Preference-Based Long-Horizon Robotic Stacking with Multimodal Large Language Models

Wanming Yu*, Adrian Röfer†, Abhinav Valada†, Sethu Vijayakumar*
*School of Informatics, University of Edinburgh
Email: {wanming.yu, sethu.vijayakumar}@ed.ac.uk
†University of Freiburg
Email: {aroefer, valada}@cs.uni-freiburg.de

***Abstract*—Pretrained large language models (LLMs) can work as high-level robotic planners by reasoning over abstract task descriptions and natural language instructions etc. However, they have shown a lack of knowledge and effectiveness for planning long-horizon robotic manipulation tasks where the physical properties of the objects are essential. An example is stacking of containers with hidden objects inside, which involves reasoning over hidden physics properties such as weight and stability. To this end, this paper proposes to use multimodal LLMs as high-level planners for such long-horizon robotic stacking tasks. The LLM takes multimodal inputs for each object to stack and infers the current best stacking sequence by reasoning over stacking preferences. Given explicit instructions to consider weight and stability at the same time as the stacking preference, the Kawada NEXTAGE humanoid robot showcased the successful stacking of three boxes with various hidden objects guided by an LLM on-the-fly in the real world. Furthermore, in order to enable the LLM to reason over multiple preferences at the same time without giving explicit instructions, we propose to create a custom dataset to fine-tune the LLM. We simulate all possible stacks of boxes with various contents in physics simulation, and generate training samples for stacking preferences including weight, stability, size, and foothold.**

***Index Terms*—Long-Horizon Manipulation, Robotic Stacking, Multimodal Reasoning, Large Language Models.**

I. Introduction

Stacking of boxes or containers is a common problem in various domains, ranging from private homes, to shops, to warehouses. The challenge in this problem lies with a multitude of preferences to consider, such as the size, weight, and structural stability of the items being stacked. Moreover, the properties of the objects being stacked might not even be readily apparent from visual observation – all moving boxes look the same, but when lifted one notices that some contain books and others bedding. Despite all these challenges and considerations, humans can easily trade off structural constraints with preferences and successfully reorganize a storage room.

Planning approaches have been proposed for performing robotic stacking tasks, such as rule-based planning, task and motion planning. However, these planning approaches usually require expert knowledge or engineering efforts, suffer from limited flexibility or scalability to planning more complex scenarios on-the-fly. Another line of related work is using reinforcement learning which usually brings more robustness and flexibility but can be difficult to scale to long-horizon task learning or scenarios with more objects.

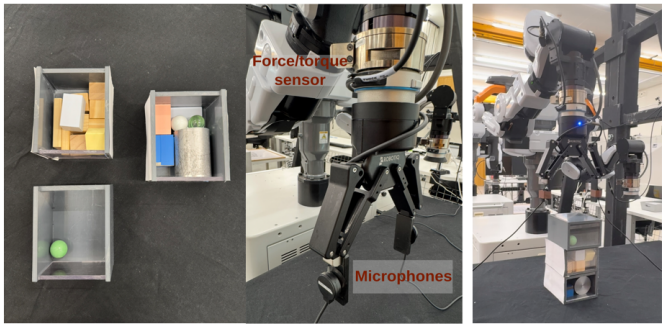

Fig. 1. Long-horizon stacking task by multimodal large language models. Left: Three containers to be stacked with various contents. Middle: Multimodal sensing setup for object hidden properties. Right: The NEXTAGE robot performing box stacking guided by an LLM.

The recent rise of large-scale foundation models such as Large Language Models (LLMs) has demonstrated promising reasoning capabilities about common-sense knowledge and advantages in long-horizon task planning [1]–[3]. Vision Language Models (VLMs) [4] combine LLMs with vision encoders have been used for manipulation tasks where objects have distinct visual appearances. VLMs have been shown to work effectively as high-level planners for generating subgoals for certain tasks but lack of knowledge of physics understandings. Built on top of VLMs, Vision Language Action Models (VLAs) are mainly developed aiming at infer actions for general robotic tasks taking physics into consideration [5]–[7], however, they require a huge amount of data to train and usually has to be finetuned for specific task in order to achieve an acceptable success rate and improve the performance.

Moreover, the objects in these previous work usually have distinct visual appearances, such as shapes and colors. However, for stacking containers, the hidden objects inside can affect the final stacking order while their latent characteristics may only become evident during manipulation. The previous approaches have rarely focused on planning or reasoning based on physics properties of objects via multimodal sensing [8]. Therefore, long-horizon stacking of objects with hidden properties remains a challenging problem in robotics research.

To this end, we propose a multimodal large language model

as orchestrator of the high-level behaviors for performing long-horizon robotic stacking of objects with hidden properties. Regarding the hidden properties, we use force/torque sensor and microphones to get weight and stability of each box, respectively. The multimodal system we present, displayed in Fig. 1, is able to infer the current best stacking sequence on-the-fly according to the hidden properties, human preferences and the current stacking status. Moreover, we introduce a lightweight paradigm to generate additional training data in physics simulation for the model to become more accustom to considering physical stability and plausibility of the generated plans. We showcased successful long-horizon stacking on a real humanoid robot. Our contributions are as follows:

- A multimodal large language model for planning long-horizon stacking of objects with hidden properties, where audio signals (for hidden stability detection) and force sensing (for hidden weight detection) are considered.
- A robust framework for offline planning and online re-planning of high-level stacking sequence with the constraints imposed by human preference (weight, stability etc).
- A dataset generation scheme in physics simulation for finetuning the large language model to balance multiple human-specified stacking preferences.

II. Related Work

A. Reinforcement Learning for Robotic Stacking

Reinforcement learning has been applied to robotic stacking of objects with diverse shapes [9]. However, this is based on the visual properties of the objects and does not consider the case where boxes may have same appearance but different content inside. Moreover, it is difficult to scale to long-horizon stacking.

B. Vision Language Action (VLAs) Models for General Robotic Tasks

These VLAs usually trained based on pretrained VLMs. Since VLMs do not include knowledge in robotics, a huge amount of data of various robotic tasks of different types of robots need to be collected on top of those VLM data in order to handle general robotic tasks. VLAs are difficult for zero-shot deployment of new tasks or on new robots, and need further data for finetuning the model. Furthermore, these VLAs mainly focus on the tasks where relevant objects have identifiable visual differences. In our object stacking task, some of the objects may have exactly the same visual appearance but different hidden properties.

III. Problem Formulation

We consider a box stacking scenario, in which the robot is required to stack B boxes, each with hidden objects inside, into a single stack. Each box b has a visual property $\mathcal{V}_b$, an initial pose ${}^{W}\mathbf{T}_{b,0}$, and an initially unknown inertial property $\mathcal{I}_b$ associated with it. Given a user prompt Q constructed from multiple modalities, the aim is to generate a stacking order $S = \langle b_i, \ldots, b_j \rangle$. As the inertial properties $\mathcal{I}_b$ are initially unknown, we need to interact with each box to get the latent characteristics before planning the stacking sequence. The long-horizon stacking can be performed in either an offline or online manner, as shown in Fig. 2.

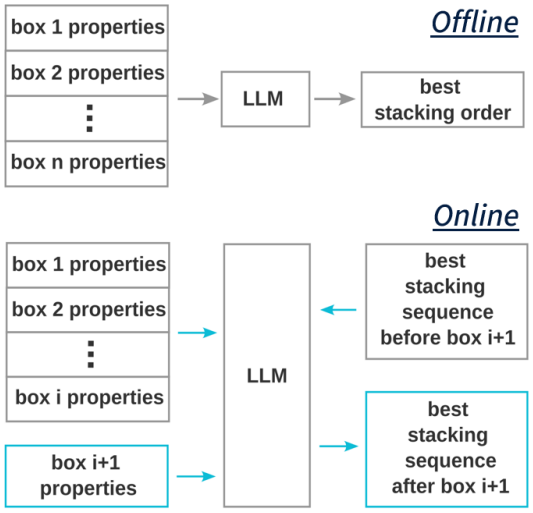

Fig. 2. The LLM can plan long-horizon stacking tasks in either an offline (top) or online manner (bottom).

Offline planning: The agent is given the observed inertial properties for all boxes all at once and generates the best stacking plan.

Online planning: After the robot interacts with box b, the agent is given the observed inertial properties $\mathcal{I}_b$ of box b. The agent can then update the previous stacking plan if necessary according to available information, or simply continue to interact with another box.

IV. Methodology

A. Framework Overview

The main research question we aim to address in this work is to leverage large-scale foundation models to reason and plan the stacking sequence according to hidden physics properties including weight and stability of the objects. To this end, we propose to use a large language model to consider multi-modal sensing data for planning the long-horizon object stacking tasks according to various stacking preferences. An overview of our proposed multimodal long-horizon stacking planning framework is shown in Fig. 3. The LLM takes multimodal inputs for each box, generates the current best stacking sequence according to specified preferences, and passes to the motion controllers to perform the sequence. Furthermore, we propose an effective dataset generation scheme for fine-tuning LLMs to balance multiple physics-based preferences without the need for giving explicit instructions.

B. Multimodal Large Language Model for Long-Horizon Stacking

Given object weight, object stability, and the current stacking status, the large language model will be able to infer

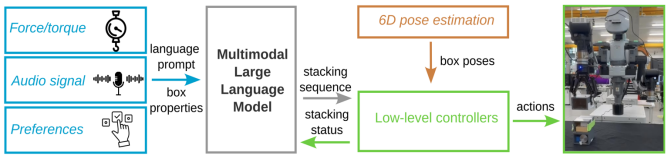

Fig. 3. Overview of the proposed long-horizon robotic stacking framework.

current best stacking sequence according to specified stacking preferences.

C. Hidden Property Sensing

The hidden properties essential to the object stacking task mainly include the object weight and stability due to unknown items inside. In the beginning, we let the NEXTAGE robot lift up each box and use force-torque sensor on the wrist to get the weight. Regarding the stability, we let the robot tilt the box within 90 degrees and record the audio signals from microphones on the gripper. The audio signals will be accumulated during the tilting motion to reflect the object stability. The values of weight and stability will be passed to construct the language prompt. Then, the constructed language prompt will be passed to the LLM together with the current stacking status.

D. Dataset Generation with Preferences for LLM Fine-Tuning

We obtain training data for fine-tuning the LLM using a physics simulation environment based on PyBullet. Examples of stable stacks generated from our physics simulation is shown in Fig. 4. In our simulation, we generate closed boxes of different sizes and varying material densities (cardboard, wood, and plastic), and fill these with smaller objects of different simple geometric shapes (spheres, boxes, and cylinders). We assign these objects a stability score

$$s_o = \frac{\min(w, d)}{h}, \tag{1}$$

where w, d are width and depth of the object's footprint and h is its height. In the case of a cylinder w, d equal the cylinder's diameter. If o is a sphere, $s_o = 0$. We compute a overall stability score for a box as the mean of the stability scores of its contents.

While satisfying an individual preference such as "Stack the boxes heavies to lightest" defines an ordering for boxes, it is not said that this order actually yields a stable stack. It might be that the heaviest box is a lot smaller than the other ones, which would lead to an overall unstable tower. In addition, it there might also be multiple valid orders to satisfy a preference or combination of them. These considerations make it very difficult to directly generate desirable executions for the agents to learn from. Instead, we simulate all possible stacks for a given initial sample of boxes and compute their scores with respect to different preferences (Fig. 5) or combinations thereof after the fact.

For a scenario S of K boxes we generate, there are $K!$ possible stacking orders. For each order, we stack the boxes one-by-one on top of one another. The boxes are stacked horizontally centered with some added Gaussian noise. We apply a linear momentum to the box being stacked on top and its contents. The momentum is downwards facing deviating up to 30 degrees from the gravity vector. This momentum simulates that a box being placed is not decelerated to $0m/s$ at the moment of contact with the bottom box, which might topple the stack. We wait for the simulation to return to rest before the next box is placed. If the stack falls over, we stop the execution of the current order.

Each stack of boxes $a_i = \langle b_1, \ldots, b_k \rangle$ with $k \leq K$ can be scored under a preference p by measuring its deviation from the optimal preference a_p^* which is obtained by stably sorting a_i under p as $a_p^* = \text{sort}_p(a_i)$. We compare the two sequences using the Levenshtein distance $\text{lev}(a_i, a_p^*)$ which we normalize by the length of the sequence $|a_i|$ as

$$\phi(a, p) = \frac{\text{lev}(a_i, \text{sort}_p(a))}{|a|} \tag{2}$$

If $a_i = a_p^*$ this metric is 0, if all positions in a_i differ from those in a_p^* the metric is 1. From our simulated set of all samples, we can now generate training examples for specific single or mixed preferences. We distinguish preferences as building on visually apparent characteristics, such as a box's size as $\hat{p}$, and characteristics requiring interaction as $\bar{p}$. Let i then be a particular order and j be the step in building the full stack i. We generate a training sample for a joint preference $P = \{\bar{p}_1, \ldots, \bar{p}_{\bar{N}}, \hat{p}_1, \ldots, \hat{p}_{\hat{N}}\}$ by picking i as

$$a_i = \arg\max_{a \in A_S} \frac{1}{\hat{N}} \sum_{n}^{\hat{N}} \phi(a, p_n), \tag{3}$$

where A_S is the set of all completed stable stacks generated for scenario S in simulation. Given this selected target, we increment j and check after each increment, if there is another partial stack $a_{l,j}$ with the same objects which scores better under the full average score of P than the current stack $a_{i,j}$. If such an l does exist we generate a unstacking and stacking actions to convert $a_{i,j}$ to $a_{l,j}$ making a_l the new current stack a_i.

For any selection P we provide different prompt options and generate text for the stacking and unstacking actions in the format which we demand of the LLM. We do not include any visual observations to prevent overfitting to the appearance of the simulation. This procedure allows us to generate the needed amount of training data in a reasonable amount of time. While simulating all $K!$ stacking order is time consuming, generating thousands of different executions from the simulation results is efficient and flexible.

V. Experimental Results

A. Ablation Studies

1) Rule-based method: We implemented rule-based method for stacking three boxes as a baseline. Instead of using LLM to generate the stacking sequence, we manually define the corresponding stacking sequence for each possible stacking status involving stacking and unstacking. In particular, when the existing stack needs to be adjusted, we found it would

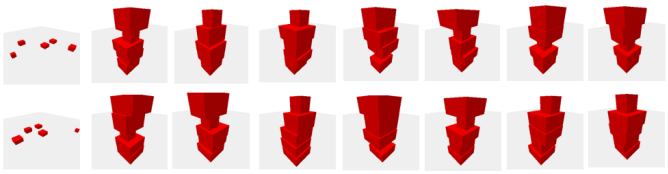

Fig. 4. Examples of stable stacks generated from physics simulation.

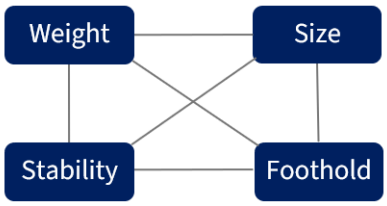

Fig. 5. Stacking preferences used for generating our custom dataset.

require quite a few engineering efforts even for as few as three boxes and becomes difficult to scale when the number of boxes to stack increases. In contrast, LLM is better at generating sequences involving unstacking and stacking for more boxes without extra costs.

2) Model size: For the pretrained LLM, we used the instruction-fine-tuned version of Mistral Small 3 Base model 2501 with 24 billion parameters. We use one Nvidia GeForce 4090 for the inference and fine-tuning of this model. However, in the early stage of this research, we experimented with a more lightweight version of this model with only 7 billion parameters, but found it is not good at processing mathematical tasks, such as sorting weights. Therefore, we argue that the selected model need to have sufficient mathematical reasoning capabilities while lightweight enough to allow fast inference for performing tasks efficiently.

B. Real-World Experiments

1) Robot platform: We deployed our proposed framework on a Kawada NEXTAGE humanoid robot equipped with a robotiq 140 gripper, wrist-mounted force-torque sensor, and two piezo microphones on each finger of the gripper, as shown in Fig. 1.

2) Object pose estimation: The 6D pose of each box is estimated by FoundationPose [10] and Segment Anything 2 (SAM2) [11]. FoundationPose estimates 6d pose from RGB, depth and masked images, where the masked images are obtained from SAM2. The object poses are then converted to the world frame for the motion controllers.

3) Motion controllers: After obtaining estimated object poses in the world frame, we use OpTaS (OPtimization-based TAsk Specification library) [12] to solve the inverse kinematics to control the gripper to reach each object.

4) Prompt tuning: Demonstrated with two or three prompt-response examples, the pretrained LLM can generate the correct stacking sequence in the desired format to be parsed to corresponding robot motions.

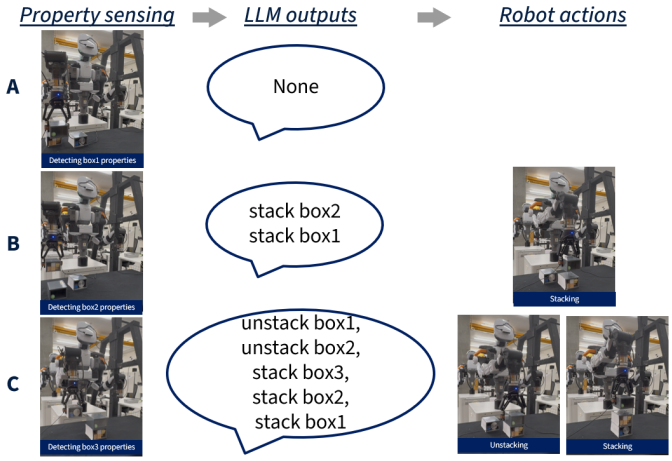

Fig. 6. The NEXTAGE robot performing long-horizon stacking task of three boxes with various contents in an online manner. Left: The NEXTAGE robot detecting weight and stability for each box. Middle: LLM generating corresponding stacking sequence on-the-fly after each detection. Right: The NEXTAGE robot executing stacking or unstacking according to LLM outputs.

5) Stacking with different preferences: Using the pretrained LLM as high-level planner, the NEXTAGE robot can perform successful stacking given different preferences including weight only, stability only, and weight and stability at the same time. Since offline planning is simple and intuitive, here we only showcase the online planning results. We have three boxes with same visual appearance but unknown hidden properties due to various contents inside (Fig. 1). As shown in Fig. 6, the NEXTAGE robot first detects the weight and stability of box 1. At this stage, since no other boxes have been manipulated and detected, LLM decides to wait for the properties of another box before reaching a decision. After getting the properties of box 2 and figuring out box 2 needs to be put under box 1, LLM generates the stacking sequence [stack box2, stack box1]. The robot performs this current best sequence for the boxes with known hidden properties. After detecting box 3 properties, LLM figures out that box 3 needs to be at the bottom and the existing stack needs to be adjusted, so it generates a new sequence to guide the robot to a series of unstacking and stacking to reach the ideal stack in the end.

VI. Conclusion

With very limited examples provided, the pretrained LLM considering multiple modalities works well as a high-level planner to reason over stacking sequence in both offline and online manner. The successful deployment of long-horizon stacking on a real robot guided by LLMs showcased the effectiveness of the proposed framework.

In future work, we plan to fine-tune multimodal LLMs with our custom dataset to enhance their reasoning capabilities, particularly in capturing physical properties for long-horizon manipulation tasks. Additionally, incorporating a broader range of user preferences and constraints would enable the model to handle more diverse stacking scenarios, extending its applicability to everyday settings, such as arranging box stacks to fit within the height limits of a cupboard.

REFERENCES

[1] J. Huang and K. C.-C. Chang, "Towards reasoning in large language models: A survey," *arXiv preprint arXiv:2212.10403*, 2022.

[2] M. Zawalski, W. Chen, K. Pertsch, O. Mees, C. Finn, and S. Levine, "Robotic control via embodied chain-of-thought reasoning," in *Conference on Robot Learning*, 2024.

[3] D. Honerkamp, M. Büchner, F. Despinoy, T. Welschehold, and A. Valada, "Language-grounded dynamic scene graphs for interactive object search with mobile manipulation," *IEEE Robotics and Automation Letters*, 2024.

[4] L. X. Shi, B. Ichter, M. Equi, L. Ke, K. Pertsch, Q. Vuong, J. Tanner, A. Walling, H. Wang, N. Fusai *et al.*, "Hi robot: Open-ended instruction following with hierarchical vision-language-action models," *arXiv preprint arXiv:2502.19417*, 2025.

[5] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[6] G. R. Team, S. Abeyruwan, J. Ainslie, J.-B. Alayrac, M. G. Arenas, T. Armstrong, A. Balakrishna, R. Baruch, M. Bauza, M. Blokzijl *et al.*, "Gemini robotics: Bringing ai into the physical world," *arXiv preprint arXiv:2503.20020*, 2025.

[7] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π0: A vision-language-action flow model for general robot control, 2024," *URL https://arxiv.org/abs/2410.24164*.

[8] M. Nazarczuk, J. K. Behrens, K. Stepanova, M. Hoffmann, and K. Mikolajczyk, "Closed loop interactive embodied reasoning for robot manipulation," *arXiv preprint arXiv:2404.15194*, 2024.

[9] A. X. Lee, C. M. Devin, Y. Zhou, T. Lampe, K. Bousmalis, J. T. Springenberg, A. Byravan, A. Abdolmaleki, N. Gileadi, D. Khosid *et al.*, "Beyond pick-and-place: Tackling robotic stacking of diverse shapes," in *5th Annual Conference on Robot Learning*, 2021.

[10] B. Wen, W. Yang, J. Kautz, and S. Birchfield, "Foundationpose: Unified 6d pose estimation and tracking of novel objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 868–17 879.

[11] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "Sam 2: Segment anything in images and videos," *arXiv preprint arXiv:2408.00714*, 2024. [Online]. Available: https://arxiv.org/abs/2408.00714

[12] C. E. Mower, J. Moura, N. Z. Behabadi, S. Vijayakumar, T. Vercauteren, and C. Bergeles, "Optas: An optimization-based task specification library for trajectory optimization and model predictive control," *arXiv preprint arXiv:2301.13512*, 2023.