This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Phoenix: A Motion-based Self-Reflection Framework for Fine-grained Robotic Action Correction

Wenke Xia[1,2,*], **Ruoxuan Feng**[1], **Dong Wang**[2], **Di Hu**[1,†]

[1]Gaoling School of Artificial Intelligence, Renmin University of China, Beijing
[2] Shanghai AI Laboratory

Abstract

*Building a generalizable self-correction system is crucial for robots to recover from failures. Despite advancements in Multimodal Large Language Models (MLLMs) that empower robots with semantic reflection ability for failure, translating semantic reflection into "**how to correct**" fine-grained robotic actions remains a significant challenge. To address this gap, we build the Phoenix framework, which leverages **motion instruction** as a bridge to connect high-level semantic reflection with low-level robotic action correction. In this motion-based self-reflection framework, we start with a dual-process motion adjustment mechanism with MLLMs to translate the semantic reflection into coarse-grained motion instruction adjustment. To leverage this motion instruction for guiding "**how to correct**" fine-grained robotic actions, a multi-task motion-conditioned diffusion policy is proposed to integrate visual observations for high-frequency robotic action correction. By combining these two models, we could shift the demand for generalization capability from the low-level manipulation policy to the MLLMs-driven motion adjustment model and facilitate precise, fine-grained robotic action correction. Utilizing this framework, we further develop a lifelong learning method to automatically improve the model's capability from interactions with dynamic environments. The experiments conducted in both the RoboMimic simulation and real-world scenarios prove the superior generalization and robustness of our framework across a variety of manipulation tasks. Our code is released at https://github.com/GeWu-Lab/Motion-based-Self-Reflection-Framework.*

1. Introduction

> "Failure is simply the opportunity to begin again, this time more intelligently." — *Henry Ford*

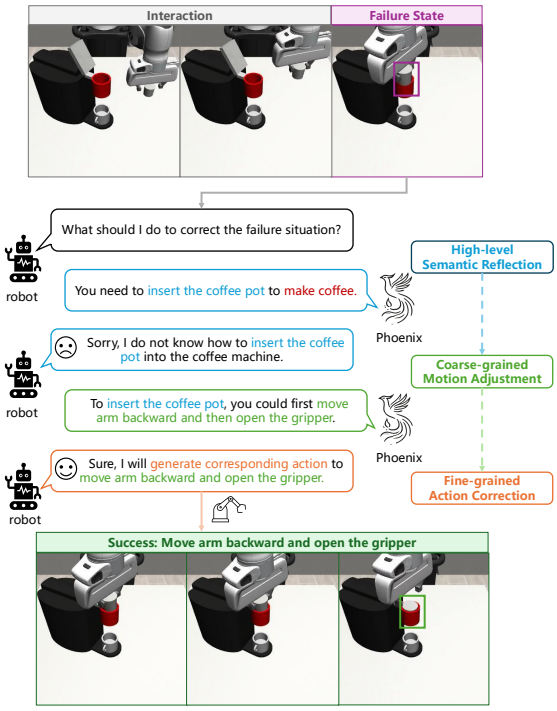


Figure 1. Our motion-based self-reflection framework utilizes coarse-grained motion instruction as a bridge to convert the high-level semantic reflection into fine-grained robotic action correction, thereby facilitating generalizable and precise action correction with perceptual and inferential capabilities of MLLMs.

Humans are naturally equipped with the ability to correct their behaviors by intentionally reflecting on actions that lead to failure. [12, 25]. By analyzing failure situations from high-level semantic reflection and low-level action correction perspective, humans can efficiently adapt to dynamic environments [8]. To emulate the correction capability and foster a continuous cycle of self-improvement in robots, researchers [26, 30, 35] have sought to develop self-reflection systems that enable robots to recover from and learn through their failure interactions.

Among them, some existing self-correction systems [19,

*Work is done during internship at Shanghai AI Laboratory
†Corresponding author

26, 34] leverage reinforcement learning to guide robots in correct low-level action execution through reward functions. However, the reliance on reinforcement learning limits the ability of these self-correction systems to generalize across long-horizon manipulation tasks, due to unstable training processes [10] and the need for task-specific prior knowledge [9, 35]. To construct a generalizable and stable self-correction system, recent works [17, 30, 31] borrow the inferential capability of Multi-modal Large Language Models (MLLMs) to propose closed-loop high-level semantic reflection framework for failure correction. Although these semantic self-reflection frameworks can decompose the failure correction process into semantic subgoals, they primarily rely on a predefined skill library to execute the detailed subgoals, which fails to utilize the generalization ability of MLLMs in fine-grained robotic action correction.

To maximize the generalization potential of MLLMs for action correction, we propose motion instruction as a bridge to convert high-level semantic reflection to fine-grained robotic action correction. Motion instruction refers to coarse-grained robotic movement commands such as "move arm backward" and "adjust gripper position". Serving as an intermediate layer, motion instruction could provide general, low-frequency decision information for high-frequency robotic action execution, which makes it an excellent medium for embedding the knowledge of MLLMs into fine-grained action correction. As shown in Figure 1, we decompose the semantic reflection knowledge into coarse-grained motion instruction adjustment to indicate "how to correct" fine-grained action for low-level policy execution. This transition shifts the perceptual and decision-making requirements from low-level robotic policy to the MLLMs-driven motion adjustment model, thereby enabling generalizable, fine-grained robotic action correction.

Hence, in this work, we build the *Phoenix* framework, a motion-based self-reflection framework designed to convert the semantic reflection of MLLMs into fine-grained robotic action correction. Initially, we develop a dual-process motion adjustment mechanism to ensure efficient prediction through a motion prediction module, while addressing failure with a motion correction module. Concretely, we first utilize expert demonstration trajectories to train the motion prediction module for efficient motion instruction generation. Despite its efficiency in generating initial instructions, this module often struggles to handle failure scenarios. To recover from failures, we collect a comprehensive failure correction dataset and fine-tune the motion correction module, which thoroughly provides adjusted motion instructions through a chain-of-thought approach. By integrating these two modules, the dual-process motion adjustment mechanism guarantees both robustness and efficiency, facilitating the generation of accurate motion instructions. As the coarse-grained motion instructions only provide general and low-frequency guidance for robotic manipulation, we further design a multi-task motion-conditioned diffusion policy that integrates visual observations to translate motion instruction into precise, high-frequency action corrections for manipulation tasks. Moreover, by leveraging these correction trajectories, we propose a lifelong learning method that iteratively enhances the model's capabilities through interaction, ensuring continuous improvements in performance and adaptability to dynamic environments.

To validate the efficacy of our framework, we conduct experiments across 9 contact-rich robotic manipulation tasks within the RoboMimic simulation [20]. The results demonstrate that our method could provide more precise action correction from failures through self-reflection and facilitate self-improvement through interactions with environments. Further, we conduct two novel manipulation tasks with color disruption and position distribution disruption, proving the generalization ability of our framework. The real-world experiments also demonstrate the applicability and robustness of our approach in practical scenarios.

2. Related Work

2.1. Robotic Self-correction Systems

Self-correction serves as a crucial mechanism enabling robots to recover from failures. To achieve self-correction on the low-level robotic action, Reinforcement Learning [1, 23] is proposed to guide robots in adjusting behaviors via reward signals. However, reinforcement learning strategies encounter difficulties in intricate robotic environments primarily due to learning inefficiency. Borrowing the common knowledge of MLLMs, Raman et al. [24] proposes to build a semantic self-reflection system for long-horizon task planning. To facilitate interaction with environments, some efforts generate robotic action through simulation APIs [17, 29, 30] and predefined action skill libraries [14, 16]. However, the reliance on predefined low-level skill libraries makes semantic self-reflection frameworks fail to directly provide fine-grained action correction. To provide low-level action feedback for robotic manipulation, recent works [6, 16, 32] suggest adjusting end-effector poses to refine actions. Nonetheless, these techniques are primarily restricted to simple manipulation tasks that employ motion planning and fail to generalize to contact-rich manipulation scenarios. In this work, we utilize motion instruction as an intermediate layer to guide robotic action correction, borrowing the perceptual and inferential capability of MLLMs for fine-grained robotic action correction.

2.2. Robotic Manipulation Policy

The development of generalizable strategies for robotic manipulation remains a persistent challenge in robotics research. ACT [38] is proposed to predict action sequences

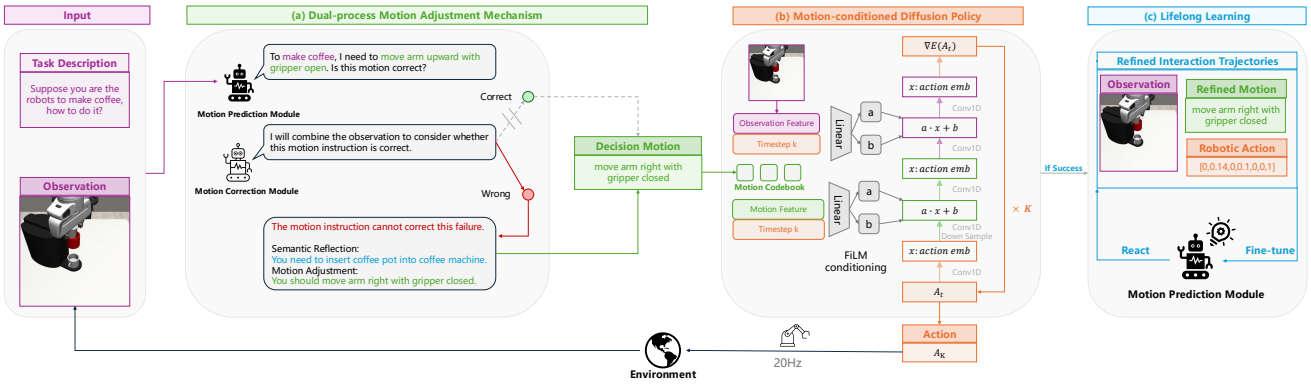


Figure 2. The pipeline of our motion-based self-reflection framework. (a) demonstrates the dual-process motion refinement mechanism which leverages the motion prediction module for efficient motion instruction prediction and motion correction module for comprehensive failure correction. (b) illustrates the motion-conditioned diffusion policy which converts the low-frequency motion instruction guidance into high-frequency robotic action. The lifelong learning approach in (c) iteratively enhance the ability of the motion prediction module from the refined interaction trajectories.

to ensure temporal alignment of actions, while the Diffusion Policy [5] addresses multi-modal action distributions to enhance robust manipulation. Driven by the development of foundation models [13, 28], recent works [4, 7, 14, 22, 36, 37] leverage the world knowledge of MLLMs to facilitate task decomposition and planning for robotic manipulation. Concurrently, other works [3, 11, 21, 27] collect large-scale robotic manipulation demonstrations to train generalizable language-conditioned manipulation policy across different robots and tasks. Moreover, RT-H [2] employs detailed motion commands instead of semantic language inputs, fostering flexible and generalizable manipulation capabilities through the advanced perceptual abilities of MLLMs. However, the prohibitive costs associated with robotic data collection limit the scalability of existing imitation learning approaches. To address this, we propose the motion-based self-reflection framework to enable autonomous self-improvement through continuous interaction with environments without human intervention.

3. Motion-based Self-Reflection Framework

3.1. Challenges in Robotic Self-Corrction Model

Building a generalizable and robust self-correction system is a key component in achieving failure correction for robots. Multi-modal Large Language Models (MLLMs) have already been applied to the construction of robotic self-reflection framework to recover from failures. However, existing systems mainly focus on semantic reflection, and their application to fine-grained action correction still faces the following two issues:

- How to enable MLLMs to understand manipulation tasks and provide detailed correction information?
- How to convert the correction information provided by MLLMs into precise, high-frequency robotic actions?

To address these issues, we propose the *Phoenix* framework, a motion-based self-reflection framework that integrates a dual-process motion adjustment mechanism and a multi-task motion-conditioned diffusion policy as illustrated in Figure 2. As detailed in Section 3.2, the dual-process motion adjustment mechanism is developed to ensure efficient and accurate motion instruction generation. Further, the motion-conditioned diffusion policy is proposed to translate the coarse-grained motion instructions into precise robotic actions, as explained in Section 3.3. Based on the refined manipulation trajectories, we propose a lifelong learning approach to facilitate robotic self-improvement, as outlined in Section 3.4.

3.2. Dual-process Motion Adjustment Mechanism

The dual-process motion adjustment mechanism is designed to ensure efficient motion prediction through a motion prediction module, while comprehensively addressing failure with a motion correction module. Given the observation O and task description T, we first train a Motion Prediction Module (MPM) with expert demonstration dataset D_e to generate initial motion instruction m_i. However, the MPM trained on expert demonstrations struggles to handle failure situations. Thus, we construct a comprehensive failure correction dataset D_c to fine-tune the Motion Correction Module (MCM), enabling it to analyze the failure situation and adjust m_i with a chain-of-thought approach. If m_i is deemed correct, we adopt it as the decision motion instruction m_d for further robotic action prediction. Otherwise, we employ the MCM to analyze the failure situation and generate adjusted motion instruction m_a as decision motion instruction m_d. Through the guidance of m_d, our motion-based diffusion policy can generate high-frequency corrections to the robotic actions. As described in Algorithm 1, we establish the dual-process motion adjustment mechanism to guarantee the efficiency and accuracy of motion instruction generation for fine-grained robotic action prediction.

Algorithm 1 Self-Reflection w/. Dual-Process Motion Adj.

Require: Task description T, Observation O, Environment E, Motion prediction module MPM, Motion correction module MCM, Motion-conditioned diffusion policy π, Exploration timestep K,

$O_1 \leftarrow E.reset()$
for $k = 1$ **to** K **do**
  $m_i \leftarrow MPM(O_k, T)$
  $failure_flag, semanic_info \leftarrow MCM(O_k, m_i)$
  if $failure_flag$ is **true then**
    $m_a \leftarrow MCM(O_k, sematic_info)$
    $m_d \leftarrow m_a$
  else
    $m_d \leftarrow m_i$
  end if
  $a \leftarrow \pi(O, m_d)$
  $O_{k+1} \leftarrow E.step(a)$
end for

Motion Prediction Module (MPM). To fully harness the perceptual and decision-making capabilities of MLLMs for efficient motion instruction prediction, we develop a motion instruction dataset from the expert demonstrations dataset D_e to fine-tune MLLMs for robotic manipulation tasks. To construct the expert dataset, we filter the robotic action to get dominant motion from expert demonstration with a threshold, generating a set of motion instructions that include arm direction and gripper control. In practice, we find that separating arm direction instruction and gripper control instruction would cause the misalignment between textual motion instruction and fine-grained robotic action. To address this issue, we combine the direction movement with gripper control, resulting in unified instruction formats for motion instruction such as "move arm right with gripper closed". In addition, we add the "make slight adjustments to gripper position" instruction to model the minor robotic actions below the threshold. Through the automatic construction method, we build 37 types of motion instructions as guidance for further robotic action prediction. By training on the expert dataset, the MPM acquires an understanding of robotic manipulation tasks and could efficiently generate an initial motion instruction m_i.

Motion Correction Module (MCM). During interactions with the environment, robots may execute incorrect actions, leading to a failure situation in the task. However, the MPM trained on success expert data often struggles to recover from these failure scenarios. Thus, we develop the motion correction module, to identify failure scenarios and correct behaviors from such situations. As shown in Figure 2(a), the MCM would evaluate the initial motion instruction m_i and conduct a dual process based on the evaluation results to achieve efficient and accurate motion instruction adjustment. Once encountering the failure situation, the MCM would first analyze the type of failure and derive a semantic-level correction goal, such as "insert the coffee pot into the coffee machine". Based on this correction goal, the MCM further adjusts the motion instructions with its learned failure-correction knowledge, ultimately generating an accurate motion instruction through a hierarchical chain-of-thought approach.

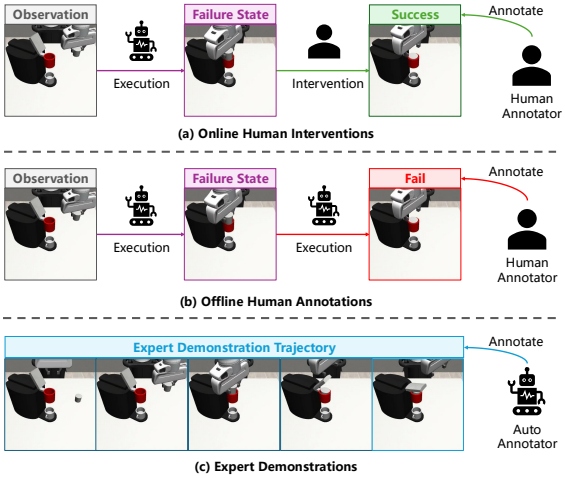


Figure 3. Illustrations of our correction data: online human interventions, offline human annotations, and expert demonstrations.

To equip MCM with the capabilities for failure detection and correction, we construct a comprehensive correction dataset as illustrated in Figure 3. This dataset includes three types of feedback data, encompassing both semantic and motion perspectives, and is categorized as follows:

- **Online Human Intervention.** We implement the human-in-the-loop method for trajectory collection. We first deploy the motion prediction model to interact with environments, then manually intervene to correct the motion instructions whenever the agent encounters failure situations. This method could collect accurate and high-quality motion correction data to ensure task completion. However, it requires frequent manual interactions with the environment, which leads to significant time consumption and makes it difficult to collect large-scale data.
- **Offline Human Annotation.** We utilize the motion prediction model to gather trajectory data, periodically sampling trajectories and annotating them with semantic reflections and motion correction details. While the accuracy of offline annotated data cannot be guaranteed due to its inability to interact with environments for verification, this method offers a significant volume of annotations.
- **Expert Demonstration.** We automate annotations on expert trajectories. Since these trajectories are successful, this data is used to provide accurate motion information to enhance the model's motion prediction capabilities.

By fine-tuning MCM on this dataset, we enhance the

MCM to thoroughly comprehend various types of failure situations and provide motion instruction corrections. Through the integration of MPM and MCM, the dual-process motion adjustment mechanism enables efficient motion instruction generation while ensuring comprehensive correction in failure situations.

3.3. Motion-conditioned Diffusion Policy

As the motion instruction only provides general and low-frequency guidance for manipulation, we train a multi-task motion-conditioned diffusion policy π to convert the motion instructions into precise, high-frequency robotic actions. This policy takes observations O and decision motion instructions m_d to output robotic actions a. To ensure the policy adheres to the motion instruction, we make adjustments as depicted in Figure 2(b):

First, we observe that existing pre-trained language models often struggle to capture the discriminative features of various motion instructions. This limitation hampers their ability to follow various motion instructions. To address this issue, we introduce a learnable motion codebook designed to provide discriminative features for motion instructions. For a given decision motion instruction m_d, the codebook would retrieve the corresponding motion feature to facilitate accurate robotic action prediction.

Further, we find that the direct concatenation of observation representation and motion instruction feature would cause the diffusion policy to prefer to rely on the vision information for action prediction, thereby hindering the effectiveness of the motion instruction guidance. To address this issue, we take the observation representation and motion instruction feature as separate conditions in different stages of the diffusion policy, allowing the model to better learn the guidance information from the motion instruction and thereby promote precise action correction.

By integrating these two adjustments, we train the diffusion policy for action prediction with the following loss:

$$\mathcal{L} = \text{MSE}(\mathcal{E}^k, \pi(\mathcal{O}, \mathcal{M}, \mathcal{A}^0 + \mathcal{E}^k, k)), \tag{1}$$

where $\mathcal{O}$ is the observation representation, $\mathcal{M}$ is the motion instruction feature, $\mathcal{A}^0$ is the ground truth robotic action, $\mathcal{E}^k$ indicates the random noise at the denoising iteration k. Through minimizing the loss function in Eq 1, the diffusion policy π could effectively predict precise, high-frequency robotic action guided by motion instruction.

3.4. Action Correction for Lifelong Learning

The dual-process motion adjustment mechanism leverages the MPM to efficiently predict motion instructions and the MCM to adjust them with a comprehensive chain-of-thought approach. However, the reliance on the chain-of-thought poses challenges in adapting to real-time scenarios due to its time-consuming. Furthermore, the collection of manipulation data and correction data is exceedingly labor-intensive. Thus, we propose a lifelong learning method that equips the MPM with both motion prediction and failure correction capabilities through learning from the refined interaction trajectories as illustrated in Figure 2(c), which enhances our model to adapt and react quickly to the environment without human intervention.

Benefiting from the motion-conditioned diffusion policy which could adhere to the motion instruction to generate task-aware robotic action, we can enhance the robot's capabilities through only improving the MPM informed by the refined interaction trajectory. To address the issue of catastrophic forgetting, we mix the refined interaction trajectory with expert demonstration for co-fine-tuning, allowing the model to simultaneously learn failure correction and enhance the motion prediction capabilities. Through updates from refined interaction trajectories, our model can achieve self-improvement by learning from the knowledge of the motion correction module, achieving fast and accurate manipulation for contact-rich manipulation tasks.

4. Experiments

To comprehensively evaluate our framework, we propose experiments to answer the following questions:

- Does our motion-guided self-reflection model enhance the precision of action correction? Section 4.2
- Can our model achieve lifelong learning from interaction with environments? Section 4.3
- Does our framework can generalize across novel tasks? Section 4.4
- Can our framework ensure reliability and robustness in real-world scenarios? Section 4.5

4.1. Experiment Settings

In this work, we conduct experiments on 9 contact-rich manipulation tasks in RoboMimic [20], ranging from long-horizon tasks like ThreePieceAssembly to fine-grained manipulation tasks like Threading. To transform high-level semantic information into motion instructions, we filter expert demonstrations to obtain over 160,000 pairs of motion instructions and observations. The dataset includes 37 types of motion instructions, which are utilized to fine-tune the LLaVA-v1.5 model [15] as the motion prediction module. Furthermore, to develop the motion correction module that integrates semantic comprehension and motion instruction adjustment, we collect correction data comprising 3,644 online human intervention data, 7,365 offline human annotations, and 6,378 expert demonstrations. We filter out the correction dataset to balance the proportion of various failure situations to enhance the model's correction capabilities. Ultimately, to translate motion instructions into precise robotic actions, we train a multi-task motion-conditioned diffusion policy using a learnable motion instruction code-

Methods	Coffee_D0	Coffee_D1	Stack_D0	Stack_D1	StackThree_D0	StackThree_D1	Threading_D0	ThreePieceAssembly_D0	ThreePieceAssembly_D1	Mean
OpenVLA [11]	42%	18%	84%	**86%**	36%	**20%**	20%	28%	**8%**	38.0%
Task-conditioned	66%	24%	88%	68%	30%	6%	74%	20%	0%	41.8%
Subgoal-conditioned	76%	26%	88%	74%	24%	6%	**78%**	20%	2%	43.8%
Motion-conditioned	68%	32%	92%	84%	38%	16%	58%	30%	4%	46.9%
Subgoal Self-reflection	80%	32%	88%	78%	32%	6%	80%	34%	2%	48.0%
Phoenix (Ours)	**94%**	**48%**	**96%**	**86%**	**50%**	**20%**	68%	**52%**	6%	**57.8%**
Human Intervention (Oracle)	100%	100%	100%	90%	70%	40%	100%	70%	40%	78.9%

Table 1. Comparison experiments results across 9 manipulation tasks in RoboMimic Simulation. The results demonstrate that our motion-based self-reflection method achieves better performance by facilitating precise correction of fine-grained robotic actions.

book, incorporating 500 demonstrations per task. During inference in simulation, our dual-process motion adjustment mechanism would provide motion instruction at $5Hz$, and the diffusion policy would extend the motion instruction with visual observations to a $20Hz$ action sequence to control the robot. For each task, we conducted 50 trials and report the average success rate. More implementation details could refer to Supp.A.

4.2. Performance of Motion Self-Reflection Model

4.2.1 Comparison Results

To evaluate our motion-based self-reflection framework, we compare our framework with other approaches. To ensure fairness, all our comparison methods are trained on the expert data from the simulation environment, with the decision model using LLaVA-v1.5 and the underlying policy employing a diffusion policy.

- **OpenVLA [11].** We fine-tune the OpenVLA model to provide baseline performance for multi-task experiments.
- **Task-conditioned policy.** We take the task description as the condition for diffusion policy without the reflection framework, as a variance of RT-1 [3] and Octo [27].
- **Subgoal-conditioned policy.** We fine-tune a LLaVA-v1.5 to predict subgoals at 5Hz, which are utilized as the condition for diffusion policy without reflection framework. This method borrows the semantic comprehension capabilities of MLLMs, and is implemented as a variance of PaLM-E [7] with an individual diffusion policy.
- **Motion-conditioned policy.** We fine-tune a LLaVA-v1.5 as the motion prediction model to provide motion instructions at 5Hz, using these predictions to condition the diffusion policy without the reflection framework. This method employs the perceptual and inferential capacities of MLLMs, realized as a variation of RT-H [2] with an individual diffusion policy.
- **Human Intervention.** We manually correct the wrong motion instructions for the motion-conditioned policy. This method provides an upper bound on the performance of self-reflection methods. Due to labor costs, the results are presented as average success rates across 10 trials.
- **Subgoal Self-reflection.** We fine-tune a LLaVA-v1.5 as subgoal self-reflection model and apply it to the subgoal-condition policy. This method is designed to validate the effectiveness of the semantic self-reflection model.

As shown in Table 1, we first compare three different condition methods. Borrowing the perception and inferential ability of MLLMs, the subgoal-conditioned and motion-conditioned policies are better than the task-conditioned policies. The results prove the potential applications of MLLMs in various complex robotic manipulation tasks.

Focusing on specific tasks, we observe that the motion-conditioned policy excels in long-horizon tasks such as StackThree_D0 and ThreePieceAssembly_D0. However, this policy depends on consistent and accurate motion instruction predictions, which poses challenges in fine-grained manipulation tasks like Threading_D0.

By providing correction subgoals, the subgoal self-reflection method consistently outperforms the subgoal-conditioned policy, particularly in long-horizon manipulation tasks such as "StackThree_D0", which demonstrates the efficacy of the self-reflection framework.

The OpenVLA model demonstrates strong performance in certain long-horizon tasks, leveraging its end-to-end action token prediction capability. However, the lack of observation history and action chunking poses significant challenges in handling complex, fine-grained manipulation tasks like Threading_D0.

Notably, our Phoenix method achieves more substantial improvements than the subgoal self-reflection method, demonstrating the effectiveness of motion-conditioned method in long-horizon sequential tasks and fine-grained manipulation tasks. Benefiting from our motion-based correction method, agent could correct fine-grained action through motion instruction adjustment while the subgoal-conditioned self-reflection model fails to recover from most failure situations. Besides, the human intervention method achieves high success rates across multiple tasks, demonstrating that our motion-conditioned diffusion policy can effectively adhere to motion instructions for manipulation tasks. This result indicates that our method can perform well under the correct motion instructions, showcasing the significant potential of motion-conditioned self-reflection.

4.2.2 Ablation Results

In this work, we propose a motion prediction module to provide initial motion instruction, and a motion correction

Methods	Coffee_D0	Coffee_D1	Stack_D0	Stack_D1	StackThree_D0	StackThree_D1	Threading_D0	ThreePieceAssembly_D0	ThreePieceAssembly_D1	Mean
Motion-conditioned	68%	32%	92%	84%	38%	16%	58%	30%	4%	46.9%
Expert-Correction Mixture	74%	36%	94%	86%	38%	22%	64%	30%	2%	49.6%
Expert-Correction Mixture with Self-Reflection	76%	30%	92%	**90%**	46%	**26%**	64%	34%	4%	51.3%
Phoenix (Ours)	**94%**	**48%**	**96%**	86%	**50%**	20%	**68%**	**52%**	**6%**	**57.8%**

Table 2. The ablation results of our dual-process motion adjustment mechanism. The results prove that our model, which separates motion prediction module and motion correction module, can provide more precise motion adjustment for action execution in manipulation tasks.

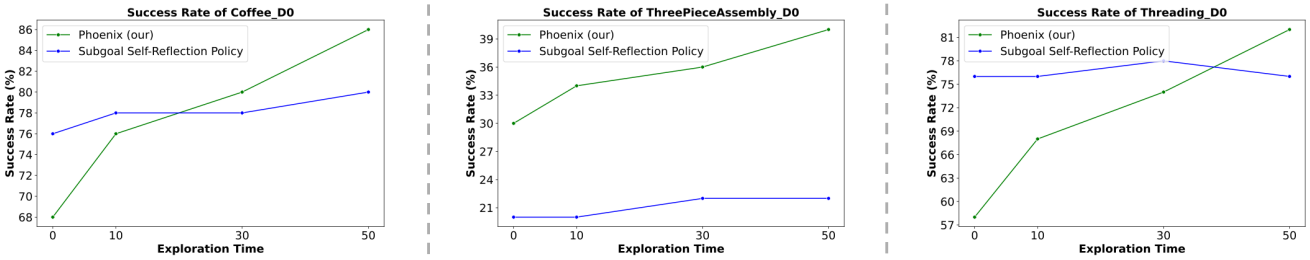


Figure 4. The lifelong learning results. The results prove that our motion-based self-reflection method could iteratively improve performance through interactions with environments.

module to provide fine-grained motion correction. Drawing upon prior research [18, 33], data mixture proportions could influence the efficacy of LLMs. In this section, we investigate whether integrating expert demonstrations with correction dataset, could also enhance the perception and decision-making capabilities of MLLMs for robotic manipulation with the following ablation methods:

- **Expert-Correction Mixture.** We mix the expert demonstration and correction data for co-training the motion prediction model.
- **Expert-Correction Mixture with Self-Reflection.** We mix the expert demonstration and correction data for co-training a unified model to provide initial motion instruction and adjust the instruction.

As illustrated in Table 2, the results show that co-training with mixture data yields superior performance compared to models trained exclusively on expert demonstration data. This indicates that combining various types of feedback data can enhance the decision-making and perception capabilities of MLLMs. It also validates the viability of our approach to achieving self-improvement through interaction.

Besides, the mixture training model with self-reflection performs better than the one without self-reflection, which suggests that our designed motion-based self-reflection method can enhance the decision-making capabilities of robots and facilitate the correction of fine-grained actions.

However, we find that utilizing the mixture of data to train a unified model to serve as both the motion prediction module and motion correction module fails to provide accurate correction information compared to our separated motion correction module. This suggests that the mixture training strategy may not fully leverage the strengths of each dataset to achieve better correction effects under the significant data scale discrepancies (160,000 expert demonstrations vs. 16,000 feedback data). The results indicate that our dual-process motion adjustment mechanism can effectively leverage the expert demonstration and correction dataset, leading to accurate motion instruction adjustment.

We also provide ablation results of our designed motion codebook in Supp.B.

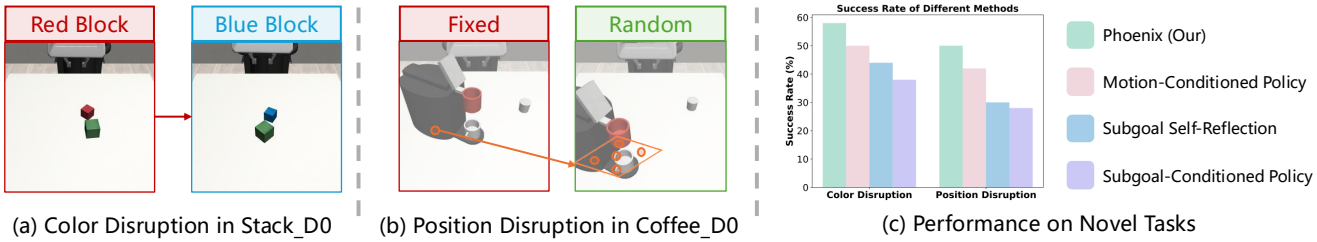
4.3. The Performance of Lifelong Learning

In this section, we explore whether our Phoenix framework can facilitate lifelong learning through interactions. Concretely, we deploy the motion self-reflection model to interact within the environments and utilize the successful trajectories to iteratively fine-tune our motion prediction model after 10, 30, and 50 rollouts. To avoid catastrophic forgetting, we combine 20 expert demonstrations to co-fine-tune the motion prediction module.

We compare the lifelong learning ability of our motion-based self-reflection model and subgoal-based self-reflection model. During testing, we record the average success rate over 50 trials. As shown in Figure 4, the subgoal-based lifelong learning fails to enhance model performance during the exploration phase due to its inability to provide fine-grained action correction. In contrast, our method corrects underlying action execution during interactions, allowing the robot to better learn from the refined trajectories, thereby achieving self-improvement.

4.4. Generalization to Novel Tasks

In this section, we evaluate the generalization ability of our Phoenix framework in color disruption and position disruption novel tasks as shown in Figure 5. In the color disruption setting, we replace the red block with the blue block in the Stack_D0 task to verify whether our model could generalize to object manipulation tasks with different visual characteristics. In the position disruption setting, we change the fixed position of the coffee machine to a randomly placed posi-



(a) Color Disruption in Stack_D0 (b) Position Disruption in Coffee_D0 (c) Performance on Novel Tasks

Figure 5. In the color disruption setting, we replace the red block with the blue block in the Stack_D0 task as shown in (a). In the position disruption setting, we change the position of the coffee machine from a fixed point ○ to a random position from the rectangle □ in the Coffee_D0 task as illustrated in (b). The results in (c) prove that our framework could generalize well to these novel task settings.

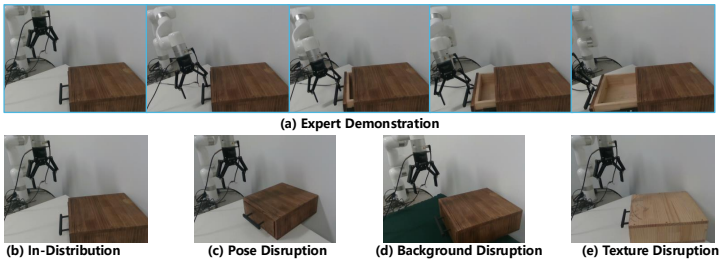


Figure 6. The real-world experiments with different variations.

Model	In-Dis.	Pose Dis.	Bg.	Tex.
OpenVLA	55%	30%	35%	45%
Task	60%	25%	25%	45%
Motion	60%	35%	30%	50%
Ours	75%	55%	45%	65%

Table 3. The real-world experiment results.

Task	Motion	10 rollout	30 rollout
In-Dis.	60%	65%	75%
Pose Dis.	35%	45%	50%

Table 4. The lifelong learning results

tion within a specific area in the Coffee_D0 task to verify whether our method could generalize to unseen scenarios.

For these novel tasks, although the subgoal-conditioned policy could predict correct high-level semantic subgoal for manipulation, this method fails to predict precise robotic action to complete the tasks. Due to its limitation of providing high-level semantic correction information, the subgoal self-reflection method fails to effectively leverage the knowledge of MLLMs for action correction to manipulation tasks. In contrast, as shown in Figure 5(c), our motion-conditioned policy could generate fine-grained motion instruction to achieve generalizable manipulation benefiting from the perception and inferential capability of MLLMs. Besides, our method could achieve better performance on novel tasks by comprehensively refining the motion instruction with the motion-based self-reflection framework.

4.5. Real-World Experiments

In real-world scenarios, we conduct the challenging "drawer open" articulated object manipulation task as shown in Fig 6(a), where the robot needs to align gripper with handle through precise rotations to open the drawer. We utilize the spacemouse device to collect 100 expert demonstrations with 14 motion instructions (e.g.,"move arm right","rotate around x-axis"). We train a motion-conditioned diffusion policy to convert instructions into robotic actions. During the inference, we introduce human-in-the-loop interventions to manually correct failure situations to collect 20 corresponding refined interaction trajectories to train our motion correction module. All models are only fine-tuned on the real-world data.

To validate generalization, we design 4 settings as shown in Fig 6(b-e). In the pose disruption setting, we change the pose distribution of the drawer. For the background disruption setting, the background color was modified to green. In the texture disruption setting, the texture of the drawer was altered to evaluate performance under significant visual variations. The results in Tab 3 demonstrate the generalization ability of our method. We also evaluate lifelong learning, the results in Tab 4 show that our model achieves self-improvement in real world.

We also provide more real-world task experiments with a rule-based manipulation policy to prove the effectiveness of our motion-based self-reflection method in Supp.C.

5. Conclusion

In this work, we propose a motion-based self-reflection framework to convert the semantic reflection of MLLMs into fine-grained robotic action correction. Based on this framework, we further automatically improve the model's capability from interactions. We hope this motion-based self-reflection framework could bring insights for enhancing the generalization capabilities of agents in robotic manipulation tasks through the integration of MLLMs.

6. Acknowledgement

The project was supported by National Natural Science Foundation of China (NO.62106272), Shanghai AI Laboratory, National Key R&D Program of China (2022ZD0160101) and CCF-Zhipu.AI Large Model Innovation Fund.

References

[1] Andrew G Barto and Sridhar Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete event dynamic systems*, 13:341–379, 2003. 2

[2] Suneel Belkhale, Tianli Ding, Ted Xiao, Pierre Sermanet, Quon Vuong, Jonathan Tompson, Yevgen Chebotar, Debidatta Dwibedi, and Dorsa Sadigh. Rt-h: Action hierarchies using language. *arXiv preprint arXiv:2403.01823*, 2024. 3, 6

[3] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 3, 6

[4] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, et al. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*, 2023. 3

[5] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. 3

[6] Yinpei Dai, Jayjun Lee, Nima Fazeli, and Joyce Chai. Racer: Rich language-guided failure recovery policies for imitation learning, 2024. 2

[7] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023. 3, 6

[8] John H Flavell. Metacognition and cognitive monitoring: A new area of cognitive–developmental inquiry. *American psychologist*, 34(10):906, 1979. 1

[9] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, Xiaodi Yuan, Pengwei Xie, Zhiao Huang, Rui Chen, and Hao Su. Maniskill2: A unified benchmark for generalizable manipulation skills. In *International Conference on Learning Representations*, 2023. 2

[10] Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado Van Hasselt, and David Silver. Distributed prioritized experience replay. *arXiv preprint arXiv:1803.00933*, 2018. 2

[11] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 3, 6

[12] Daeyeol Lee, Hyojung Seo, and Min Whan Jung. Neural basis of reinforcement learning and decision making. *Annual review of neuroscience*, 35(1):287–308, 2012. 1

[13] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024. 3

[14] Jacky Liang, Wenlong Huang, Fei Xia, Peng Xu, Karol Hausman, Brian Ichter, Pete Florence, and Andy Zeng. Code as policies: Language model programs for embodied control. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 9493–9500. IEEE, 2023. 2, 3

[15] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 5

[16] Jiaming Liu, Chenxuan Li, Guanqun Wang, Lily Lee, Kaichen Zhou, Sixiang Chen, Chuyan Xiong, Jiaxin Ge, Renrui Zhang, and Shanghang Zhang. Self-corrected multimodal large language model for end-to-end robot manipulation. *arXiv preprint arXiv:2405.17418*, 2024. 2

[17] Kehui Liu, Zixin Tang, Dong Wang, Zhigang Wang, Bin Zhao, and Xuelong Li. Coherent: Collaboration of heterogeneous multi-robot system with large language models. *arXiv preprint arXiv:2409.15146*, 2024. 2

[18] Qian Liu, Xiaosen Zheng, Niklas Muennighoff, Guangtao Zeng, Longxu Dou, Tianyu Pang, Jing Jiang, and Min Lin. Regmix: Data mixture as regression for language model pre-training. *arXiv preprint arXiv:2407.01492*, 2024. 7

[19] Jingxian Lu, Wenke Xia, Dong Wang, Zhigang Wang, Bin Zhao, Di Hu, and Xuelong Li. Koi: Accelerating online imitation learning via hybrid key-state guidance. *arXiv preprint arXiv:2408.02912*, 2024. 1

[20] Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021. 2, 5

[21] Abhishek Padalkar, Acorn Pooley, Ajinkya Jain, Alex Bewley, Alex Herzog, Alex Irpan, Alexander Khazatsky, Anant Rai, Anikait Singh, Anthony Brohan, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023. 3

[22] Xincheng Pang, Wenke Xia, Zhigang Wang, Bin Zhao, Di Hu, Dong Wang, and Xuelong Li. Depth helps: Improving pre-trained rgb-based policy with depth information injection. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7251–7256. IEEE, 2024. 3

[23] Shubham Pateria, Budhitama Subagdja, Ah-hwee Tan, and Chai Quek. Hierarchical reinforcement learning: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 54(5): 1–35, 2021. 2

[24] Shreyas Sundara Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. Planning with large language models via corrective re-prompting. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022. 2

[25] Roger C Schank. What we learn when we learn by doing. Technical report, Technical report, 1995. 1

[26] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. 1, 2

[27] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey

Hejna, Tobias Kreiman, Charles Xu, et al. Octo: An open-source generalist robot policy. *arXiv preprint arXiv:2405.12213*, 2024. 3, 6

[28] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 3

[29] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023. 2

[30] Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*, 2023. 1, 2

[31] Wenke Xia, Dong Wang, Xincheng Pang, Zhigang Wang, Bin Zhao, Di Hu, and Xuelong Li. Kinematic-aware prompting for generalizable articulated object manipulation with llms. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2073–2080, 2024. 2

[32] Chuyan Xiong, Chengyu Shen, Xiaoqi Li, Kaichen Zhou, Jiaming Liu, Ruiping Wang, and Hao Dong. Aic mllm: Autonomous interactive correction mllm for robust robotic manipulation. *arXiv preprint arXiv:2406.11548*, 2024. 2

[33] Jiasheng Ye, Peiju Liu, Tianxiang Sun, Yunhua Zhou, Jun Zhan, and Xipeng Qiu. Data mixing laws: Optimizing data mixtures by predicting language modeling performance. *arXiv preprint arXiv:2403.16952*, 2024. 7

[34] Naoki Yokoyama, Alex Clegg, Joanne Truong, Eric Undersander, Tsung-Yen Yang, Sergio Arnaud, Sehoon Ha, Dhruv Batra, and Akshara Rai. Asc: Adaptive skill coordination for robotic mobile manipulation. *IEEE Robotics and Automation Letters*, 9(1):779–786, 2023. 2

[35] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pages 1094–1100. PMLR, 2020. 1, 2

[36] Jia Zeng, Qingwen Bu, Bangjun Wang, Wenke Xia, Li Chen, Hao Dong, Haoming Song, Dong Wang, Di Hu, Ping Luo, et al. Learning manipulation by predicting interaction. *arXiv preprint arXiv:2406.00439*, 2024. 3

[37] Junjie Zhang, Chenjia Bai, Haoran He, Wenke Xia, Zhigang Wang, Bin Zhao, Xiu Li, and Xuelong Li. Sam-e: leveraging visual foundation model with sequence imitation for embodied manipulation. *arXiv preprint arXiv:2405.19586*, 2024. 3

[38] Tony Z Zhao, Vikash Kumar, Sergey Levine, and Chelsea Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023. 2