Appendix

Mitigating the Human-Robot Domain Discrepancy in Visual Pre-training for Robotic Manipulation

Jiaming Zhou[1], Teli Ma[1], Kun-Yu Lin[2], Zifan Wang[1], Ronghe Qiu[1], Junwei Liang[1,3†]
[1]AI Thrust, The Hong Kong University of Science and Technology (Guangzhou)
[2]Sun Yat-sen University, [3]The Hong Kong University of Science and Technology
jia_ming_zhou@outlook.com junweiliang@hkust-gz.edu.cn

S1. Downstream Policy Learning on RLBench

Existing visual pre-training works often assess the learned representation in downstream environments using a single-task setting. However, our work diverges from this standard by also evaluating the adapted pre-trained models on the large-scale RLBench benchmark [5], where a single language-conditioned policy is learned to complete various tasks. Figure S1 shows examples of the 18 tasks and the corresponding human instructions.

On RLBench, existing works [1–4, 7] usually develop sophisticated models to model the semantics of the robot's multi-view observations and their correlations with the language commands. For example, RVT [2] utilizes four attention layers as the visual encoder to model intra-image relations and four more attention layers to model image-language correlations. In this work, we adopt the same design as RVT, but replace the visual encoder (i.e., four intra-image attention layers) with either an existing pre-trained model or our adapted one. Additionally, we employ just one attention layer to fuse the extracted image features and language features. Since RVT predicts the end-effector's actions based on the features without down-sampling the spatial dimension, we discard all spatial down-sampling operations (e.g., max-pooling) in both the pre-trained models and our adapted models. Please note that to validate the effectiveness of our adaptation method, we freeze the visual representation of the pre-trained model or our adapted model while learning the downstream policy on RLBench.

S2. Real-world Experiments

Setups. For the real-world manipulation experiments, we use a 7-DoF xArm robot arm equipped with an Inspire gripper. Visual observations are captured by an Orbbec Femto Bolt (RGB-D) camera positioned in front of and to the upper right of the robot arm. The positions of the robot arm, working area, and camera remain fixed during data collection and policy testing. Additionally, we use a DJI Osmo Action 4 camera to record videos of the policy testing.

Data Collection. We design five different real-world tasks, namely, *put fruit in plate*, *stack cups*, *put tennis in mug*, *hang mug*, and *put item in box*. For each task, we collect 40 human teleoperation demonstrations for training. Figure S2 demonstrates some examples of the collected tasks. For each demonstration, we manually move the robot arm and change the states of the gripper (i.e., open or close) to complete the target task. We record these operations and replay them to record the demonstration. We simultaneously record the robot arm end-effector state (i.e., positions in the x-axis, y-axis, and z-axis, and rotations in roll, pitch, and yaw), gripper state, and Orbbec camera RGB stream with an image size of 1280×960.

Model Designs. We train our manipulation policy network under a single-task setting, utilizing the ACT [8] framework for policy learning[1]. In addition, following RVT [2] which predicts the next key-action, we predict the following key-actions of the end-effector. The visual backbone of the network is replaced with either the pre-trained models or our adapted models. During both training and testing, the RGB images are resized to 320×240.

Models	learned params.	*pen*	*relocate*	Averaged
R3M	0M (frozen:25M)	78.0	70.0	74.0
R3M-Align-L	1.6M	**81.3**	**81.3**	**81.3** (+7.3)
R3M-Align-L *w/o lang.*	1.6M	79.3	80.7	80.0 (+6.0)

Table S1. Success rates of two tasks in **Adroit**. Removing the language-guided feature enhancement will degrade the model's performance.

S3. Additional Ablations

In this work, our adaptation method uses task description features as queries to better capture task-aware semantics

[1]We follow the implementation of https://github.com/Shaka-Labs/ACT.

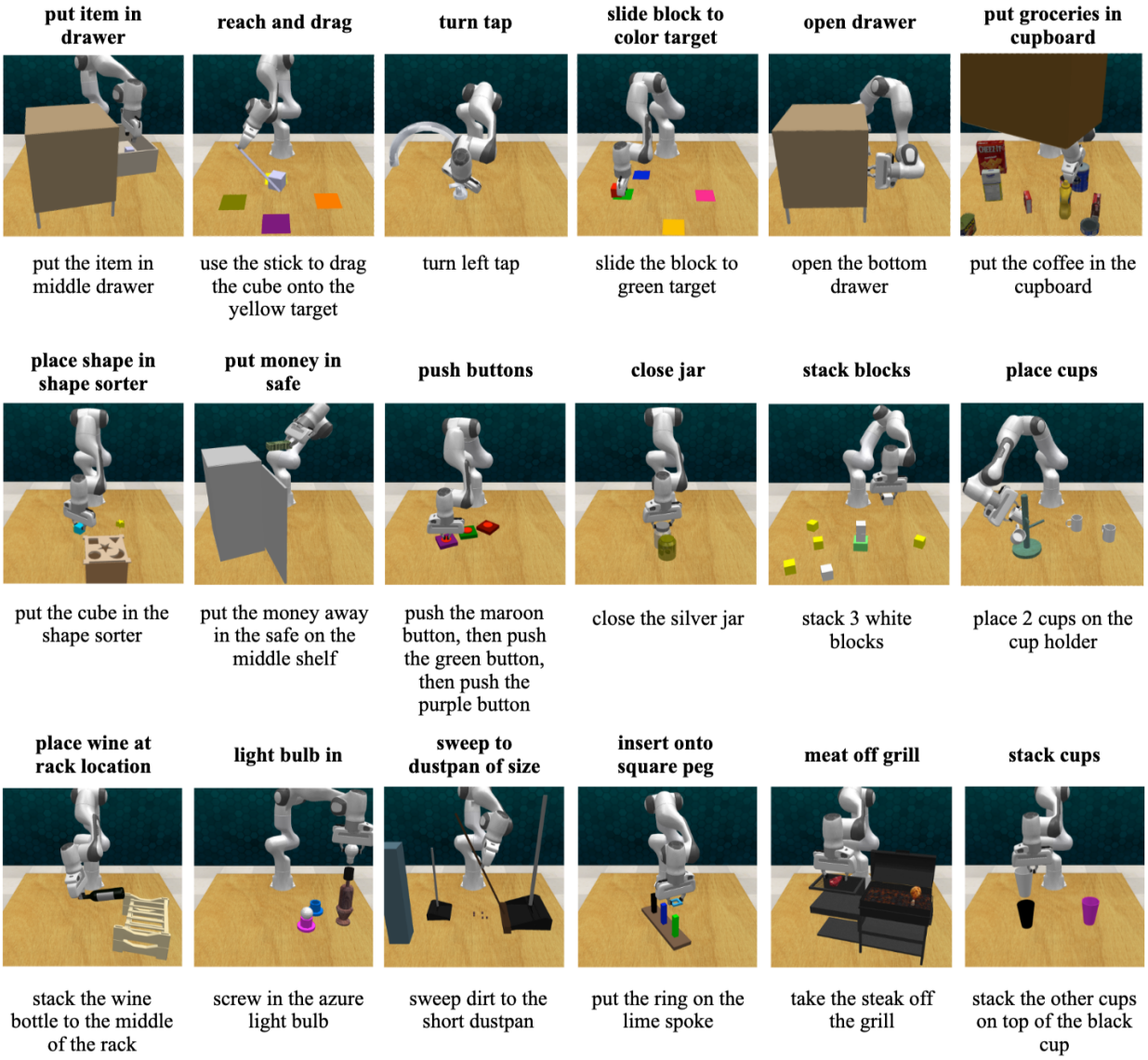

Figure S1. Examples of the 18 RLBench tasks (front view) with corresponding human instructions (sourced from [6]).

from video features. As shown at the bottom of Table S1, by removing this operation, the adapted model, i.e., R3M-Align-L *w/o lang.*, will result in performance degradation. This demonstrates that the language-guided feature enhancement is effective in promoting human-robot semantic alignment.

In Table 3, we only used the robot data from the RH20T subset we used to train R3M-PreT and R3M-ClS. For fair comparisons, we train both R3M-PreT and R3M-ClS using human and robot videos (i.e., the same amount of training data of HR-Align). Table S2 shows that our R3M-Align still outperforms R3M-PreT and R3M-ClS trained by full-data. In addition, the R3M-PreT and R3M-ClS in Table 3 are full-parameter fine-tuned, while our HR-Align is fine-tuned with parameter-efficient Adapter. To ensure a fair comparison, we instantiate R3M-PreT and R3M-ClS by inserting an adapter into frozen R3M (i.e., the same amount of learnable parameters as our R3M-Align), and train them using both human and robot data (i.e., the same amount of training data of HR-Align), denoted as R3M-PreT$_A$ and R3M-ClS$_A$. Table S3 shows that our R3M-Align still performs better. The above shows the effectiveness of our HR-align method.

R3M-PreT	R3M-ClS	R3M-Align (**Ours**)
78.1	77.5	**81.3**

Table S2. Comparisons between models when training with full data.

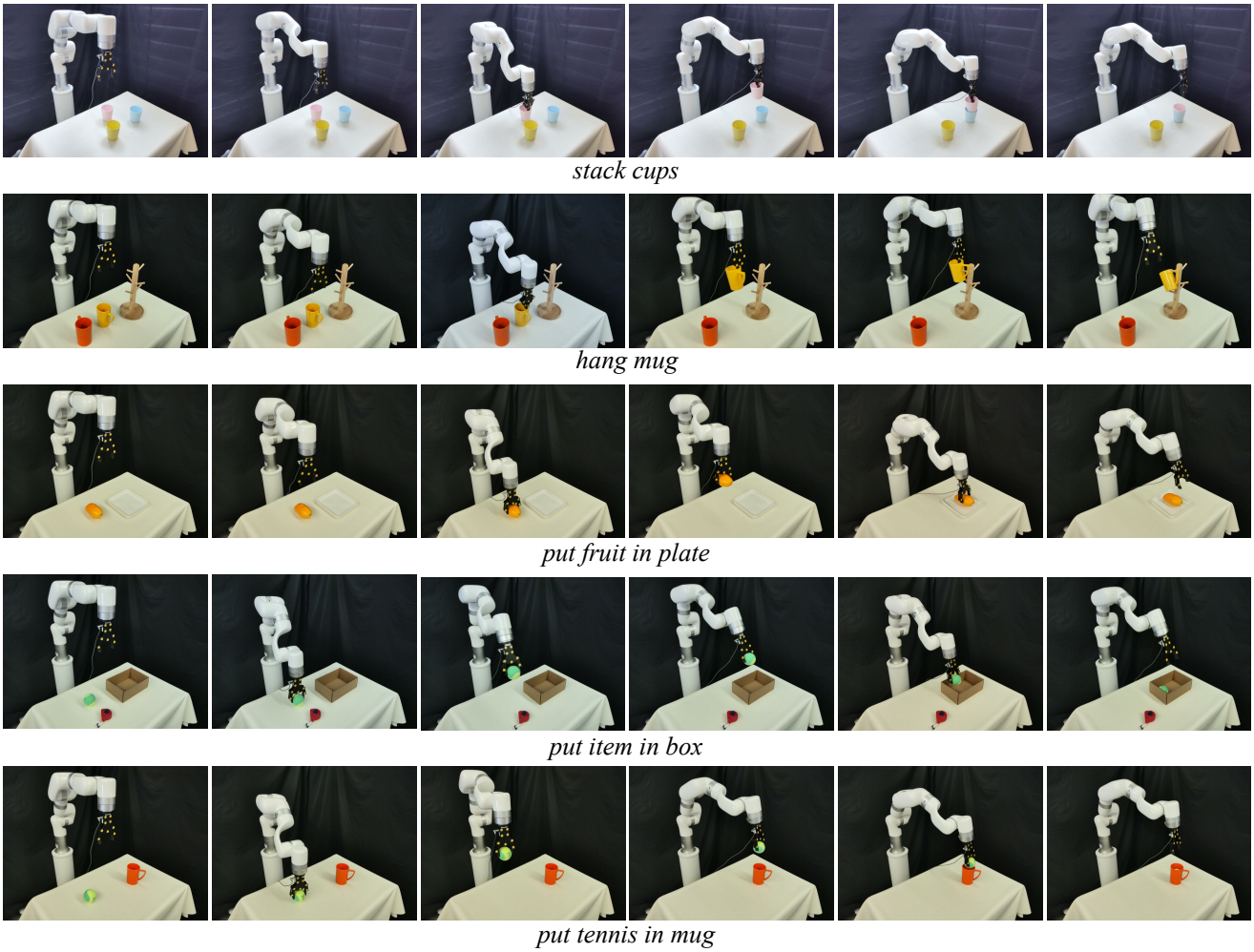

Figure S2. Examples of the five real-world tasks are shown, with each row presenting an instance of the corresponding task. For each demonstration, we provide visual observations at six different timestamps.

R3M-PreT$_A$	R3M-ClS$_A$	R3M-Align (**Ours**)
77.2	76.9	**81.3**

Table S3. Comparisons between models when training with Adapters.

References

[1] Shizhe Chen, Ricardo Garcia, Cordelia Schmid, and Ivan Laptev. Polarnet: 3d point clouds for language-guided robotic manipulation. *arXiv preprint arXiv:2309.15596*, 2023. 1

[2] Ankit Goyal, Jie Xu, Yijie Guo, Valts Blukis, Yu-Wei Chao, and Dieter Fox. Rvt: Robotic view transformer for 3d object manipulation. In *Conference on Robot Learning*, pages 694–710. PMLR, 2023. 1

[3] Haoran He, Chenjia Bai, Ling Pan, Weinan Zhang, Bin Zhao, and Xuelong Li. Large-scale actionless video pre-training via discrete diffusion for efficient policy learning. *arXiv preprint arXiv:2402.14407*, 2024.

[4] Stephen James and Andrew J Davison. Q-attention: Enabling efficient learning for vision-based robotic manipulation. *IEEE Robotics and Automation Letters*, 7(2):1612–1619, 2022. 1

[5] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J Davison. Rlbench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters*, 5(2):3019–3026, 2020. 1

[6] Teli Ma, Jiaming Zhou, Zifan Wang, Ronghe Qiu, and Junwei Liang. Contrastive imitation learning for language-guided multi-task robotic manipulation. *arXiv preprint arXiv:2406.09738*, 2024. 2

[7] Mohit Shridhar, Lucas Manuelli, and Dieter Fox. Perceiver-actor: A multi-task transformer for robotic manipulation. In *Conference on Robot Learning*, pages 785–799. PMLR, 2023. 1

[8] 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. 1