Supplementary Materials For Human-Agent Joint Learning for Efficient Robot Manipulation Skill Acquisition

Shengcheng Luo[1*], Quanquan Peng[1*], Jun Lv[1*], Kaiwen Hong[2],
Katherine Rose Driggs-Campbell[2], Cewu Lu[1], Yong-Lu Li[1]

APPENDIX

I. Implementation details

Here we lay down the details of the data collection, training, and testing process. More technical details are given here to illustrate our method and implementations better.

A. Shadow Hand and Parallel Gripper Teleoperate System

To adapt to Isaac Gym and our vision system, we made certain modifications to the XML file of the Shadow Hand. We followed [1–3], removed the entire arm part, and added six degrees of freedom to the base mount of the Shadow Hand. This allows it to move freely in the virtual environment without depending on a base. Similarly, to obtain the rigid body Jacobian matrices of the five fingertips of the Shadow Hand, we added a massless rigid body to the tips of all five fingers of the Shadow Hand. This facilitates direct inverse kinematics calculations for the entire finger. In inverse kinematics (IK) calculations, we employed the Damped Least Squares (DLS) method [4, 5], this approach helps to prevent instability issues when approaching singularity points. Additionally, the DLS method supports real-time applications because it can provide fast and stable solutions, which is particularly crucial for teleoperation systems. Focusing solely on the five fingertips and wrist is regarded as the most balanced approach between computational efficiency and the precision required for complex hand movements in real-time applications. The system operates on a computer with an RTX 4070 graphics card and a monitor.

To mitigate the accumulation of errors, the process involves mapping hand motion from the real world into the virtual environment and then comparing each action with the action from the previous frame to calculate a delta action. The reason for calculating delta action is to identify and apply only the changes in movement from one frame to the next, rather than applying the absolute positions and orientations directly. This approach helps reduce the accumulation of errors that might occur due to discrepancies between the real-world movements and their representation in the simulated environment. By focusing on the changes (delta) rather than absolute values, the system can more accurately replicate the intended movements in the simulator, leading to more precise and consistent control of the shadow hand.

* denotes equal contribution
[1]Shanghai Jiao Tong University, [2]University of Illinois Urbana-Champaign.

B. Baselines

In this section, we provide the implementation details for BC and BC-RNN models. In Behavior Cloning (BC), the objective is to minimize $\mathbb{E}_{(s,a)\sim\mathcal{D}}||\pi_\theta(s)-a||^2$. We use a 3-layer multi-layer perception (MLP) with a ReLU activation function. All layers are fully connected layers with 128 hidden dimensions with a learning rate of $2\cdot10^{-3}$. We also use the AdamW [6] to be the optimizer. The training epoch in dexterous tasks Pick-and-Place, Articulated-Manipulation, and Tool-Use is $60, 100, 100$ separately.

As for BC-RNN, we use an LSTM as the backbone network for BC-RNN [7], which we find a slight performance improvement compared to the vanilla RNN model. Following [7], during the training phase, a state-action sequence $\{(s_i,a_i),\cdots,(s_{i+T-1},a_{i+T-1})\}$ of length T is sampled from the dataset $\mathcal{D}$ and the network will predict the action sequence based on the states as its input. During the inference phase $a_t, h_{t+1} = \pi_\theta(s_t, h_t)$ where h_t, h_{t+1} are the hidden states. Here we set the learning rate to be $2\cdot10^{-3}$, and the training epoch to be 60.

C. Diffusion-Model-Based Assistive Agent

The assistive agent's noise prediction model ε_θ's backbone network is a 4-layer multi-layer perception (MLP) with a Softplus activation function. All layers are fully connected layers with 128 hidden dimensions. Moreover, we set the diffusion steps $K=50$, $\beta_{\min}=10^{-4}, \beta_{\max}=0.1$ in Eq. 2 with Sigmoid scheduling and use Exponential Moving Average (EMA) to stabilize the training. The learning rate of ε_θ is 10^{-3}.

II. Experiment Setups

A. Tasks

Dexterous Hand Pick-and-Place aims at picking an object on the table and placing it into a container. The observation space is 24 dimensions, including the dexterous robot hand state (18-dim), the object's position (3-dim), and the container's position (3-dim). The dexterous robot hand state is the position of each fingertip (15-dim) and the wrist position (3-dim). The action space is 28 dimensions, including the state change of each joint (22-dim) and the wrist transformation (6-dim). The object's position is randomized for each attempt within a $10cm\times10cm$ square on the table.

Dexterous Hand Articulated-Manipulation aims at grasping and unscrewing the door handle to open the door. The observation space is 32 dimensions, including the dexterous

robot hand state (18-dim), the door handle's position (3-dim) and quaternion (4-dim), and the door base's position (3-dim) and quaternion (4-dim). In contrast, the action space is 28 dimensions. The door's position is randomized for each attempt within a $40cm \times 40cm$ square on the floor.

Dexterous Hand Tool-Use aims at picking a hammer and using it to drive a nail into a board. The observation space is 32 dimensions, including the dexterous robot hand state (18-dim), hammer's position (3-dim) quaternion (4-dim), and nail's position (3-dim). At the same time, the action space is 28 dimensions. The nail's position is randomized for each attempt within a $10cm \times 10cm$ square on the table.

Parallel Gripper Pick-and-Place aims at picking an object on the table and placing it into a container. The observation space is 27 dimensions, including the five rigid bodies of the gripper to object distances (15-dim), the distance between left and right grippers (3-dim), the object's position (3-dim), the distance between object and target (3-dim,) and the distance between flange and target (3-dim). The action space is 8 dimensions, including the state change of each joint (7-dim) and gripper (1-dim). The object's position is randomized for each attempt within a $10cm \times 10cm$ square on the table.

Parallel Gripper Articulated-Manipulation aims at picking an object on the table and placing it into a container. The observation space is 16 dimensions, including the five rigid bodies of gripper to object distances (15-dim), and the distance between object and target (1-dim). The action space is 7 dimensions, including the state change of each joint (7-dim). The object's position is randomized for each attempt within a $10cm \times 10cm$ square on the table.

Parallel Gripper Cube-Push aims at pushing an object on the table to the target position. The observation space is 22 dimensions, including the three rigid bodies of the gripper to object distances (9-dim), the flange's position (7-dim), the distance between object and target (3-dim,) and the distance between flange and target (3-dim). The action space is 7 dimensions, including the state change of each joint (7-dim). The object's position is randomized for each attempt within a $5cm \times 5cm$ square on the table.

B. Ablation study

We implement the shared control agent with different methods like the diffusion model and BC. BC adapts a classical way for blending policy to achieve shared control [8]. We use it in the ablation study to blend BC policy with pure human action to achieve shared control in Fig.1. Compared to the classical way which explicitly averages human action a^h and agent action a^r to get the shared action a^s, we instead use the diffusion model, which is a popular implicit model, to blend two actions. It models the process as the forward and reverse process. The forward/diffuse process is about adding Gaussian noise to human action a^h, and the reverse process uses a neural network $f(\cdot|\cdot)$ to denoise a^k to get the shared action a^s.

BC agent is trained using a specific sequence of data collection and fine-tuning steps to optimize performance across different levels of shared control. Initially, we collect data sets of 10, 10, and 20 episodes under various task conditions. These initial datasets are used to train a preliminary agent. Following this initial training phase, we employ the trained agent to assist in further data collection under three different control ratios represented by γ values of 0.25, 0.5, and 0.75. The data collected with the assistance of the agent under these γ settings are then used to fine-tune the agent.

TABLE I: Agent performance on human expert or amateur datasets.

Dexterous Hand	*Pick-and-Place*		*Articulated-Manipulation*		*Tool-Use*	
	Skilled	*Unskilled*	*Skilled*	*Unskilled*	*Skilled*	*Unskilled*
BC	0.45	0.02	0.43	0.18	0.40	0.05
BC-RNN	0.41	0.05	0.62	0.04	0.27	0.05
DP	0.71	0.01	0.68	0.10	0.81	0.03

TABLE II: Ablation study on DP performance between r.

	Pick-and-Place	*Articulated-Manipulation*	*Tool-Use*
$r = 0.0$	0.565	0.661	0.512
$r = 0.1$	**0.620**	**0.681**	**0.547**
$r = 0.2$	0.575	0.407	0.115
$r = 0.3$	0.435	0.216	0.029

As shown in Fig.1, experiments demonstrated that the success rate of an assistive agent based on BC is lower than that of an agent based on diffusion models, indicating a reduced capacity for assistance. In certain instances, the action even becomes worse at particular control ratios.

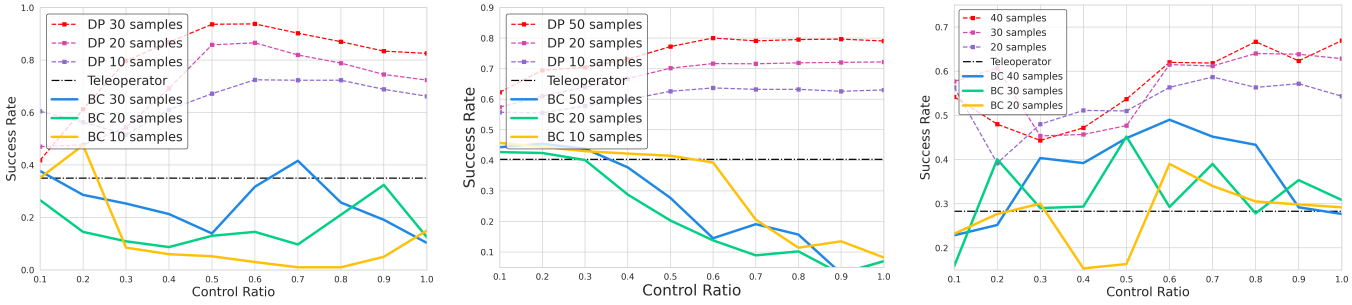
We test the performance of training with different data compositions. For a task, we gathered two manipulation datasets from both skilled and unskilled human operators. We consider operators to be skilled workers if they can practice for more than five hours and reach a success rate and efficiency comparable to those with assistive agents. As shown in Tab. I, the performance of agents trained on the dataset of unskilled operators is much lower than that on the dataset of skilled operators. Therefore, all the human operation datasets $\mathcal{H}$ we use in the main text are from skilled operators.

In our framework, r represents the modification ratio of noise between the state and action. Specifically, during the training, the noise added to state s satisfies $\varepsilon_s = u \cdot \mathcal{N}(\mathbf{0}, \mathbf{I})$ while the noise added to action a satisfies $\varepsilon_s = v \cdot \mathcal{N}(\mathbf{0}, \mathbf{I})$. Then $r = \frac{u}{v}$. We test different r as shown in Tab. II, to ensure the best agent performance. We default to using $r = 0.1$ in our model.

C. Real World Experiment

In this section, we evaluate the real-world performance of our method. We use the setup shown in Fig.2, which includes a Flexiv Rizon4 arm equipped with a gripper and two Intel RealSense D435i RGB-D cameras. One camera is mounted on the wrist of the robotic arm, while the second is positioned on the side. One task here is to pick the red pot shown in Fig.2 and place it onto the induction cooker.

During the real-world data collection phase, we estimate the human hand's pose using RGBD input. Considering the significant difference in morphology between the human hand



(a) *Pick-and-Place* (b) *Articulated-Manipulation* (c) *Tool-Use*

Fig. 1: Ablation on different dexterous agents trained with different compositions of data.

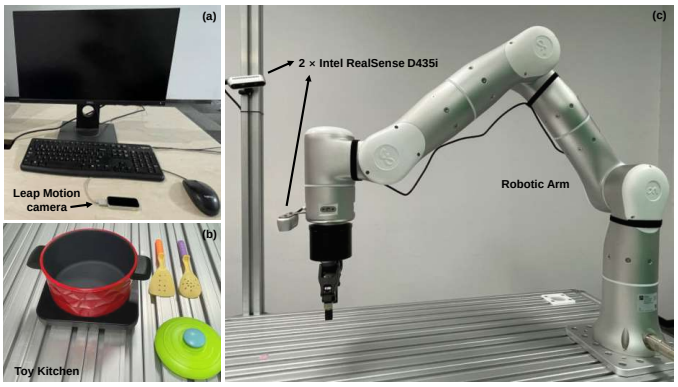


Fig. 2: Realworld Pick-and-Place Experiment. The hardware setup comprises (a) a Leap Motion camera utilized for teleoperation data collection, (b) a toy kitchen environment set up for the pick-and-place task, and (c) a Flexiv Rizon4 robotic arm equipped with a gripper and two cameras. One camera is mounted on the wrist of the robotic arm, while the second one is positioned on the side.

and a 7-DoF robotic arm, we chose to track the end effector's position by monitoring the position of the hand's wrist. Additionally, we used the action of closing or opening the human hand as the condition for determining whether to grasp or release an object. This approach leverages the greater dexterity of the human hand to enhance the control and precision of the robotic arm. We record RGB images from two camera views, joint poses (7-dim), gripper width (1-dim), the end effector's position (3-dim), and its quaternion (4-dim). The RGB images have a size of 640×480 pixels, each episode is sampled at a frequency of 10 Hz.

In real-world experiments, the network architecture is generally similar to the simulation environment's. Our input has changed from the original hand states and object states to the position and orientation of the robot arm end effector, as well as images from the first-person and third-person perspectives. We made two main modifications: 1) For the images, we used a ResNet-18 model. We used a standard ResNet-18 (without pretraining) as the encoder with its global average pooling replaced with a spatial softmax pooling to maintain spatial information. 2) We deepened the layer of the neural network, increased its hidden layer dimension, and expanded action horizon prediction from predicting the next frame action to predicting actions for the subsequent T frames, *i.e.*, $a_{t+1:t+T-1}$ (where $T = 8$).

III. Discussion and Limitation

A. Human-Machine Interface

Our approach has demonstrated success across a diverse set of Human Machine Interfaces(HMI), including:

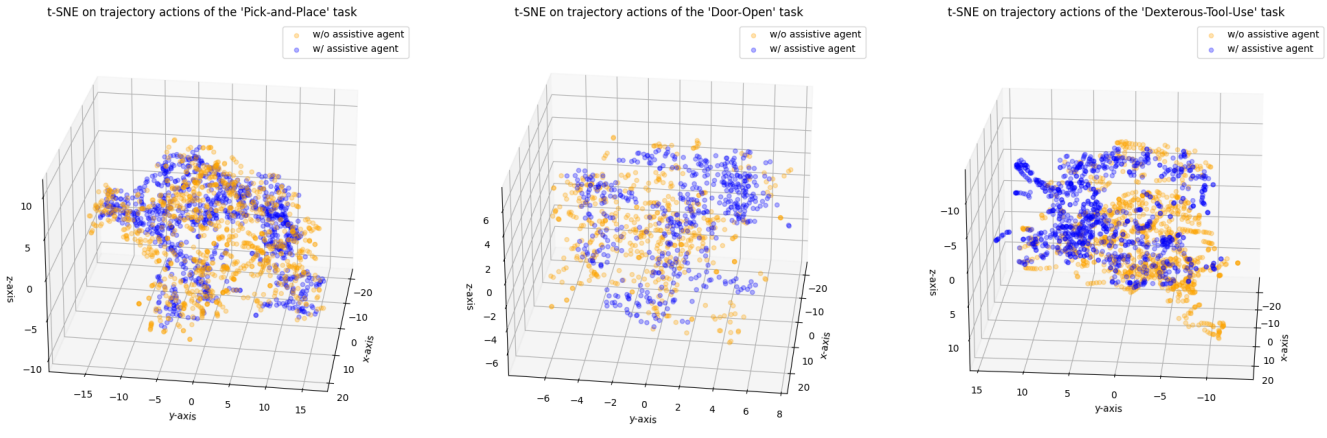
Sigma.7 Teleoperation Devices: Our system has successfully utilized Sigma devices to achieve precise control for tasks involving limited DoF. These devices require intricate control and feedback mechanisms, demonstrating our interface's robustness and effectiveness in physical UI scenarios.

RGB-D Cameras: Our system can accurately interpret spatial environments by leveraging depth perception, making it highly effective for freehand teleoperation. This capability lays the foundation for handling physical UIs with equal precision.

Virtual Reality (Meta Quest3): In VR environments, our interface provides an immersive and intuitive experience that closely mimics real-world interactions. This shows its capability to handle complex interfaces with precision and ease. As shown in Tab. III, we repeated the dexterous articulated-manipulation experiment with Leap Hand [9] in a VR environment and validated that our paradigm is applicable across different HMIs. This demonstrates the versatility of our approach, ensuring consistent operation across various human-machine interfaces.

TABLE III: Articulated-Manipulation task success rate under increasing data with Quest3.

VR Dexterous	*Articulated-Manipulation* BC	DP
$10\mathcal{H}$	0.04	0.10
$10\mathcal{H}$ + $10\mathcal{H}$	0.15	0.25
$10\mathcal{H}$ + $20\mathcal{H}$	0.26	0.26
$10\mathcal{H}$ + $30\mathcal{H}$	0.40	0.30
$10\mathcal{H}$ + $10\mathcal{S}$	0.34	0.28
$10\mathcal{H}$ + $20\mathcal{S}$	0.30	0.35
$10\mathcal{H}$ + $30\mathcal{S}$	0.44	0.63



(a) *Pick-and-Place* (b) *Articulated-Manipulation* (c) *Tool-Use*

Fig. 3: t-SNE visualization of distributions of trajectory actions (w/ and w/o assistive agent).

B. Data Analysis

We visualize the Preference Alignment [10] for dexterous hand articulated-manipulation task, as shown in Fig. 4. We find that as time progresses, the preference alignment increases across all three phases: reaching the door latch, twisting the door latch, and pulling. This indicates a growing synchronization between the user and the assistive agent throughout each stage of the task. Also, the preference alignment between the user and the assistive agent improves across different control ratios.

We use t-SNE to visualize the distribution of trajectory actions on different dexterous tasks, as shown in Fig. 3. Specifically, we have reduced the trajectory of actions to three dimensions using t-SNE, for both data collected by human operators with and without our system. To ensure a fair comparison, we uniformly sampled the same number of actions across both scenarios. We find that the distribution of the same task tends to cluster in the same space, whether with or without an assistive agent. This indirectly demonstrates that our system can enhance data collection speed and efficiency without compromising data quality.

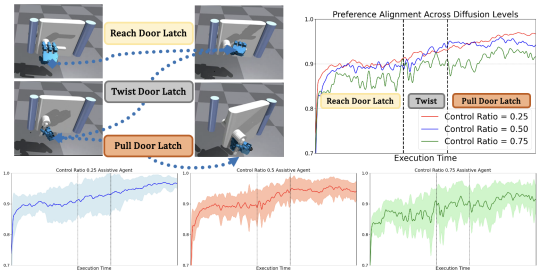


Fig. 4: The articulated-manipulation task consists of three phases: reaching the door latch, twisting it to the correct angle, and pulling it. We plotted the dot product between the user input action and the assistive agent's output action(Preference Alignment). In the plot, the red, green, and blue lines represent control ratios of 0.25, 0.5, and 0.75, respectively.

C. Limitations

Our current system's task-specific assistive agent, while effective for certain applications, does have its limitations. It currently can not handle tasks that involve multiple subtasks or targets that change dynamically, as these scenarios often require more flexibility, including the ability to adjust the control ratio throughout the sequence. To broaden the system's applicability, integrating large language models could also allow it to handle a wider range of robot learning tasks by conditioning on text input. Additionally, we think adding a learnable control ratio adjustment mechanism, especially for long-horizon tasks, could improve the system's adaptability and efficiency. We believe that our proposed joint-learning framework has the potential to leverage more powerful multi-task diffusion policies, allowing it to handle more complex scenarios in future enhancements.

ACKNOWLEDGMENT

The authors would like to thank anonymous reviewers for helping improve exposition. This work is supported in part by the National Natural Science Foundation of China under Grants 62306175.

REFERENCES

[1] T. Wu, M. Wu, J. Zhang, Y. Gan, and H. Dong, "Graspgf: Learning score-based grasping primitive for human-assisting dexterous grasping," 2023.

[2] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system," *arXiv preprint arXiv:2307.04577*, 2023.

[3] Y. Qin, Y.-H. Wu, S. Liu, H. Jiang, R. Yang, Y. Fu, and X. Wang, "Dexmv: Imitation learning for dexterous manipulation from human videos," in *European Conference on Computer Vision*. Springer, 2022, pp. 570–587.

[4] S. R. Buss and J.-S. Kim, "Selectively damped least squares for inverse kinematics," *Journal of Graphics tools*, vol. 10, no. 3, pp. 37–49, 2005.

[5] A. N. Pechev, "Inverse kinematics without matrix inversion," in *2008 IEEE International Conference on Robotics and Automation*. IEEE, 2008, pp. 2005–2012.
[6] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.
[7] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," *arXiv preprint arXiv:2108.03298*, 2021.
[8] A. D. Dragan and S. S. Srinivasa, "A policy-blending formalism for shared control," *The International Journal of Robotics Research*, vol. 32, no. 7, pp. 790–805, 2013.
[9] K. Shaw, A. Agarwal, and D. Pathak, "Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning," *arXiv preprint arXiv:2309.06440*, 2023.
[10] H. J. Jeon, D. P. Losey, and D. Sadigh, "Shared autonomy with learned latent actions," *arXiv preprint arXiv:2005.03210*, 2020.