Supplementary Material

Yuanqi Yao[1,7*] Siao Liu[2*] Haoming Song[1,3*] Delin Qu[1,2] Qizhi Chen[1,4]
Yan Ding[1] Bin Zhao[1,5] Zhigang Wang[1] Xuelong Li[6] Dong Wang[1†]
[1]Shanghai AI Laboratory [2]Fudan University [3]Shanghai Jiao Tong University [4]Zhejiang University
[5]Northwestern Polytechnical University [6]TeleAI, China Telecom Corp Ltd [7]INSAIT, Sofia University

1. Overview

In this supplementary material, we provide more details of PPL, organized as follows:
In Section 2, we present the details of our role-based approach to constructing the large-scale skill dataset, corresponding to Section 5.1 of the main body.
In Section 3, we present some ablation studies, including semantically-rich language instruction experiments.
In Section 4, we provide experimental details including Hyper-parameter setting for all methods employed in our experiments, supplementing Section 5 of the main body.

2. Skill Dataset Details

2.1. Construction of skill dataset

To facilitate knowledge reuse and transfer across skills, we adopt a role-based approach for decomposing simulation datasets like MimicGen [3] and LIBERO [2], transitioning from task-level to skill-level demonstrations. As mentioned in MimicGen, each MimicGen task comprises a sequence of object-centric subtasks [3] — we aim to parse every task in the source dataset into skills, where each task corresponds to a set of skills. Specifically, in universal robot simulation environment, we can easily access contact information between objects and environment, inter-object interactions, and also object-gripper interactions, which enables us to establish success metrics for each subtask. By running through the demonstration sets in the simulation environment, we could identify the ending timesteps and completion boundaries for each subtask, enabling us to decompose task-level demonstrations into skill-level demonstrations. When deployed in real-world settings, this can be implemented by sensor signals or manual annotations.

2.2. Specific Examples

In this section, we will introduce multiple skills contained in each task and explain the success metric for each skill.

- **Stack**: (1) grasp the red cube; (2) place the red cube. The grasp skill is identified by detecting contact between gripper and cube. The place skill is determined when two conditions are met: the gripper releases contact with the cube and contact is established between the two cubes.
- **StackThree**: (1) grasp the red cube; (2) place the red cube; (3) grasp the blue cube; (4) place the blue cube. Similar to Stack, the grasp skill is identified by the contact detecting. The place skill is determined when two conditions are met: the gripper releases contact with the cube and contact is established between the two cubes.
- **Square**: (1) grasp the square; (2) place the square. Similar to Stack, the grasp skill is identified by the contact detecting. The place skill is determined when two conditions are met: the gripper releases contact with the square and contact is established between the two squares.
- **Coffee**: (1) grasp the coffee mug; (2) place the coffee mug; (3) close the machine lid. The grasp skill is identified by the contact detecting. The place skill is determined when two conditions are met: the gripper releases contact with the coffee mug and contact is established between the coffee mug and the coffee machine. The close skill is detected through the angle of the machine lid.
- **Mug Cleanup**: (1) open the drawer; (2) grasp the mug; (3) place the mug. The open skill is determined by monitoring the drawer's displacement. The grasp skill is identified by gripper-object contact. The place skill is determined by both gripper-object and object-object contact.
- **Three Piece Assembly**: (1) grasp the piece 1; (2) place the piece 1; (3) grasp the piece 2; (4) place the piece 2. The open skill is determined by monitoring the drawer's displacement. The grasp skill is identified by the gripper-object contact detecting. The place skill is determined by both gripper-object and object-object contact.
- **Coffee Preparation**: (1) grasp the mug; (2) open the machine lid; (3) open the drawer; (4) grasp the coffee mug; (5) place the coffee mug; (6) close the machine lid. The grasp skill is identified by the gripper-object contact. The place skill is determined by both gripper-object and

*Equal contribution: yaoyuanqi@pjlab.org.cn.
†Corresponding author: dongwang.dw93@gmail.com.

object-object contact. The open skill is determined by monitoring the drawer's displacement. The close skill is detected through the angle of the machine lid.

3. Ablation Studies

3.1. Effect of Motion-Aware Prompt Query

In this section, to further demonstrate the effectiveness of our motion-aware prompting (MAP) module, we construct language instructions with different densities (semantically-rich/brief), and analyze the impacts on manipulation tasks using these text-only queries and our flow-text querie. Specifically, since language instruction cannot effectively provide the motion information needed by robots, we introduce optical flow in this paper to capture motion-aware information, thereby enabling the modeling of primitives.

Task	Query Type	Succ. Rate
Task 1	Brief	0.61 ± 0.03
	Semantically-rich	0.60 ± 0.04
	Brief w/ Flow (ours)	$\mathbf{0.99 \pm 0.03}$
Task 2	Brief	0.57 ± 0.08
	Semantically-rich	0.53 ± 0.04
	Brief w/ Flow (ours)	$\mathbf{0.62 \pm 0.02}$

Table 1. Effect of query type and instruction density.

Task	Query Type	Instruction
Task 1	Brief	*Grasp banana* *Place banana*
	Semantically-rich	*Reach close to banana, and then grasp banana* *Keep banana grasped, reach close to the other side and then place banana*
Task 2	Brief	*Grasp block* *Place block*
	Semantically-rich	*Reach close to block, and then grasp block8i* *Keep block grasped, reach close to the other side and then place block*

Table 2. Examples of Instructions with Different Densities.

As shown in Tab. 1 and 2, we design semantically-rich and brief language instructions across four skills and compare the manipulation performance under different query types. The experimental results demonstrate that although semantically-rich instructions provide more dense and comprehensive narrative information, they show no significant performance improvement compared to brief and effective instructions in practical experiments. This may be attributed to the fact that language embeddings merely serve to differentiate between different tasks. This phenomenon was also observed in LIBERO [2], which further discovered that there is no statistically significant difference among various language embeddings, including the task-ID embedding. Furthermore, the performance of using either semantically-rich or brief language as the prompting query consistently underperforms compared to our text-flow query, validating the effectiveness and advantages of our MAP approach.

4. Implementing Details

We employ ResNet [1] as our visual encoder, CLIP [4] as the text encoder, and RAFT [5] as flow encoder. During training, we use the Adam optimizer with an initial learning rate of 0.0005 and a decay factor of 0.2. In the multi-skill pre-training stage, we train for a total of 450 epochs in simulation and 80 epochs on the real robot. In the lifelong learning stage, we train for a total of 50 epochs in simulation and 20 epochs on the real robot.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[2] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 2

[3] Ajay Mandlekar, Soroush Nasiriany, Bowen Wen, Iretiayo Akinola, Yashraj Narang, Linxi Fan, Yuke Zhu, and Dieter Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *7th Annual Conference on Robot Learning*. 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. 2

[5] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 402–419. Springer, 2020. 2

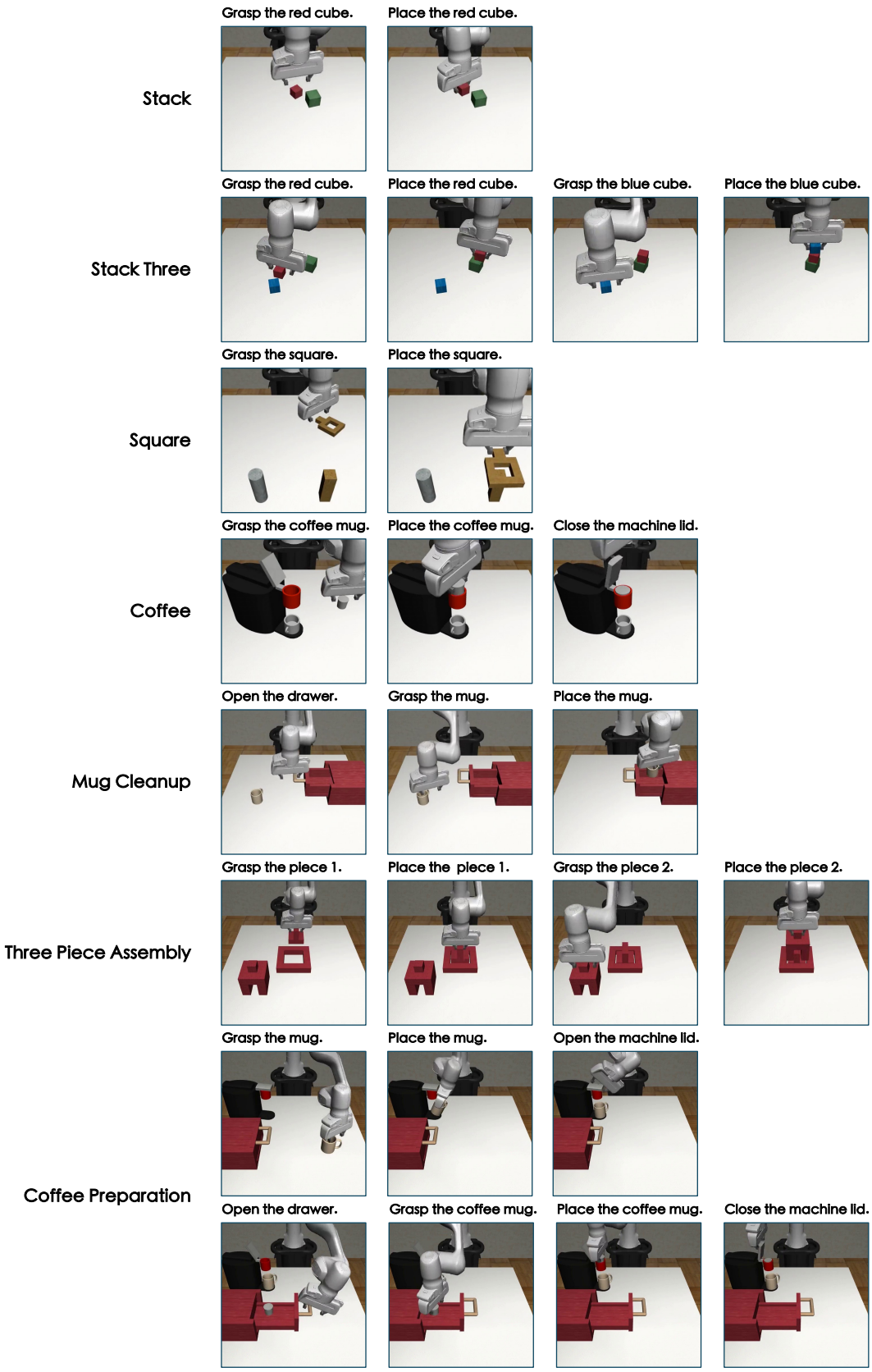

Figure 1. Example skills in our dataset corresponding to original MimicGen tasks.