A Universal Embodied Artificial Intelligent Scheme Combined Large Language Model with Diffusion Policy for Robot Manipulation Tasks

Yue Zhao*,1,Xiaozhu Ju*,1,Zhihang Li*,2, Xuelian Geng[2] and Zizhuang Guo[3]

https://yue-zhao-robot.github.io/Embodied-AI-with-DP-and-LLM/

Abstract— The development of large language model (LLM) and visual language model(VLM) has driven rapid progress in embodied artificial intelligence (Embodied AI) technology, opening up new avenues for robots to interact with the physical world and providing vast space for robots to perform more complex manipulation tasks. The diffusion policy does not rely on the robot dynamics model and can elegantly handle multimodal action distributions, it's suitable for high-dimensional action spaces, and exhibits impressive training stability, making it highly suitable for robot precise control task. However, the diffusion policy cannot interact with the physical world and is difficult to guarantee the generalization for different task scenarios. To address these issues, this article proposes an Embodied AI scheme based on the LLM and the diffusion policy. This solution firstly converts human natural language instructions into robot manipulation subtasks by LLM and then generates robot actions based on the diffusion policy. Compared to existing Embodied AI solutions, the novelty of our method lies in two aspects: even when facing complex human instructions, it only needs to call the LLM once, reducing the time consumption of frequently calling the LLM. And, we increase the accuracy of task execution by suppressing the states of objects that we are not interested in while enhancing the states of task objects to guide the specific behavioral actions of robots. To validate our idea, we conducted simulation experiments on a 7 dof Franka Emika Panda using Isaac Sim. The experimental results show the correctness and generalization of our method.

Index Terms— Large language model, visual language model, embodied artificial intelligence, diffusion policy, Franka Emika Panda.

I. INTRODUCTION

In recent years, large language model(LLM) and visual language model(VLM) have been greatly applied in the field of robot manipulation. Embodied Artificial Intelligence(Embodied AI) [1], [2] refers to an intelligent agent that has a body and supports physical interaction. Under Embodied AI, artificial intelligence has a body and robots have a brain. The current technical roadmap for Embodied AI can be divided into two coarse-grained approaches. One major approach is to use multimodal VLM and traditional control algorithms to achieve multimodal perception and decision-making for robots [3], [4], [5], [6], [7], enabling robots to handle complex and unfamiliar scenes. Then, traditional control algorithms can be called through API interfaces to achieve motion control for robots. Another major approach is to attempt to build an integrated visual language action(VLA) model [8], [9], [10], [11] that directly maps perception signals to robot operation and control instructions, skipping the complex signal conversion process and achieving a closed-loop of perception decision action. Although the latter approach is more direct in controlling robots, it is currently not mature. The former utilizes both the generation capability of large models and the advantages of stable and efficient traditional control, making it the mainstream technology solution for embodied intelligence.

In previous robot operations, we needed to pre-define trajectories, which made robots more limited. Diffusion Policy [12], [13] is a novel robot trajectories [14], [15], [14], [16] or/and actions [17] generation method that represents the visual trajectories or/and actions policy of a robot as a conditional denoising diffusion process. The advantage of using the diffusion policy to plan robot trajectories or/and actions is that the diffusion policy can sample from complex arbitrary distributions. The distribution of robot trajectories or/and actions sequences is usually complex and multimodal, meaning that robots can typically complete a task in multiple ways. Using a universal diffusion policy to train robot can enable robots to learn multimodal behavior, making robot training more stable.

However, diffusion policy cannot interact with the physical world, which limits the generality of robots. For example, it doesn't know how to execute higher priority tasks, nor does it know how to execute flexible and variable tasks. The excellent response of LLM such as ChatGPT 4 makes us feel that it is possible for robots to become universal robots. We can clearly tell LLM the priority of executing tasks, or we can tell LLM which specific task to execute in complex scenarios. The LLM can reason and provide useful steps for robots. Due to the generality of LLM and diffusion policy, the proposed scheme in this paper has strong generalization ability. In theory, robots can interact with the real world

[1]Y. Zhao and X. Ju are with the Innovation Center of Beijing Embodied Artificial Intelligence Robot, Beijing 101111, China. Y. Zhao and X. Ju are the Embodied Intelligence researchers. (devin.zhao@x-humanoid.com; devin.zhao@x-humanoid.com)

[2]Z. Li and X. Geng are the interns of the Innovation Center of Beijing Humanoid Robot, they are currently working toward the Master's degree at Beijing Jiaotong University, Beijing, 100091, China. (22120882@bjtu.edu.cn; 22120368@bjtu.edu.cn)

[3]Z. Guo is the interns of the Innovation Center of Beijing Humanoid Robot, he is currently working toward the Master's degree at Johns Hopkins University, Baltimore, Maryland, MD 21218-2683, America. (e-mail: Zguo61@jh.edu)

through natural language. Through this method, robots can theoretically complete any operational task.

Although there are currently some Embodied AI achievements [18], [19], [20], there are still some issues at present. For example, the process of frequently calling LLM is very time-consuming, and the quality of the entire Embodied AI solution heavily depends on the ability of the VLM. To address these problems, this paper combines the advantages of LLM and diffusion policy to convert complex human natural language instructions into specific robot action. We use LLM to perform top-level tasks by designing prompts. By calling LLM once, we can achieve the transformation from human natural language instructions to robot executable tasks. We also reduce the dependence on VLM during task execution by suppressing the states of objects that we are not interested in while enhancing the states of task objects, indirectly improving the success rate of task execution.

Overall, this paper makes the following main contributions:

1) We propose an end-to-end solution for robot manipulation tasks, which utilizes the understanding ability of large language models for human instructions and the stable and efficient control advantages of diffusion policies.
2) Our method utilizes a large language model for top-level task planning, which only needs to be called once even in complex task scenarios.
3) We guide the specific behavioral actions of robots by suppressing the states of objects that we are not interested in while enhancing the states of task objects, which emphasizing the accuracy of task execution.

II. Method

A. Overall Framework

In Figure 1, the scene image is acquired by an RGB-D camera. Through the analysis of the open semantic visual language model, the complete scene state S can be obtained, which contains key information such as the position and speed of all scene objects. Another input is the task description L. The large language model that serves as the top-level planner will first parse the instructions and then break down task L into a series of coherent subtasks l_i $(i = 1, 2, 3...N)$, (N is the total number of subtasks) that conform to real logic. For each subtask, after parsing the subtask, the large language model will clarify the object of interest f_i of the current task and the background object b_i that needs to be ignored, and generate the corresponding python code, which then interacts with the original scene state S output by the visual language model, thereby emphasizing the object of interest f_i and suppressing the background object b_i, and finally obtaining the sub-state s_i $(i = 1, 2, 3...N)$ after conditional influence is applied to each subtask l_i. Finally, each sub-state s_i will be used as the diffusion condition of the diffusion model and after T steps of diffusion, the trajectory τ_i is obtained. When we reach the end point along τ_i $(i = 1, 2, 3...N)$ in sequence, task L is completed. In summary, it is a robot operation trajectory generation problem, which is formulated as follows:

$$\sum_{i=0}^{N} \tau_i = \sum_{i=0}^{N} Diffuser_i^T(S_i(l_i, f_i, b_i), \varepsilon_i) \tag{1}$$

where $\sum_{i=0}^{N} \tau_i$ is the trajectory corresponding to task L, S_i is the original scene state corresponding to subtask l_i and satisfies $\tau_i \in S_i$, ε_i is the pure noise obtained by gaussian sampling $\varepsilon_i \sim N(0, I)$, $Diffuser_i^T$ is the pre-trained diffusion model corresponding to subtask l_i.

B. Design of top-level planner

Enabling robots to understand human language and achieve human-robot interaction has always been a hot research area. However, it is almost impossible to directly transmit human language to robots to command their behavior. Traditional rule-based methods rely on a series of predefined rules to guide robots on how to parse and understand human language. However, these methods are limited in dealing with the complexity and diversity of natural languages, because natural languages often do not conform to strict rules and have a large number of exceptions and irregularities. The emergence of large language models provides new ideas. Thanks to their amazing natural language understanding and conversion capabilities, coupled with their python code generation capabilities, they provide new methods for robots to understand human instructions.

Using a large language model to parse task instructions and decompose them into multiple subtasks, and then independently processing each subtask, can greatly reduce the complexity of interaction. However, frequent calls to the large language model take up too much time, so we only call the large language model once when the instruction is passed in to decompose the task [21]. For example, when the instruction L "Put the bread on the plate and then put the beef on the bread" is input, it will be parsed according to the prompt as: l1) pick up the bread, move it above the plate and open the gripper, l2) pick up the beef and move it above the bread and open the gripper. Then, for different subtasks, the target object or target area is retrieved. For l1), the objects of interest are "bread" and "plate", where "bread" is the object of action, which we call cause, and "plate" is the object of action, which we call effect. In l2), "beef" is the cause and "bread" is the effect. Subsequently, the object of interest is input into OWL-vit [22] as an index to obtain its corresponding bounding box, and then the corresponding image is cropped and passed to SAM2

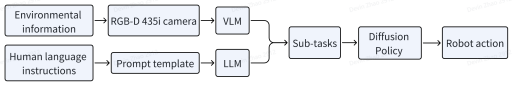

Fig. 1. The overall framework of embodied artificial intelligent scheme.

to obtain its mask. Finally, the mask is mapped to the point cloud obtained by the RGB-D camera to obtain the position and size information of the object of interest, and then the scene state information is updated to obtain the sub-state s_i corresponding to the sub-task l_i.

In this paper, only using the large language model to decompose tasks can significantly reduce the calling time and increase the model stability. The trajectory planning and control strategy generation of subsequent subtasks are achieved through the trained diffusion model.

C. Design of Action Generator

The Action Generator is get by training a diffusion model. The working mechanism of training a diffusion model is divided into two stages. Firstly, noise is introduced into the dataset, which is a fundamental step in the forward diffusion process, and then systematically reverse this process. The diffusion model is trained by a temporal U-Net [23], [24]. The Denoising Diffusion Probabilistic Model [25] (DDPM) is a generative model designed to fit a given target distribution so that samples can be taken from it. It uses Markov chain modeling, with noise as input, gradually denoised through neural networks, and finally produces an output that conforms to the target distribution. The generated data can be arbitrary, including images, speech, robot trajectories, and actions.The execution of robot manipulation is currently a shortcoming, which limits the generality of robot operation. Diffusion Policy has shown excellent performance and great potential in solving the problem of agile robot operation execution.

The predefined forward diffusion process involves adding noise to a variable step by step until the variable becomes pure noise. The form and parameters of this process are manually defined. The forward process is a predefined Markov chain, which is defied as:

$$q(x^k|x^{k-1}) = N(\sqrt{1-\beta_k}x^k, \beta_k I) \tag{2}$$

where $q_\theta(x^{k-1}|x^k)$ represents the distribution of real data. $x^0 = x$, $k = 1, \cdots, K$ is the diffusion time step, and $\beta_k \in (0, 1)$.

From Eq.(2), we can get

$$q(x^k|x^0) = N(\sqrt{\alpha_k}x^0, (1-\alpha_k)I) \tag{3}$$

where $\alpha_k = \prod_{i=1}^{k}(1-\beta_i)$, and $q(x^K|x^0) \approx N(0, I)$.

The reverse denoising diffusion process that needs to be learned is to gradually denoise from pure noise until the variables are obtained. This process is represented by a learnable neural network. The reverse denoising diffusion process is defied as:

$$p_\theta(x^{k-1}|x^k) = N(\mu_\theta x^k, \sum x_k) \tag{4}$$

And, μ_θ in Eq.(4) is expressed as:

$$\mu_\theta = \frac{1}{\sqrt{1-\beta_k}}(x^k - \frac{\beta_k}{\sqrt{1-\alpha_k}}\varepsilon_\theta x^k)\sum x_k) \tag{5}$$

where $\varepsilon_\theta \sim N(0, I)$.

Since the environment dynamics and action information of the diffuser are independent of the reward function, the objective functions of multiple tasks can be combined to achieve the combination of multiple tasks, which corresponds to the sequential execution of the subtasks mentioned above. In addition to task composability, the diffuser framework also has the ability of time composability and variable time planning. Since the diffusion model is fully convolutional in the prediction time dimension, the field of view of its trajectory planning is not determined by the framework, but by the dimension of the noise $\varepsilon_i^T \sim N(0, I)$ initialized in the backward process, which makes it possible to plan with variable time lengths. For the above-mentioned sequence subtasks, the diverse planning time lengths do not affect the quality of the diffusion model trajectory planning. For long-period complex interactive tasks, the diffuser can also generate globally consistent trajectories by iteratively improving local consistency, that is, it can generate a complete task trajectory by combining subsequence trajectories.

D. Data Collection and Sampling Trajectories

The collection of dataset D was obtained by randomly sampling a pre-defined action space A and pre-defined state space S. The termination condition for the data collection process is when the state leaves the state space S or the current position reaches the target point g.

Consider the system is governed by a unknown discrete-time system model:

$$s_{t+1} = f(s_t, a_t) \tag{6}$$

where $s_t \in \mathbb{R}^n$ is the state of system (6), $a_t \in \mathbb{R}^m$ is the action of system (6).

Define the trajectory of the system (6) is

$$\tau(z) = ((s_0, a_0), \cdots, (s_n, a_n)) \tag{7}$$

Trajectories (7) follow the following distribution

$$q(\tau) = \prod_{t=1}^{m} \delta(s_t - \hat{s}_t) \tag{8}$$

where $q(\tau)$ is the distribution of action. $q = (q_1, \cdots, a_n)$ is the trajectory, $q(A)$ is the distribution of trajectory, and $\hat{s}_t = f(s_{t-1}, a_{t-1})$.

Based on the collected dataset D, we obtain the trajectory distribution by distributing the action space according to Eq. (8), and then sample the trajectory using Eq. (9).

$$p_\theta(\tau^{k-1}|\tau^k) = N(\mu_\theta \tau^k, \sum \tau_k) \tag{9}$$

where $k = K, \cdots, 1$ and $\tau^K \sim N(0, I)$.

Algorithm 1 The implementation process of the entire task execution.

Input: Environmental information and the Human natural language instructions, goal g, robot state space S, robot action space A, trajectories $\tau(z)$

Output: Robot action a_0

According to the given prompt template, use ChatGPT 4 to split the total task L and obtain several sub tasks $L_1, \cdots, L_n$ that robots can execute
Initialization $t = 0$
Collect data set D in Isaac Sim
Training a Trajectory diffusion model ε_τ by utilizing the collected data set D according to Eq (2) to Eq (5)
repeat
  for each $k = \overline{K}, \cdots, 1$ do
    Sampling the trajectories $\tau(z)$ according to Eq. (9)
    Implementing the entire diffusion process
  end for
  Get the action trajectory $\tau(a)$
  Apply the first action a_0 in $\tau(a)$
until Reaches the goal g
return a_0

E. Overall algorithm implementation

Combining the descriptions in Sections II-A, II-B, II-C, and II-D, the overall embodied intelligence implementation solution is as follows:

In Algorithm 1, based on the environmental information collected by the RGB-D depth camera, LLM is first used to decompose human natural language instructions into several sub-tasks that can be executed by robots according to the designed prompt template. Then, a dataset combined states with actions is obtained in NVIDIA Isaac Sim. Based on this dataset, a diffusion model is trained to approximate the trajectory distribution. Then, based on the diffusion model, a reverse denoising process is carried out, ultimately obtaining the robot trajectory action.

III. Numerical Experiments

In this section, we validated the correctness of the proposed algorithm on a 7-degree-of-freedom Franka Emika Panda Robot in NVIDIA Isaac Sim.

A. Data Collection of Robot

We collected 11 pieces of data in NVIDIA Isaac Sim, each containing 3668 points. The 70 dimensional states in the state space S include: the 7-degree-of-freedom position of the Franka Emika Panda Robot, the position of the left and right ends of the end effector; The speed of the 7-degree-of-freedom Franka Emika Panda Robot and the speed of the left and right ends of the end effector; The positions, angular velocity, and linear velocity of 4 blocks and 1 box in the scene; The target positions of the four blocks. The 9-dimensional actions in the action space A include: the angular velocity of the 7-degree-of-freedom Franka Emika Panda Robot and the linear velocity of the left and right ends of the end effector.

TABLE I

The hyperparameters in training process.

Hyperparameter	Value
Optimizer	Adam
Batch size	32
Learning rate	2×10^{-5}
Traning steps	10^6
Epochs	100
Diffusion steps k	20

B. Traning Process and Training Results

The hyperparameters of the training process are shown in table 1.

C. Scene and Task Design

The experimental scene is shown in Figure 2 below:

As shown in Figure 2 above, a 7-degree-of-freedom Franka Emika Panda is used in NVIDIA Isaac Sim to complete the grasping task. The scenario contains four blocks of different colors and a box. We utilize two Realsense RGB-D cameras to observe environmental information. LLM calls the GPT-4 API, and we trained a VLM based on OWL-Vit to recognize objects in the scene.

We have planned 6 categories of tasks, which are designed in Table 2.

D. Experiment Results

IV. CONCLUSIONS

This paper has presented a universal Embodied Artificial Intelligence scheme for robot manipulation tasks, which combines the generation ability of large language model and the stable advantages of diffusion policy to generate robot actions. This scheme improves the flexibility of robot action generation, enhances the success rate and generalization of task execution. The simulation results have demonstrated the effectiveness and correctness of this method. In the future, we will verify the reliability of the proposed algorithm through physical experiments and apply this universal Embodied AI solution to humanoid robot manipulation tasks.

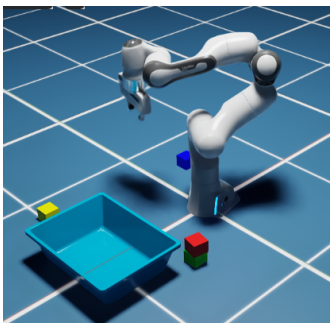

Fig. 2. A simulation experimental scenario for a Franka Emika Panda in NVIDIA Isaac Sim.

ACKNOWLEDGMENT

We sincerely thank Beijing Embodied Artificial Intelligent Robot Innovation Center Co., Ltd. for their strong support of our research work.

References

[1] R. Chrisley, "Embodied artificial intelligence," Artificial intelligence, vol. 149, no. 1, pp. 131–150, 2003.

[2] R. Pfeifer and F. Iida, "Embodied artificial intelligence: Trends and challenges," Lecture notes in computer science, pp. 1–26, 2004.

[3] 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.

[4] C. Zhao, S. Yuan, C. Jiang, J. Cai, H. Yu, M. Y. Wang, and Q. Chen, "Erra: An embodied representation and reasoning architecture for long-horizon language-conditioned manipulation tasks," IEEE Robotics and Automation Letters, vol. 8, no. 6, pp. 3230–3237, 2023.

[5] S. Vemprala, R. Bonatti, A. Bucker, and A. Kapoor, "Chatgpt for robotics: Design principles and model abilities. 2023," Published by Microsoft, 2023.

[6] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2motion: From natural language instructions to feasible plans," Autonomous Robots, vol. 47, no. 8, pp. 1345–1365, 2023.

[7] K. Bousmalis, G. Vezzani, D. Rao, C. Devin, A. X. Lee, M. Bauza, T. Davchev, Y. Zhou, A. Gupta, A. Raju et al., "Robocat: A self-improving foundation agent for robotic manipulation," arXiv preprint arXiv:2306.11706, 2023.

[8] 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.

[9] S. Belkhale, T. Ding, T. Xiao, P. Sermanet, Q. Vuong, J. Tompson, Y. Chebotar, D. Dwibedi, and D. Sadigh, "Rt-h: Action hierarchies using language," arXiv preprint arXiv:2403.01823, 2024.

[10] 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.

[11] H. Zhen, X. Qiu, P. Chen, J. Yang, X. Yan, Y. Du, Y. Hong, and C. Gan, "3d-vla: A 3d vision-language-action generative world model," arXiv preprint arXiv:2403.09631, 2024.

[12] T.-W. Ke, N. Gkanatsios, and K. Fragkiadaki, "3d diffuser actor: Policy diffusion with 3d scene representations," arXiv preprint arXiv:2402.10885, 2024.

[13] B. Kang, X. Ma, C. Du, T. Pang, and S. Yan, "Efficient diffusion policies for offline reinforcement learning," Advances in Neural Information Processing Systems, vol. 36, 2024.

[14] M. Janner, Y. Du, J. B. Tenenbaum, and S. Levine, "Planning with diffusion for flexible behavior synthesis," arXiv preprint arXiv:2205.09991, 2022.

[15] X. Ma, S. Patidar, I. Haughton, and S. James, "Hierarchical diffusion policy for kinematics-aware multi-task robotic manipulation," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 18 081–18 090.

[16] Z. Wang, J. J. Hunt, and M. Zhou, "Diffusion policies as an expressive policy class for offline reinforcement learning," arXiv preprint arXiv:2208.06193, 2022.

[17] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," arXiv preprint arXiv:2303.04137, 2023.

[18] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, W. Ai, B. Martinez et al., "Behavior-1k: A human-centered, embodied ai benchmark with 1,000 everyday activities and realistic simulation," arXiv preprint arXiv:2403.09227, 2024.

[19] X. Kong, W. Zhang, J. Hong, and T. Braunl, "Embodied ai

in mobile robots: Coverage path planning with large language models," arXiv preprint arXiv:2407.02220, 2024.
[20] 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.
[21] 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.
[22] M. Minderer, A. Gritsenko, A. Stone, M. Neumann, D. Weissenborn, A. Dosovitskiy, A. Mahendran, A. Arnab, M. Dehghani, Z. Shen et al., "Simple open-vocabulary object detection," in European Conference on Computer Vision. Springer, 2022, pp. 728–755.
[23] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu, "Unet 3+: A full-scale connected unet for medical image segmentation," in ICASSP 2020-2020 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2020, pp. 1055–1059.
[24] X. Li, H. Chen, X. Qi, Q. Dou, C.-W. Fu, and P.-A. Heng, "H-denseunet: hybrid densely connected unet for liver and tumor segmentation from ct volumes," IEEE transactions on medical imaging, vol. 37, no. 12, pp. 2663–2674, 2018.
[25] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," Advances in neural information processing systems, vol. 33, pp. 6840–6851, 2020.