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

Language-Conditioned Robotic Manipulation with Fast and Slow Thinking

Minjie Zhu[1,*], Yichen Zhu[2,*], Jinming Li[3], Junjie Wen[1], Zhiyuan Xu[2], Zhengping Che[2], Chaomin Shen[1,†], Yaxin Peng[3], Dong Liu[2], Feifei Feng[2], and Jian Tang[2,†]

***Abstract*— The language-conditioned robotic manipulation aims to transfer natural language instructions into executable actions, from simple "pick-and-place" to tasks requiring intent recognition and visual reasoning. Inspired by the dual-process theory in cognitive science—which suggests two parallel systems of fast and slow thinking in human decision-making—we introduce *Robotics with Fast and Slow Thinking (RFST)*, a framework that mimics human cognitive architecture to classify tasks and makes decisions on two systems based on instruction types. Our RFST consists of two key components: 1) an instruction discriminator to determine which system should be activated based on the current user's instruction, and 2) a slow-thinking system that is comprised of a fine-tuned vision-language model aligned with the policy networks, which allow the robot to recognize user's intention or perform reasoning tasks. To assess our methodology, we built a dataset featuring real-world trajectories, capturing actions ranging from spontaneous impulses to tasks requiring deliberate contemplation. Our results, both in simulation and real-world scenarios, confirm that our approach adeptly manages intricate tasks that demand intent recognition and reasoning.**

I. Introduction

Originally designed to generate robot actions based on language instructions and observations, robotic controls have demonstrated an expanding capability to handle a broader array of manipulation tasks beyond simple pick-and-place operations. Interestingly, at the heart of this manipulation model lies an auto-regressive mechanism for trajectory generation, which offers a direct mapping from an "instruction-observation" pair to an action space [1], [2]. Can such a straightforward mechanism truly be the foundation for a robot aiming to become a general agent, assisting humans in real-world scenarios? If it falls short, what issues might challenge this current approach, and what alternative mechanisms should be considered?

The literature concerning human cognition offers insights into these questions. Dual-process model research indicates that individuals engage with decisions in two primary ways: a rapid, instinctive, subconscious manner (referred to as "System 1 or Fast-thinking") and a measured, deliberate, conscious manner ("System 2 or Slow-thinking") [3], [4], [5], [6], [7]. Notably, these two systems have been associated with various mathematical models in machine learning. For instance, studies on reinforcement learning in both humans and animals have delved into conditions that prompt either associative "model-free" learning or the more contemplative "model-based" planning [8], [9], [10]. The straightforward associative command-action of the policy network bears similarities to "System 1". Therefore, it could be enhanced with a more intentional "System 2" planning approach. This would involve reasoning that (1) preserves and examines a range of options for present choices beyond the straightforward command like "pick-and-place objects" and (2) assesses its existing state, proactively forecasting or revisiting decisions for a more comprehensive perspective.

To design such a planning process, we draw inspiration from the human cognitive system, originated from Kahneman [7]. We propose a novel language-conditioned Robotic manipulation framework with Fast and Slow Thinking (RFST, in short), depending on the complexity of the user's language instruction. As Figure 1 illustrates, while existing methods simply output the robot's action via a policy network, we actively maintain a Think Bank, where each thought is divided into either a fast-thinking system or a slow-thinking system that serves as an intermediate step toward problem-solving. Such a high-level semantic unit allows the robot to self-evaluate the progress different thoughts make toward solving the problem through a deliberate reasoning process or an intuitive action. Finally, we combine this language-conditioned capability to perform manipulation tasks. We leverage different models for two types of systems. As System 1 only involves fast and straightforward thinking, we allow a simple, shallow policy network to do the jobs. For difficult tasks that need reasoning or planning, we opt for a Vision-Language Model (VLM). This model is designed to either break down the tasks into manageable sub-tasks or clarify the user's intent. Subsequently, a policy network outputs action based on these augmented instructions.

Empirically, we validate the efficacy of RFST across a spectrum of tasks, ranging from basic pick-and-place and rotation to more intricate tasks such as mathematical and visual reasoning. While traditional robotic manipulation methods can address the former tasks, the latter requires systematic planning or an in-depth search for the true user intention, challenges that direct policy networks often struggle with. Our results indicate that RFST delivers superior performance on complex tasks in simulated benchmarks. Moreover, we have curated a dataset featuring real-world trajectories span-

[1]School of Computer Science, East China Normal University, China {51255901028, 51255901019}@stu.ecnu.edu.cn, cmshen@cs.ecnu.edu.cn
[2]Midea Group, China {zhuyc25, xuzy70, chezp, liudong13, feifei.feng, tangjian22}@midea.com
[3]Department of Mathematics, School of Science, Shanghai University, China {ljm2022, yaxin.peng}@shu.edu.cn
*Equal contributions. This work was done during Minjie Zhu, Jinming Li, and Junjie Wen's internship at Midea Group.
†Corresponding authors: Chaomin Shen and Jian Tang.

ning nine tasks: three for fast-thinking systems and six for slow-thinking systems. Our data reveals that RFST can adeptly tackle tasks from both categories.

To sum up, our contributions are threefold:

- We present a fast and slow thinking framework for robotics manipulation that categorizes incoming instruction into two systems and performs control correspondingly.
- We design a framework for slow thinking, which leverages the fine-tuned VLM to perform instruction-observation conditioned reasoning and re-write the instruction for robotics affordance.
- We collect a set of real-world datasets, including tasks like math reasoning and intent recognition, and examine the effectiveness of our approach on both simulation and real-world scenarios.

II. Related Work

Reasoning in Language and Vision. Chain-of-thought (CoT) [11] use a coherent language sequence that served as a meaningful intermediate step toward problem-solving of mapping input questions with output language. Self-consistency with CoT [12] ensembles CoT and prioritizes the most frequent output. Tree-of-Thought [13] uses a tree structure to classify input questions into different sub-trees for the final answer. Least-to-most prompting [14] breaks down a complex problem into a series of simpler subproblems and then solves them in sequence. Program-of-Thought [15] translates natural language into program format, assisting LLM in mathematical reasoning. Collectively, these "X-of-Thought" methodologies empower LLM to engage in chats that demand reasoning.

A recent trend in vision-language models [16], [17], [18], [19], [20], [21] allows for the comprehension of images and the provision of answers in natural language, albeit with constrained reasoning capabilities. These reasoning skills are realized by integrating large language models [22] with a vision backbone. Instead, we establish a framework that discerns the "amount of minds" needed to process the instruction.

Large Language Model for Robotics. Large language models possess the power of reasoning. With the advancement of LLM in the past year, a rising number of projects have been proposed to use LLM as a high-level model for task planing [23], code generation [24], navigation [25], [26], manipulation [27], and action correction [28]. RT-2 [1] introduced an end-to-end model capable of processing tasks with text and one or more images, producing a sequence of tokens to control a robot. When integrated with expansive vision-language models like PaLI-X [29] and PaLM-E [30], RT-2 demonstrates reasoning within this framework. While the end-to-end method is indeed appealing, nevertheless, these networks typically demand vast amounts of training data and come with considerable computational costs. In contrast, our proposed RFST allows reasoning, symbolic understanding, and intent recognition. It maintains the delicate balance of high-level reasoning with low-level control models. Furthermore, RFST requires significantly less training data, making it advantageous when paired with extensive datasets.

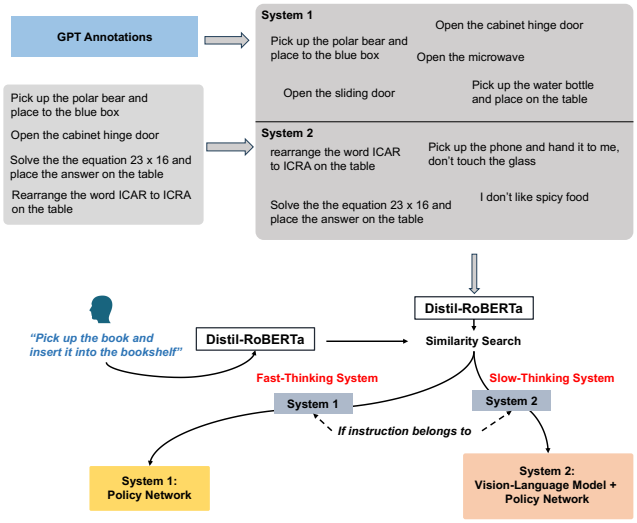

Fig. 1: The overview of RFST. We collected a number of instructions and employed GPT4 [31] for annotation. Upon receiving an instruction, the robot processes it through Distil-RoBERTa to obtain an embedding. Leveraging embedding similarity search, we classified the instruction into either a fast-thinking system or a slow-thinking system.

III. Methodology

In this section, we provide a detailed description of RFST. We first give a formal definition of fast and slow thinking. Then, we introduce the overall framework of RFST and present our slow-thinking system.

A. Formal Definition of Fast Thinking and Slow Thinking

Given a language instruction x, the policy network is a mapping function to get an output y. The complexity of the mapping function p_θ is determined by the x. For a simple instruction, e.g., pick up an apple, the mapping function could be simple $y \sim p_\theta(x)$. We consider these tasks as fast-thinking tasks. When the mapping of input x to output y is non-trivial (i.e., when x is a math question and y is the numerical answer), we need to introduce an intermediate step z to bridge x and y. Then, the mapping function is $y \sim p_\theta(x|z)$. Our task is to classify the given instruction as either a fast-thinking system or a slow-thinking system. Notice that the fast-thinking system can be arbitrary language-conditioned robot manipulation algorithm that has been well developed over the past year, e.g., GATO [32], VIMA [33], RT-1 [1]. Therefore, in the second part, we focus on introducing our slow-thinking system, which we need to design the z for correct manipulation meticulously.

B. Overall Framework of RFST

To determine whether an incoming language instruction corresponds to System 1 or System 2, we have established an instruction bank comprising many language instructions.

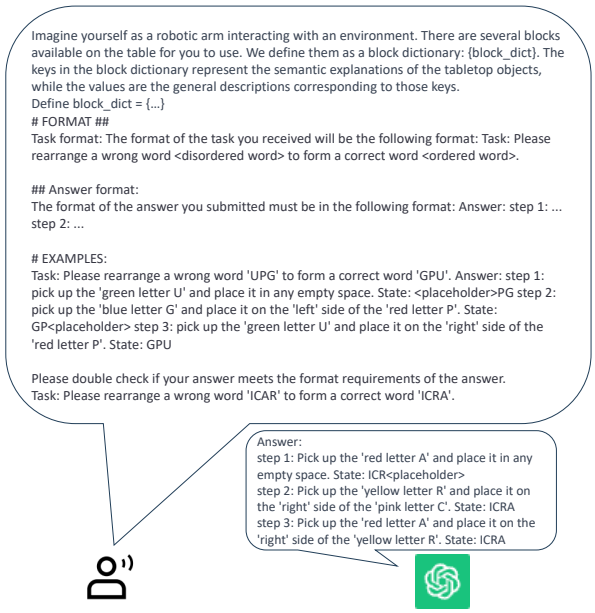

Fig. 2: An illustrative example of step-by-step task planning originates from GPT-3.5-turbo. The planning produced by the LLM serves as the foundation for formulating our text-image pairs used for VLM training.

We employ GPT-4 [31] to simulate a robot. Given specific scenarios, we prompt GPT-4 to produce a list of language instructions and specify their association with either System 1 or System 2. Using this curated set of categorized instructions as our foundational seed, we enable GPT-4 to augment these instructions, reshaping them into diverse formats while maintaining consistent meaning. After ten iterations, this process yields thousands of pre-classified language instructions. Additionally, we go through a manual review of the generated instructions for accuracy. We give an overview in Figure 1.

Because the instruction is entirely generated by LLM, it is obvious that LLM can be used to decide which category the user utterances. However, due to the significant computational demands of LLM, we have been motivated to seek a more lightweight approach. Toward this goal, we encode the language and undertake instruction retrieval. The utterance is embedded using a frozen version of the DistilRoBERTa [34], [35] language model, as provided by the Sentence-BERT [36] project. Supported by an "unnatural language processing" nearest neighbors index, inference-time utterances are matched with the closest training exemplars. These exemplars are then retrieved and processed by the model. The classification of a given instruction is determined based on the category of the retrieved instruction. Empirically, we used 500 instructions from GPT-4 to form our "Think Bank" and do instruction classification at test time. In our experiments, this approach can perfectly classify language instruction.

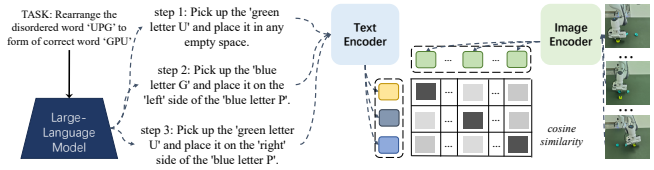

Fig. 3: An illustration of CLIP computing the similarity between step-wise text description and observations.

C. Details of Slow Thinking

We illustrate the details of our slow-thinking mode. There are two key factors in System 2: 1) a vision-language model that could perform reasoning and intent recognition, given the language instruction and current observation, and 2) a policy network that can understand the planning from the vision-language model to act precisely.

Empower Vision-Language Model with Reasoning and Intent Recognition. The vision-language models we utilize in this study accept a text-image pair as input and yield a sequence of tokens, typically representing natural language text. These models are versatile, capable of a broad spectrum of visual interpretation tasks—from deciphering an image's composition to responding to queries about individual objects and their relationships. However, standard pre-trained vision-language models lack an understanding of the physical world. Our objective is a vision-language model that not only grasps the relationship between observed scenes and natural language, but can also recognize the user's intention and provide logical, step-by-step instructions to guide a robot's actions. To realize this, we require a dataset featuring instruction-observation pairs and the finetune a VLM.

Multi-modal Planning Data Collection. We demonstrate how to build up our multi-modal instruction data. First of all, we need to delineate tasks step-by-step, aligned with the user's intention. To achieve this, we seek the help of LLM. We first convert the scene into natural language to ensure the LLM comprehends it effectively. We use the pre-trained vision-language model, i.e., BLIP-2 [16], to do the image caption. Then, for each set of tasks, such as math reasoning, grammar check, and user's intention understanding, we draft a prompt script. This script incorporates in-context learning and a chain-of-thought approach, enabling the LLM to yield our anticipated planning or clarify user intentions. We give an example of the prompt for instruction generation of word reordering is present in Figure 2. After gathering all the data, manual verification was conducted. We also include a number of texts that recognize the user's intention and transfer them into actionable instructions for the robot. All these text scripts are used for training only, and they are generated by GPT-4. Empirically, we found the majority of responses from GPT-4 were accurate.

Mapping Sub-goal with Observations. For tasks requiring step-by-step planning, a conventional approach involves using these steps as instructions and subsequently grounding them into robotic actions. Yet, to ensure the robotic agent thoroughly understands the instructions, it's crucial

to synchronize the instruction with the observation for that specific step. We advocate the use of CLIP [18] to bridge visual inputs with text descriptions. By computing the dot product between the text and image embedding vectors, we pair a text and image if the result surpasses a threshold value, denoted as α. In our implementation, α is set to 0.75. Furthermore, we fine-tune CLIP using a limited dataset from the scene, which we have labeled manually. A brief illustration can be found in Figure 3. To ensure accuracy, we manually inspect the data post-processing. Unlike the planning and user intention understanding derived from GPT-4 in the preceding step, manual verification is vital since CLIP's accuracy can waver if observations between two consecutive steps aren't sufficiently distinct.

Vision-Language Model Architecture. We employed the pre-trained ViT-L/14 from CLIP as our visual encoder, paired with the LLaMA-2-7B as our LLM [18], [22], [37]. To maintain modal alignment and facilitate a compatible input dimension for the LLM, a fully connected layer has been integrated. This layer transforms the ViT's output embedding 16×16 output embedding $V \in \mathbb{R}^{16\times16\times1024}$ to $V^{'} \in \mathbb{R}^{256\times4096}$. We tap into the robust vision-language capabilities inherent to the text-image alignment [38]. Moreover, we fine-tune the associated networks, holding the language and visual embeddings constant. Only the alignment layers are subject to adjustments.

Policy Networks with Language Instruction. To craft an efficient multi-task robotic policy, we utilize policy networks featuring a multi-task decoder architecture. Specifically, our goal is to derive a robotic policy represented by $\pi(a_t|P, H)$, where $H := \{o_1, a_1, o_2, a_2, \cdots, o_t\}$ encapsulates the trajectory of historical interactions. The $o_t \in \mathbb{O}$ and $a_t \in \mathbb{A}$ denote the observations and actions at each interaction step, respectively. These policy networks are designed to handle multi-modal tokens, and for their encoding, we incorporate multi-modal prompts. The images are processed via a vision backbone (ResNet-50 [39], [40]) while the text is tokenized. The image embedding and text embedding are connected with FiLM [41]. The policy network is followed up, and it consists of three MLP layers with ReLU activation.

IV. A Dataset of Two Systems

To study our pre-training approach, we collect a large dataset of real-world robot trajectories. We collect different tasks that belong to different systems. In this section, we describe our data collection process and show qualitative examples from our dataset.

Hardware. We use the Franka robot with a 7-degree-of-freedom arm, which is equipped with a parallel jaw gripper (see Figure 4, top). Proprioceptive data, including joint positions and velocities, are recorded throughout our experiments. Actions in the joint space are determined by the differences between successive states. Our workspace boasts two high-quality D435i RealSense RGBD cameras. We only use the RGB information in our experiments. One egocentric camera is attached to the robot's hand, and one exocentric camera is positioned at the robot's front.

Math Reasoning [Slow-Thinking]. Our objective is to engage the robot in mathematical reasoning tasks, including equation solving. We present two sets of tasks. The first requires the robot to directly compute the mathematical expression presented on a table. The second involves solving for an unknown variable x. For example, when presented with an image displaying $11 \times 13 =$, or $1 + x = 6$, the robot is tasked with completing the equation or substituting x with the correct number. These tasks are generally single-step challenges. Their success hinges on the vision-language model's capability to comprehend the mathematical reasoning within the scene.

Word Correction [Slow-Thinking]. The robot is responsible for correcting word spellings, be it due to incorrect sequences or specific word designations. These tasks can range from simple single-step actions to more intricate multi-step processes. Take, for instance, the task of rearranging the word "ICAR" to form "ICRA". This task demands three distinct steps: 1) pick up the letter "A" and place it in the empty space, 2) pick up the letter "R" and place it next to the letter "C", 3) pick up the letter "A" and place next to the letter "R". This kind of task not only tests the robot's linguistic aptitude but also its dexterity and ability to perform sequential operations accurately. The combination of language and motor skills is paramount to execute such tasks efficiently.

Sort Cube by Color [Slow-Thinking]. We've arranged several cubes on the table, each coming in one of four distinct colors. The robot's task is to identify individual cubes, grasp them, and then group them with other cubes of the same color. The complexity of the task stems not just from the robot's ability to recognize colors but also from its spatial reasoning in determining where to place each cube to create organized color clusters. This challenges the robot's visual processing capabilities and its precision in handling objects.

Intent Recognition [Slow-Thinking]. We've designed several tasks that necessitate visual reasoning. Consider a scenario where an image depicts various foods on a table. If a user provides the instruction, "I'm allergic to spicy food," the robot would identify spicy items, such as chili or other spicy ingredients, and relocate them to the opposite side of the table. This exemplifies a typical situation in which robots must discern the user's intent based on verbal directives.

Pick Cube based on Color [Fast-Thinking]. The robot's assignment is to grasp a cube according to the color information from language instruction.

Pick Cube and Place to left/right box[Fast-Thinking]. The robot is asked to select a cube by color and put it into a box, either on the left side or right side, based on the instructions.

Pick Toy and Place to box [Fast-Thinking]. The robot is asked to pick up a toy put into a box.

Statistics for Data Collection. We collect approximately 2,000 real-world trajectories, where the average length of the trajectories is around 100. The dataset contains variations in object poses, shape, and appearance. Objects are randomly placed on the table. We give multiple examples for our aforementioned tasks in Figure 4.

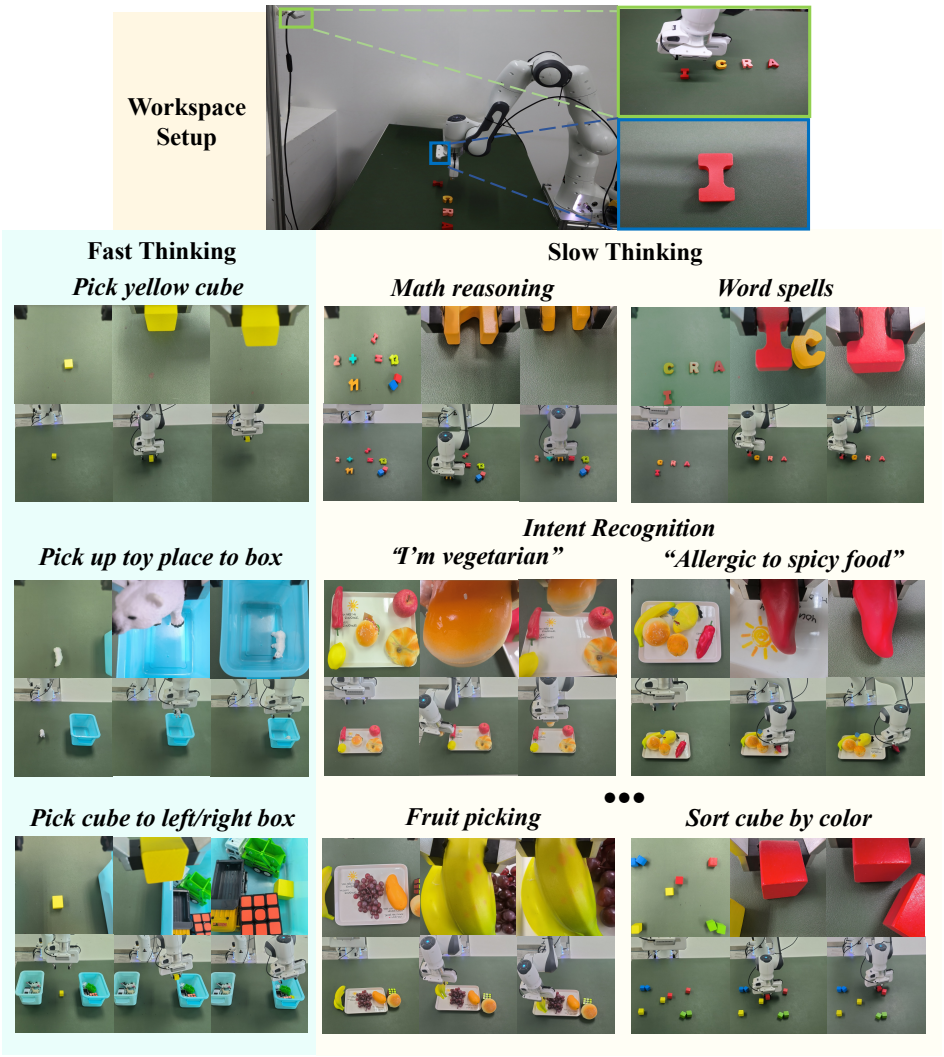

Fig. 4: We collect a dataset with real-world trajectories using a Franka robotic arm. Each trajectory is a sequence of images from two cameras. We consider multiple tasks that belong to either the fast-thinking system or the slow-thinking system.

V. Experiments

In this section, we empirically assess the broad applicability of RFST across diverse tasks in both simulated and real-world settings.

A. Simulation Experiments

Experiments Setup. We conduct our simulation experiments on VIMA-Bench [33], built on the Ravens simulator [42]. We chose VIMA-Bench because this benchmark consists of tasks that require reasoning and multi-step manipulation. We train the policy network on all six tasks. We select two tasks as fast-thinking tasks: rotation and simple object manipulation. In the "Rotation" task, the robot is instructed to rotate objects clockwise by specific degrees along the z-axis. The "Simple Object Manipulation" task involves placing one object inside another. Both these tasks are executed in a single step.

For more deliberate (slow-thinking) tasks, we selected four distinct assignments. The "Rearrange the Scene" task provides a description of the desired scene, instructing the robot to rearrange objects accordingly. "Visual Reasoning" requires stacking multiple objects in a specified square sequence. The "Stacking Multiple Objects" task delineates the stacking order, allowing the model to determine the sequence of object placement. Finally, the "Stack the Same Texture" task demands the model to identify and stack objects that share identical textures. A visual representation of these six tasks can be found in Figure 5. In our experiments, we use Task 1 and Task 2 to represent the first two fast-thinking tasks and Task 3-6 to denote the latter four slow-thinking tasks. For those slow-thinking tasks, we follow our proposed pipeline to use a trained vision-language model to give a step-by-step plan. and then feed it into our policy networks. Please note that in the standard VIMA-Bench, the instruction comprises both image and text components. For images that depict the entire scene, we employ LLaVA-13B [17] to describe the scene and manually correct any errors. For images representing specific objects, we utilize the look-up table from VIMA-Bench to convert them into

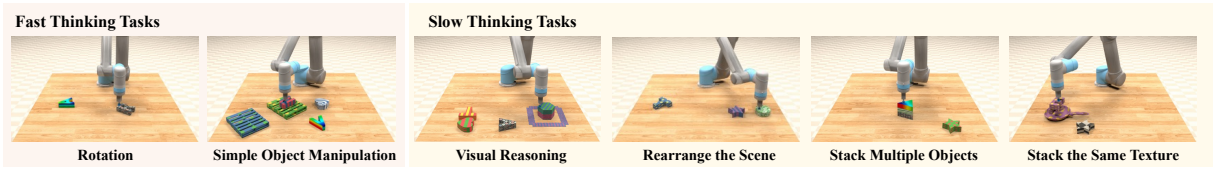

Fig. 5: Example of tasks in simulation. We select six tasks in VIMA-Bench [33] and categorize them into fast-thinking and slow-thinking tasks accordingly.

TABLE I: Success rates on VIMA-Bench over six tasks. The Tasks 1 and 2 belong to fast-thinking system, and Task 3-6 belong to slow-thinking system. Our proposed RFST significantly outperforms other methods in accomplishing slow-thinking tasks, achieving notably higher success rates.

Method	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Gato	37	50	11	18	22	16
Flamingo	39	37	15	25	34	14
VIMA	58	**52**	27	17	31	26
RFST (Ours)	**60**	49	**36**	**47**	**42**	**35**

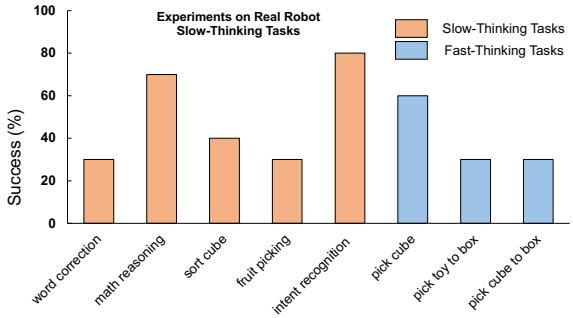

Fig. 6: The experiments on the real robot. **Orange Bars**: Slow-thinking tasks. **Blue Bars**: Fast-thinking tasks. RFST empowers real robots to execute complex tasks such as mathematical reasoning and intent recognition, which were traditionally beyond the scope of conventional robotic manipulation techniques.

text. We train our model with 2M parameters. The number of training trajectories for each task is 1,000. We mixed rotation and pick-and-place task data to train a fast-thinking policy network. Each task is evaluated with 20 trials, where objects are randomly placed.

Baselines. We compare our model with GATO [32], Flamingo [43], and VIMA [33]. Gato is a decoder-only generalist agent. Flamingo is a state-of-the-art vision-language model. VIMA is a multi-task robotic manipulation model that receives multi-modal prompts. We follow the publicly released code in VIMA to implement their methods. For fair comparisons, we use text-only prompts in all experiments. All methods are trained on the same amount of data.

Main Experimental Results. Table I presents the experimental results. It can be observed that our model does exhibit superiority in fast-thinking tasks. For example, while we surpass VIMA by 2% on Task 1, we lag behind by 3% on Task 2. However, upon examining slow-thinking tasks, our approach demonstrates notably superior performance compared to the baseline. These outcomes underscore the importance of devising an appropriate intermediate step z for addressing complex tasks. Such tasks may require capabilities like reasoning and symbolic understanding.

B. Real-world Experiments

We conduct experiments on real-world robotics manipulation tasks to give a full comprehension of RFST on intent recognition, reasoning, and symbolic understanding. The real-world experiments are more challenging due to imperfect camera sensors and increased object quantity and diversity. For each task, we conduct ten trials, and the objects are randomly placed on the table.

Experimental Results. Figure 6 demonstrates the experimental results from the real-world experiments. On the right are the tasks that involve slow thinking. Our proposed RFST achieves good performance, especially on two reasoning tasks: math reasoning and intent recognition. The relatively low success rate of word correction is due to the size of a word being too small and non-regular, which makes it hard for the gripper to grasp it successfully. It is worth noting that our framework succeeded in intent recognition on eight out of ten trials, underscoring its exceptional capability in handling intricate tasks demanding human-like cognition.

VI. Conclusion

We introduce an approach to robotic manipulation that seamlessly addresses both straightforward tasks and intricate tasks demanding visual reasoning, all within a unified framework. Our strategy emphasize the dual system humans employ: “System 1” for rapid, intuitive actions and “System 2” for more deliberate, contemplative thinking. To operationalize this, we have crafted tools that encompass instruction classification — determining which system an incoming instruction pertains to — and refining a vision-language model for visual reasoning. This aids policy networks in executing multi-step manipulations. Our method’s efficacy is demonstrated in simulations requiring multi-step control and actual robots executing a gamut of tasks.

Acknowledgment

This work was supported by the Large-scale Numerical Simulation Computing Sharing Platform of Shanghai University, and the Shanghai Science and Technology Innovation Action Plan under Grant 22511105400.

REFERENCES

[1] 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.

[2] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "BC-Z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[3] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[4] K. E. Stanovich, *Who is Rational? Studies of Individual Differences in Reasoning*. Psychology Press, 1999.

[5] D. Kahneman, S. Frederick *et al.*, "Representativeness revisited: Attribute substitution in intuitive judgment," *Heuristics and biases: The psychology of intuitive judgment*, vol. 49, no. 49-81, p. 74, 2002.

[6] S. A. Sloman, "The empirical case for two systems of reasoning." *Psychological Bulletin*, vol. 119, no. 1, p. 3, 1996.

[7] D. Kahneman, *Thinking, Fast and Slow*. Macmillan, 2011.

[8] B. Schölkopf, F. Locatello, S. Bauer, N. R. Ke, N. Kalchbrenner, A. Goyal, and Y. Bengio, "Toward causal representation learning," *Proceedings of the IEEE*, vol. 109, no. 5, pp. 612–634, 2021.

[9] S. Bubeck, V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg *et al.*, "Sparks of artificial general intelligence: Early experiments with GPT-4," *arXiv preprint arXiv:2303.12712*, 2023.

[10] T. M. Moerland, J. Broekens, A. Plaat, C. M. Jonker *et al.*, "Model-based reinforcement learning: A survey," *Foundations and Trends in Machine Learning*, vol. 16, no. 1, pp. 1–118, 2023.

[11] J. Wei, X. Wang, D. Schuurmans *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24 824–24 837.

[12] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," *arXiv preprint arXiv:2203.11171*, 2022.

[13] S. Yao, D. Yu, J. Zhao *et al.*, "Tree of thoughts: Deliberate problem solving with large language models," *arXiv preprint arXiv:2305.10601*, 2023.

[14] D. Zhou, N. Schärli, L. Hou *et al.*, "Least-to-most prompting enables complex reasoning in large language models," *arXiv preprint arXiv:2205.10625*, 2022.

[15] W. Chen, X. Ma, X. Wang, and W. W. Cohen, "Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks," *arXiv preprint arXiv:2211.12588*, 2022.

[16] J. Li, D. Li, S. Savarese, and S. Hoi, "BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," *arXiv preprint arXiv:2301.12597*, 2023.

[17] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023.

[18] 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*, 2021, pp. 8748–8763.

[19] M. Shridhar, L. Manuelli, and D. Fox, "CLIPort: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.

[20] Y. Zhu, M. Zhu, N. Liu *et al.*, "Llava-phi: Efficient multi-modal assistant with small language model," *arXiv preprint arXiv:2401.02330*, 2024.

[21] X. Liu, Y. Zhu, Y. Lan, C. Yang, and Y. Qiao, "Query-relevant images jailbreak large multi-modal models," *arXiv preprint arXiv:2311.17600*, 2023.

[22] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[23] L. Fan, G. Wang, Y. Jiang, A. Mandlekar, Y. Yang, H. Zhu, A. Tang, D.-A. Huang, Y. Zhu, and A. Anandkumar, "MineDojo: Building open-ended embodied agents with internet-scale knowledge," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 18 343–18 362.

[24] 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.

[25] Z. Mandi, S. Jain, and S. Song, "RoCo: Dialectic multi-robot collaboration with large language models," *arXiv preprint arXiv:2307.04738*, 2023.

[26] 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)*, 2023, pp. 10 608–10 615.

[27] J. Wen, Y. Zhu, M. Zhu *et al.*, "Object-centric instruction augmentation for robotic manipulation," *arXiv preprint arXiv:2401.02814*, 2024.

[28] Y. Cui, S. Karamcheti, R. Palleti, N. Shivakumar, P. Liang, and D. Sadigh, "No, to the right: Online language corrections for robotic manipulation via shared autonomy," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, 2023, pp. 93–101.

[29] X. Chen, J. Djolonga, P. Padlewski, B. Mustafa, S. Changpinyo, J. Wu, C. R. Ruiz, S. Goodman, X. Wang, Y. Tay *et al.*, "PaLI-x: On scaling up a multilingual vision and language model," *arXiv preprint arXiv:2305.18565*, 2023.

[30] 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.

[31] OpenAI, "GPT-4 Technical Report," *arXiv preprint arXiv:2303.08774*, 2023.

[32] S. Reed, K. Zolna, E. Parisotto, S. G. Colmenarejo, A. Novikov, G. Barth-Maron, M. Gimenez, Y. Sulsky, J. Kay, J. T. Springenberg *et al.*, "A generalist agent," *arXiv preprint arXiv:2205.06175*, 2022.

[33] Y. Jiang, A. Gupta, Z. Zhang, G. Wang, Y. Dou, Y. Chen, L. Fei-Fei, A. Anandkumar, Y. Zhu, and L. Fan, "VIMA: General robot manipulation with multimodal prompts," *arXiv preprint arXiv:2210.03094*, 2022.

[34] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[35] Y. Liu, M. Ott, N. Goyal *et al.*, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[36] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[37] 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.

[38] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.

[39] S. Nair, A. Rajeswaran, V. Kumar, C. Finn, and A. Gupta, "R3m: A universal visual representation for robot manipulation," *arXiv preprint arXiv:2203.12601*, 2022.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[41] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[42] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.

[43] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 23 716–23 736.