a compact observation prompt consisting of base linear and angular velocities. The benchmark result is summarized in TABLE I. Unlike other experiments, to emphasize the performance in different normalization methods, we extend the walking time to 20 seconds. We found that the normalization of the observation and action prompt is crucial as LLMs might parse a value of observation or action into several text tokens.

Based on the investigation of the text prompt, we can answer Question Q2: how should we design prompts for robot walking? We believe a synergy between description prompt and observation and action prompt is the key to utilizing LLMs to prompt a robot to walk.

Experiment	E1	E2	E3	E4	E5
NWT(↑) [%]	0.137	0.086	0.700	0.504	**0.721**
Success Rate(↑) [%]	0.0	0.0	**0.6**	0.2	**0.6**
No. Input Tokens(↓)	4947	5117	**3135**	**3135**	**3135**
No. Output Tokens(↓)	62	62	**38**	**38**	**38**

TABLE I: **Normalization Method Benchmark.** E1. Original values. E2. Normalize to positive values. E3. Normalize to integer values. E4. Discard the decimal and then normalize the integer to positive integer values. E5. Normalize to positive integer values. NWT is normalized walking time.

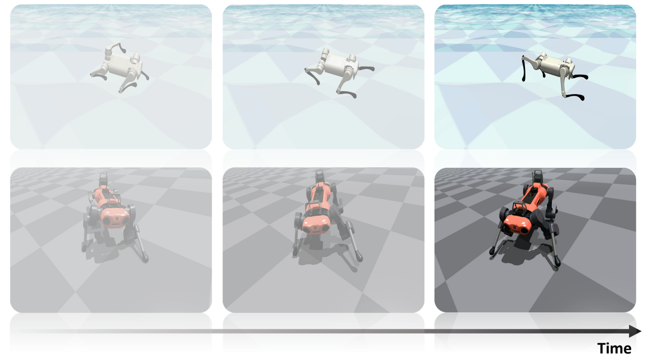

Fig. 8: **Robot Walking Visualization.** Top: A1 robot is prompted to walk on uneven terrain in MuJoCo, where the LLM policy can make it recover from terrain disturbance. Bottom: ANYmal robot is prompted to walk on flat ground in Isaac Gym using the same approach.

E. Different Robots

In addition to the A1 robot, we further validate our approach with a different robot: the ANYmal robot [17]. It is different from the A1 robot in terms of size, mass, mechanical design, etc. In this experiment, we use Isaac Gym instead of MuJoCo as our simulator to see the effect of change in the simulation environment. Following the same approach, we train a 10 Hz RL policy for initialization. With the proposed text prompt, we successfully prompt the ANYmal robot to walk on flat ground. Snapshots of ANYmal walking are shown in Fig. 8. Having been validated by the A1 and ANYmal experiments over various terrains, we believe that the proposed method generalizes to different robots and environments, which is our answer to Question Q3.

IV. Discussion

After validating our approach with experimental results, we provide a discussion on what we learned in this study and the limitations of the current approach.

A. Text Is Another Interface for Control

It is interesting to note that the description prompt plays a crucial role in utilizing LLMs to prompt a robot to walk, which indicates that text is another interface for control. The existing control approaches for robot walking do not rely on any task description in textual form. If we follow the convention of RL or model-based control that uses numerical values such as observations and actions, LLMs have a low chance of making a robot walk, as demonstrated in Fig. 5. Instead, with a proper design of the description prompt, LLMs can achieve a high success rate for walking. We hypothesize that a description prompt provides a context for LLMs to interpret the observations and actions properly. While we provide a prompt example for robot walking, the prompt design for robot motions is still under-explored.

B. LLMs In-Context Learn Differently

Our experiments demonstrate that LLMs in-context learn to prompt a robot to walk. Initially, we hypothesized that LLMs might learn a robot walking behavior in a manner akin to behavior cloning [47]. However, as shown in Fig. 4, the joint trajectories generated by the LLM policy are sufficiently different from those generated by an RL policy. Moreover, the LLM policy shows a more regular pattern which is not present in the RL policy. If we pay attention to the left calf joint trajectory, the pattern coincides with the biomechanics study of animal walking [12]. Thus, we believe that LLMs in-context learn differently to enable a robot to walk.

C. Limitations

While this work takes us closer towards utilizing LLMs for robot walking control, there are some limitations in the current framework. First, the current prompt design is fragile. Minor alterations in the prompt can dramatically affect the walking performance, as described in our experiments. In general, we still lack a good understanding of how to design a reliable prompt for robot walking. Secondly, as we design and test the prompt based on a specific initialization policy, our prompt design inevitably becomes biased towards this policy. Although we have tested our framework with several different RL initialization policies, it is possible that some initialization policies do not work with our prompt.

Another major limitation is that we are only able to carry out simulation experiments instead of hardware experiments. One reason is the low inference speed of GPT-4. Our pipeline requires LLMs to be queried at 10 Hz, which is much faster than the actual inference speed through OpenAI API. Thus, we have to pause the simulation to wait for the output of GPT-4. Furthermore, due to the limited token size, we have to choose a low-frequency policy, i.e., 10 Hz, to maximize the time horizon of the context. As a side note for future research, this work is expensive and roughly costed $2,000 US dollars for querying OpenAI API to test the prompt.

V. Conclusions

In this paper, we presented an approach for prompting a robot to walk. We use LLMs with text prompts, consisting of a description prompt and an observation and action prompt collected from the physical environment, without any task-specific fine-tuning. Our experiments demonstrate that LLMs can serve as low-level feedback controllers for dynamic motion control even in high-dimensional robotic systems. We further systematically analyzed the text prompt with extensive experiments. Furthermore, we validated this method across various robotic platforms, terrains, and simulators.

REFERENCES

[1] 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.

[2] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, *et al.*, "On the opportunities and risks of foundation models," *arXiv preprint arXiv:2108.07258*, 2021.

[3] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, *et al.*, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," *arXiv preprint arXiv:2307.15818*, 2023.

[4] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *arXiv preprint arXiv:2212.06817*, 2022.

[5] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[6] K. Caluwaerts, A. Iscen, J. C. Kew, W. Yu, T. Zhang, D. Freeman, K.-H. Lee, L. Lee, S. Saliceti, V. Zhuang, *et al.*, "Barkour: Benchmarking animal-level agility with quadruped robots," *arXiv preprint arXiv:2305.14654*, 2023.

[7] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. d. O. Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, *et al.*, "Evaluating large language models trained on code," *arXiv preprint arXiv:2107.03374*, 2021.

[8] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, *et al.*, "Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality," *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[10] D. Driess, F. Xia, M. S. Sajjadi, C. Lynch, A. Chowdhery, B. Ichter, A. Wahid, J. Tompson, Q. Vuong, T. Yu, *et al.*, "Palm-e: An embodied multimodal language model," *arXiv preprint arXiv:2303.03378*, 2023.

[11] Y. Guo, Y.-J. Wang, L. Zha, Z. Jiang, and J. Chen, "Doremi: Grounding language model by detecting and recovering from plan-execution misalignment," *arXiv preprint arXiv:2307.00329*, 2023.

[12] P. Holmes, R. J. Full, D. Koditschek, and J. Guckenheimer, "The dynamics of legged locomotion: Models, analyses, and challenges," *SIAM review*, vol. 48, no. 2, pp. 207–304, 2006.

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

[14] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch, "Language models as zero-shot planners: Extracting actionable knowledge for embodied agents," in *International Conference on Machine Learning*. PMLR, 2022, pp. 9118–9147.

[15] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and L. Fei-Fei, "Voxposer: Composable 3d value maps for robotic manipulation with language models," *arXiv preprint arXiv:2307.05973*, 2023.

[16] 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.

[17] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, *et al.*, "Anymal-a highly mobile and dynamic quadrupedal robot," in *2016 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2016, pp. 38–44.

[18] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, p. eaau5872, 2019.

[19] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[20] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science robotics*, vol. 5, no. 47, p. eabc5986, 2020.

[21] C. Li, M. Vlastelica, S. Blaes, J. Frey, F. Grimminger, and G. Martius, "Learning agile skills via adversarial imitation of rough partial demonstrations," in *Conference on Robot Learning*. PMLR, 2023, pp. 342–352.

[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," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 9493–9500.

[23] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa, *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[24] G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid locomotion via reinforcement learning," *arXiv preprint arXiv:2205.02824*, 2022.

[25] 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.

[26] OpenAI, "Gpt-3.5 documentation," 2023. [Online]. Available: https://platform.openai.com/docs/models/gpt-3-5

[27] ——, "Gpt-4 technical report," *arXiv*, 2023.

[28] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[29] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, and S. Levine, "Learning agile robotic locomotion skills by imitating animals," *arXiv preprint arXiv:2004.00784*, 2020.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[31] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, *et al.*, "Improving language understanding by generative pre-training," 2018.

[32] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[33] U. Robotics, 2023. [Online]. Available: https://unitreerobotics.net/

[34] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 91–100.

[35] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[36] D. Shah, B. Osiński, S. Levine, *et al.*, "Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action," in *Conference on Robot Learning*. PMLR, 2023, pp. 492–504.

[37] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine, "Legged robots that keep on learning: Fine-tuning locomotion policies in the real world," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 1593–1599.

[38] T. Sumers, S. Yao, K. Narasimhan, and T. L. Griffiths, "Cognitive architectures for language agents," *arXiv preprint arXiv:2309.02427*, 2023.

[39] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots," *arXiv preprint arXiv:1804.10332*, 2018.

[40] Y. Tang, W. Yu, J. Tan, H. Zen, A. Faust, and T. Harada, "Saytap: Language to quadrupedal locomotion," *arXiv preprint arXiv:2306.07580*, 2023.

[41] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Alpaca: A strong, replicable instruction-following model," *Stanford Center for Research on Foundation Models. https://crfm. stanford. edu/2023/03/13/alpaca. html*, vol. 3, no. 6, p. 7, 2023.

[42] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *2012 IEEE/RSJ international conference on intelligent robots and systems*. IEEE, 2012, pp. 5026–5033.

[43] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[44] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[45] 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.

[46] N. Wagener, A. Kolobov, F. Vieira Frujeri, R. Loynd, C.-A. Cheng, and M. Hausknecht, "Mocapact: A multi-task dataset for simulated humanoid control," *Advances in Neural Information Processing Systems*, vol. 35, pp. 35 418–35 431, 2022.

[47] J. Wei, J. Wei, Y. Tay, D. Tran, A. Webson, Y. Lu, X. Chen, H. Liu, D. Huang, D. Zhou, *et al.*, "Larger language models do in-context learning differently," *arXiv preprint arXiv:2303.03846*, 2023.

[48] P. Wu, A. Escontrela, D. Hafner, P. Abbeel, and K. Goldberg, "Daydreamer: World models for physical robot learning," in *Conference on Robot Learning*. PMLR, 2023, pp. 2226–2240.

[49] Z. Xie, X. Da, B. Babich, A. Garg, and M. v. de Panne, "Glide: Generalizable quadrupedal locomotion in diverse environments with a centroidal model," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2022, pp. 523–539.

[50] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conference on Robot Learning*. PMLR, 2022, pp. 773–783.

[51] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," *arXiv preprint arXiv:2210.03629*, 2022.

[52] S. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T.-Y. Yang, V. Jain, A. W. Clegg, J. Turner, *et al.*, "Homerobot: Open-vocabulary mobile manipulation," *arXiv preprint arXiv:2306.11565*, 2023.

[53] W. Yu, N. Gileadi, C. Fu, S. Kirmani, K.-H. Lee, M. G. Arenas, H.-T. L. Chiang, T. Erez, L. Hasenclever, J. Humplik, *et al.*, "Language to rewards for robotic skill synthesis," *arXiv preprint arXiv:2306.08647*, 2023.

[54] D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le, *et al.*, "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[55] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao, "Robot parkour learning," *arXiv preprint arXiv:2309.05665*, 2023.