Phoenix: A Motion-based Self-Reflection Framework for Fine-grained Robotic Action Correction (Supplementary Material)

1. Implementation Details

1.1. Dual-process Motion Adjustment Mechanism

Training Details. In this mechanism, we construct a motion prediction module to efficiently obtain the initial motion prediction and a motion correction module to provide comprehensive motion adjustment. All of our models are fine-tuned based on the LLaVA1.5 framework [3], which encompasses three essential components: (1) a vision encoder utilizing the capabilities of the CLIP-Large model [4], which operates at a resolution of 336x336 and utilizes a patch size of 14, (2) a two-layer MLP projector that facilitates the fusion of visual and linguistic modalities, and (3) a language model, derived from the open-source Vicuna-v1.5 [1], building on the LLaMA2 foundation. We fine-tune the projector and train the LoRA layer [2] across each transformer attention block. The learning rates are set at 1e-5 for the projector layer and 1e-4 for the LoRA layer, with a LoRA alpha of 256 and a dimension of 128. The motion prediction module undergoes training over five epochs on the motion instruction dataset with a batch size of 16. The motion correction module is trained for 20 epochs on the correction dataset, also with a batch size of 16.

Motion Instruction Dataset from Expert Demonstrations. To empower the motion prediction module with the comprehension of manipulation tasks, we construct a motion instruction dataset from expert demonstrations to obtain over 160,000 pairs of motion instructions and observations. Due to the limited inference speed of MLLMs, it is difficult to utilize the motion instruction from MLLMs for real-time robotic control. Therefore, during motion instruction annotations, we aggregate 4 timestep robotic actions to form a single temporal robotic action and annotate motion instruction. We demonstrate the automatic motion instruction annotation process in Figure 1. We first filter the temporal robotic action to obtain the dominant direction with a threshold of 0.3. If an action exhibits more than one direction exceeding the threshold, the top two directions are selected as the dominant directions. Further, we obtain the gripper action from the temporal robotic action and combine the gripper action with the motion instruction. Furthermore, we incorporate the instruction to "make slight adjustments to gripper position" to model the temporal robotic actions that fall below the threshold. Utilizing the automated construction methodology, we develop a diverse set of 37 distinct motion instructions. These instructions serve as a comprehensive guide for enhancing the precision of subsequent robotic action predictions.

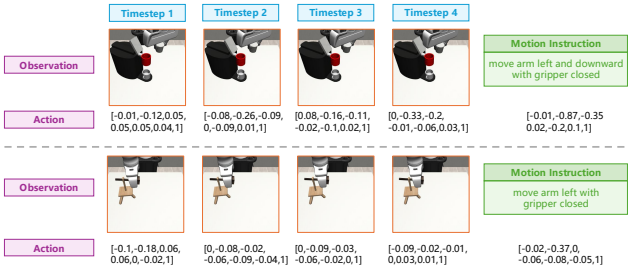


Figure 1. The motion instruction dataset annotation process.

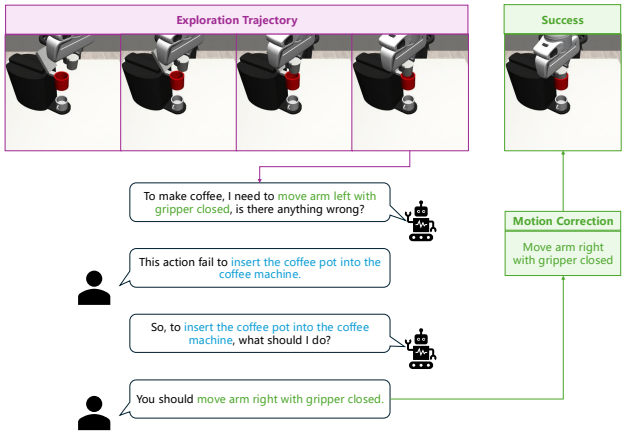


Figure 2. The demonstration of online human intervention data collection process.

Motion Correction Dataset. To equip the motion correction module with the capabilities for failure correction, we build a comprehensive correction dataset to provide se-

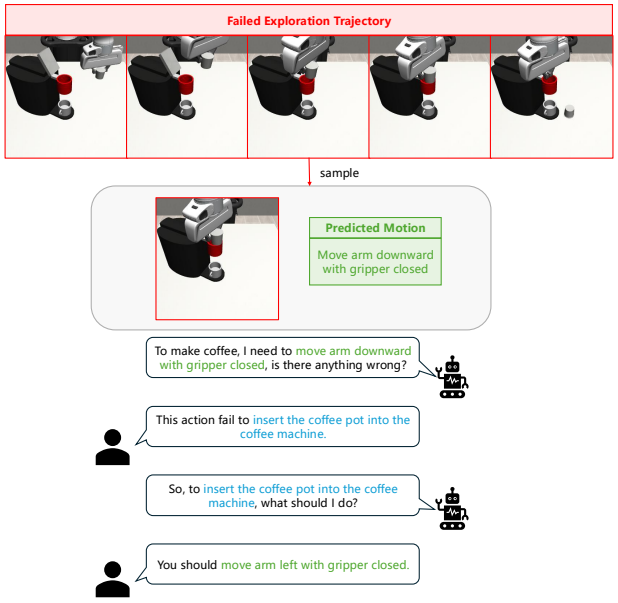


Figure 3. The demonstration of offline human annotation data collection process.

mantic reflection and motion adjustment annotation.

The online human intervention data are collected as shown in Figure 2. When the robot interacts with the environment, the human checks the task execution situation in real-time. When a failure occurs, humans provide semantic reflection and adjust motion instructions. Consequently, the robot corrects its fine-grained actions based on the adjusted motion instruction through the implementation of a low-level diffusion policy.

The offline human annotation data are collected as shown in Figure 3. We first deploy the motion prediction module to explore the environment and record the predicted motion instruction. For these collected trajectories, we sample the trajectories every 30 timestep, offering semantic feedback and adjusting the motion instructions accordingly.

1.2. Motion-conditioned Diffusion Policy.

To convert the coarse-grained motion instruction into fine-grained, high-frequency robotic action, we train a multi-task, motion-conditioned diffusion policy. We take both the image observation and robotic proprioceptive as input, the image observation shape is 84, and the robotic proprioceptive consists of end-effector position, end-effector rotation, and the gripper width. To enhance the model's temporal perception capabilities, thereby improving its ability to predict actions that adhere to the motion instructions, we integrate historical information from past 5 time steps with a temporal attention mechanism to extract temporal information. Subsequently, the observation features endowed with temporal information are used as conditional inputs in the diffusion policy. We employ 500 expert demonstrations for each task to compose the training dataset. This dataset is then used to train the diffusion policy over 200 epochs, utilizing a learning rate of 3e-4.

The learnable motion codebook is proposed to capture the discriminative features of various motion instructions. In most cases, the motion instruction predicted by MLLMs could be directly retrieved from the dictionary to obtain the corresponding language feature. However, when the predicted motion instruction is not in the dictionary, we utilize a clip text encoder to calculate the similarities between the predicted motion instruction and motion instructions in the dictionary, selecting the closest motion instruction to obtain the index.

Motion Correction	Codebook	SR
×	×	44.4%
×	✓	46.9%
✓	×	48.2%
✓	✓	57.8%

Table 1. Ablation result of codebook

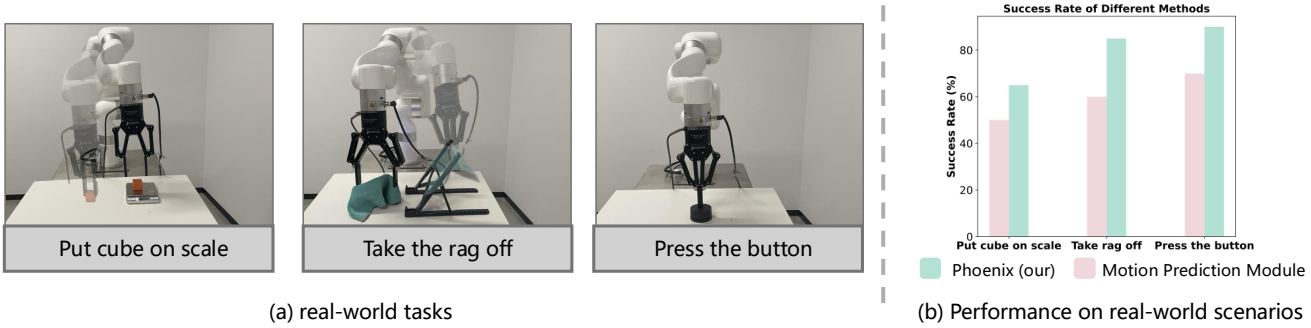
1.3. Evaluation Tasks

We demonstrate 9 simulation manipulation tasks in Figure 5, which include long-horizon manipulation tasks such as "Coffee" and "ThreePieceAssembly", and fine-grained manipulation tasks such as "Threading". By evaluating our framework on these tasks, we could verify the effectiveness of our method.

2. Ablation Results of Motion Codebook

In this work, we train a motion codebook to provide discriminative motion instruction features for motion-conditioned policy. As demonstrated in Table 1, the policy guided by the motion codebook can better adhere to motion instructions, thus achieving better performance in manipulation tasks (44.4% v.s. 46.9%). Besides, benefiting from the discriminative motion instruction feature, our model could correct its action to achieve better performance (48.2% v.s. 57.8%) when the motion correction module is proposed to correct motion instruction.

Furthermore, we employ the CLIP model [4] to extract features from the motion instructions, with the resulting similarity matrix presented in Figure 6(a). Additionally, we extract the motion instruction feature from our motion codebook, and the corresponding similarity matrix is displayed in Figure 6(b). The similarity matrix indicates that the CLIP model struggles to effectively provide discriminative motion instruction features, as the representational



(a) real-world tasks

(b) Performance on real-world scenarios

Figure 4. The real-world experiments. The results prove the generalization ability of our framework in real-world scenarios.

similarity between any two features exceeds 90%. In contrast, our learnable motion codebook offers discriminative representations, which facilitates the understanding of textual information for the low-level diffusion policy, thereby enhancing precise robotic action prediction.

3. More Real-world Experiments Results

We also prove the effectiveness of our method in rule-based manipulation policy with an xArm robot arm. As shown in Figure 4(a), we conduct experiments on three tasks: putting the cube on the scale, taking the rag off, and pressing the button. For each manipulation task, we collect 80 trajectories with corresponding motion instructions. To deploy the MLLMs in real-world scenarios, we fine-tuned a TinyLLaVA-OpenELM-450M-SigLIP-0.89B model [5] to operate at a frequency of 3Hz on a 10G 4070. We also replace the diffusion policy with a rule-based operation to execute the robotic actions to adhere to the motion instructions. During the inference process, we introduced human-in-the-loop interventions to manually correct failure situations and collect corresponding refined interaction trajectories. We collect 20 refined trajectories per task, which serve as the training dataset for the motion correction model to implement a motion-based self-reflection framework.

We conduct 20 trials and report the average success rate results in figure 4(b), the results prove that the motion prediction module could leverage the perceptual and inferential capabilities of MLLMs for manipulation tasks. Besides, our motion-based self-reflection model further significantly enhances the success rate with comprehensive motion adjustment, demonstrating the effectiveness of our approach in real-world scenarios.

In real-world experiments, we fine-tune a TinyLLaVA-OpenELM-450M-SigLIP-0.89B model [5] to predict the motion instruction. We employ a third-person perspective Realsense D435 camera to acquire observational images. These images are subsequently center-cropped to shape of 384x384 and inputted into the finely-tuned TinyLLaVA model to derive motion instructions. We collect 8 motion instructions: “move arm upward”, “move arm downward”, “move arm right”, “move arm left”, “move arm forward”, “move arm backward”, “open the gripper” and “close the gripper”. Each movement instruction directs the arm to move 2 cm toward the target direction.

References

[1] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 1

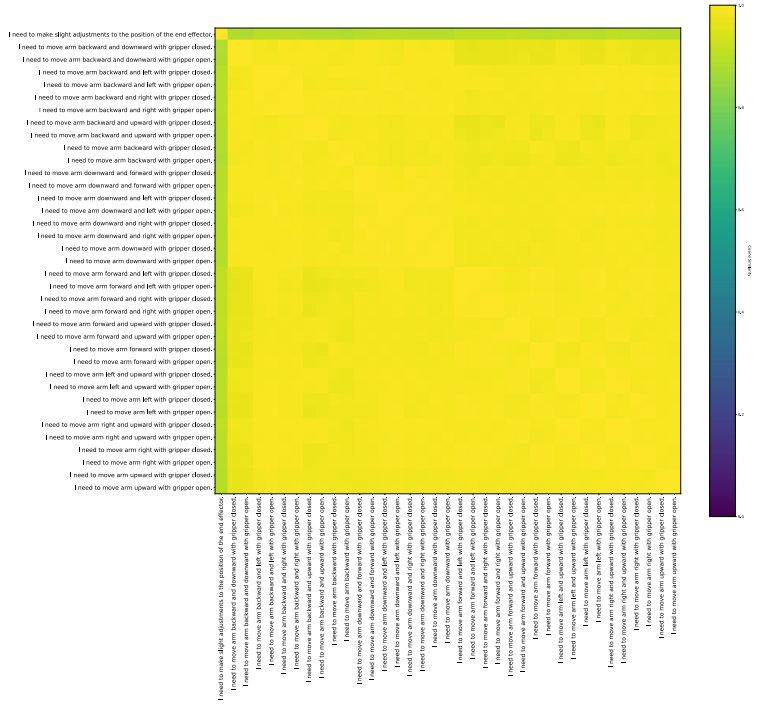
[2] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 1

[3] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. 1

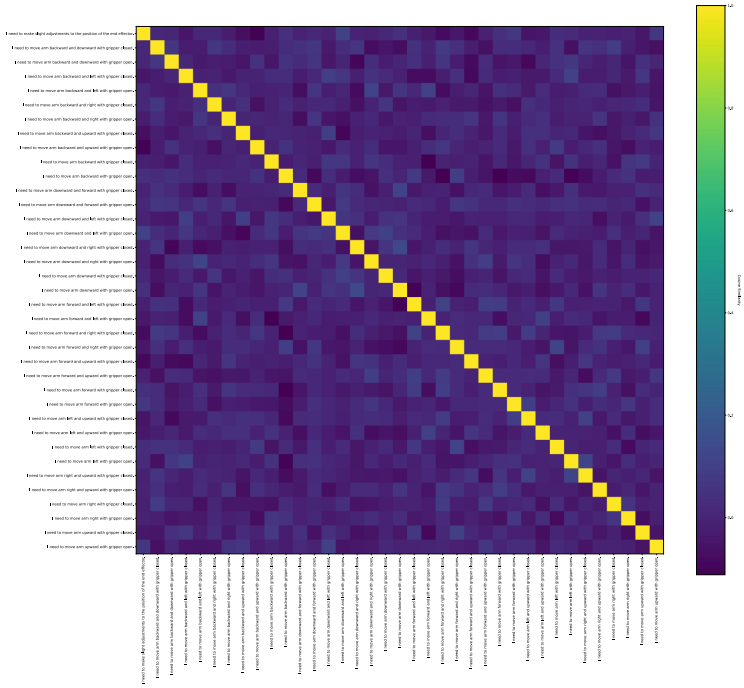
[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2

[5] Baichuan Zhou, Ying Hu, Xi Weng, Junlong Jia, Jie Luo, Xien Liu, Ji Wu, and Lei Huang. Tinyllava: A framework of small-scale large multimodal models. *arXiv preprint arXiv:2402.14289*, 2024. 3

Figure 5. The motion instruction dataset annotation.



(a) The similarity of pretrained clip feature



(b) The similarity of codebook feature

Figure 6. The similarity matrix of different motion instruction feature.