Towards Scaling Robotic Manipulation: Leveraging World Knowledge as Automated Teleoperator

Supplementary Material

Paper-ID [127]

This supplementary material provides additional implementation details and experimental results for our paper, *Towards Scaling Robotic Manipulation: Leveraging World Knowledge as Automated Teleoperator*. The appendix elaborates on system components, training configurations, and evaluation protocols, and includes additional experiments that further validate and extend the findings in the main paper. We additionally provide a demo video of PaPA, illustrating the real-world data collection process, including the cyclic data collection pipeline described in Appendix B7.

APPENDIX

A. Implementation and Experimental Details

We give details on the pipeline implementation and the experimental setup details. These include the model implementation in perception stage, the training hyperparameters of behavior cloning policies, and the experimental details on cognitive effort measurement and the workload survey.

1) ***Perception Module Implementations****:* We implement the perception module using zero-shot multimodal models. For vision-language reasoning, we adopt the off-the-shelf InternVL3-8B [11] without any task-specific fine-tuning. For grounded perception, we similarly leverage Grounded-SAM [8] in a zero-shot manner. This design choice is motivated by efficiency and reduced engineering overhead, and is further analyzed in the ablation studies in Appendix B6. Together, these models provide robust grounded object localization without requiring additional training.

2) ***Training Hyperparameter****:* We provide the training hyperparameters for all behavior cloning policies used in our experiments in Table VI. To ensure fair comparison, we use consistent training settings across different datasets and baselines unless explicitly stated otherwise.

3) ***VLM Prompts and Reasoning Examples****:* We representative examples of the prompts and reasoning outputs used in the vision-language reasoning stage of PaPA. These examples provide insight into how high-level language instructions and visual observations are processed to produce structured semantic guidance, which is subsequently used for trajectory planning.

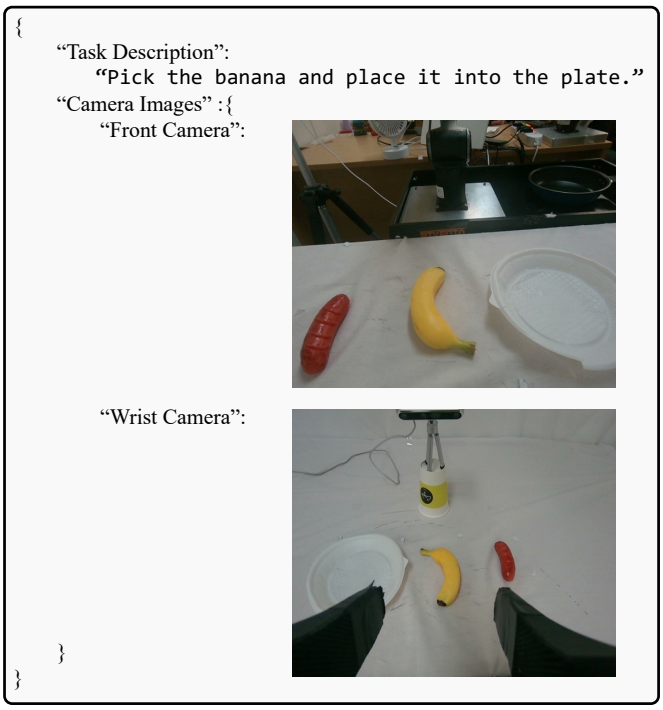
Specifically, the full prompt template used for vision-language reasoning is shown in Fig. 6, while an example of the reasoning input and output is illustrated in Fig. 1. During inference, {TASK_DESCRIPTION} in the prompt is instantiated with the given task instruction. These examples clarify the interface between vision-language reasoning and grounded segmentation, making the role of world knowledge in PaPA explicit.

4) ***Cognitive Effort Details****:* We evaluate the cognitive effort involved in data collection using a PASAT-style dual-task paradigm [2]. During data collection, participants simultaneously perform a verbal cognitive task while operating the given data collection method. Specifically, they are verbally presented with a randomly generated digit sequence of length between 5 and 7 and are instructed to report the sum of the last two digits after the sequence ends. For example, given the audio sequence "1 3 2 7 5 1 8", the correct response is "9" (1+8). We adopt an auditory modality to avoid interfering with the visual and motor channels required for manipulation, while still imposing sustained cognitive load. Participants are not informed of the sequence length in advance, and the end of each trial is indicated only by the termination of the audio stream. This design requires continuous attention to the digit stream and maintenance of the two most recent digits in working memory. Detailed results and analysis are provided in Appendix B4.

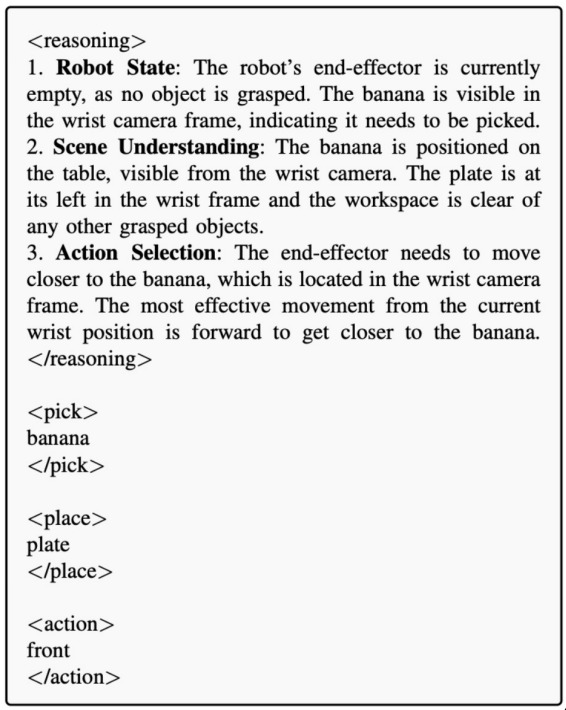
5) ***Workload Survey Details****:* After completing data collection with each method, participants fill out a NASA-TLX questionnaire [3] to assess perceived workload across the four data-collection methods. As shown in Fig. 7, the survey consists of six subscales: mental demand, physical demand, temporal demand, performance, effort, and frustration. Participants rate each subscale by selecting one of 21 evenly spaced tick marks ranging from (-1.0) to $(+1.0)$, where lower score indicates lower perceived workload. To minimize social influence, participants complete the questionnaire individually in separate locations, without discussion or exposure to others' responses. We aggregate the scores across participants and compare workload profiles across methods. Appendix B5 reports the resultant scores and additional analysis.

B. Additional Experiments

We conduct and analyze additional experiments to support and extend the contributions of the main paper. These include behavior cloning performance on novel pick-and-place object



(a) User Input



(b) Reasoning Output

Fig. 1: **Example of Vision-language Reasoning Input and Output in PaPA.** (a) User input consisting of a natural language task description and multi-view camera observations (front and wrist cameras). (b) Corresponding reasoning output generated by the vision-language model, including structured semantic reasoning, selected pick and place targets, and the next end-effector action. This example illustrates how high-level language instructions and visual observations are transformed into explicit semantic guidance for downstream trajectory planning.

combinations, generalization across different imitation learning architectures, additional qualitative trajectory visualizations, and analyses of cognitive effort and workload surveys, as well as ablation studies on the perception module. We further explore extended variants of PaPA, including a sequential and cyclic data collection pipeline and data collection driven by open-vocabulary language instructions, highlighting its potential toward more autonomous and scalable robotic data collection.

*1) **Diverse Pick-and-Place Combinations**:* We evaluate the generalization of PaPA by introducing novel object–container pairings that are not included in the main experiments. Specifically, we design four additional scenarios: Task 5 (banana, pan), Task 6 (sausage, pot), Task 7 (corn, basket), and Task 8 (carrot, plate). The complete results for these configurations are reported in Table I, and the full set of experimental objects is visualized in Fig. 2.

Consistent with the main experimental results, policies trained on PaPA-collected data achieve high success rates that remain competitive with those trained on human teleoperation. Across the additional tasks, the PaPA-trained policy achieves an average success rate of 85%, compared to 92.5% for human teleoperation. Despite this gap, the results indicate that PaPA can robustly collect high-quality demonstration data across diverse object–container combinations with only minor degradation in performance.

TABLE I: **Behavior Cloning Performance on Diverse Pick-and-Place Combinations.** Task success rates for additional object–container pairs not used in the main experiments, evaluating generalization across object identities.

Baseline	**Behavior Cloning Task Success**			
	Task 5	**Task 6**	**Task 7**	**Task 8**
Human Teleoperation	**10 / 10**	8 / 10	**9 / 10**	**10 / 10**
PaPA (ours)	8 / 10	**10 / 10**	8 / 10	8 / 10

TABLE II: **ACT Behavior Cloning Performance.** Task success of ACT policies trained on datasets collected by different methods.

Baseline	**Behavior Cloning Task Success**			
	Task 1	**Task 2**	**Task 3**	**Task 4**
Human Teleoperation	**7 / 10**	5 / 10	6 / 10	**8 / 10**
CLIP-RT	0 / 10	4 / 10	4 / 10	1 / 10
VoxPoser	4 / 10	4 / 10	2 / 10	1 / 10
PaPA (ours)	6 / 10	**6 / 10**	**8 / 10**	6 / 10

*2) **Additional Behavior Cloning Policies**:* We further examine whether the effectiveness of PaPA-collected data generalizes beyond a specific imitation learning methods. To do so, we train policies using ACT [10] on datasets collected by PaPA and baseline methods under identical task settings.

The results in Table II demonstrate that the effectiveness of

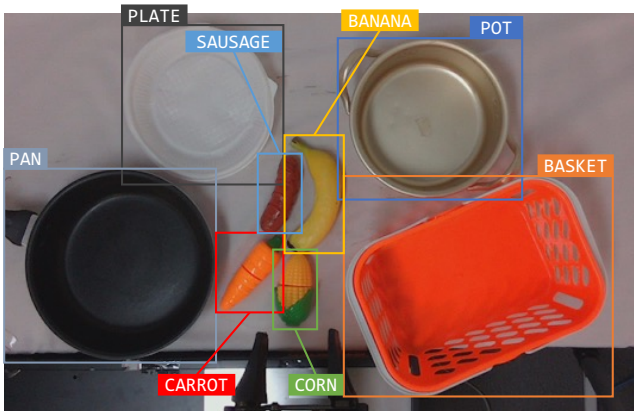


Fig. 2: **Experimental Objects**. The figure illustrates the objects used in our experiments. The set of pick objects includes {banana, sausage, corn, and carrot}, while the set of place targets includes {pot, plate, pan, and basket}.

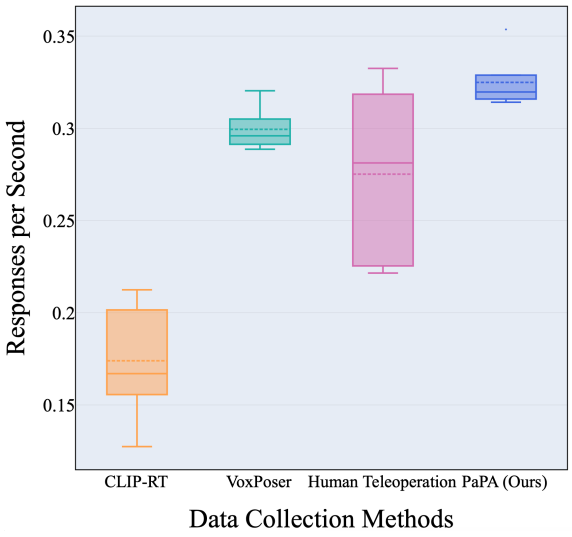


Fig. 3: **PASAT-Based Cognitive Load Analysis.** Box plots show participants' response rates (responses per second) in a PASAT-style dual-task setting during data collection with different methods. Higher values indicate lower cognitive load. PaPA achieves the highest mean response rates, followed by VoxPoser, Human Teleoperation, and CLIP-RT, indicating that PaPA imposes the least cognitive burden during data collection.

PaPA-generated data is not limited to a specific architecture, as ACT policies trained on our data achieve high success rates comparable to those trained on human teleoperation data. Notably, the performance trends observed with ACT closely mirror those of the Diffusion Policy: PaPA and human teleoperation consistently lead the success rates, followed by VoxPose [4] and CLIP-RT [5].

Our analysis indicates that while PaPA consistently provides high-quality data, Diffusion Policy [1] achieves higher absolute success rates than ACT. We hypothesize that this performance gap is attributed to differences in how the two policies represent and handle the 6D action space $(x, y, z, \text{roll}, \text{pitch}, \text{yaw})$. In particular, ACT's temporal ensemble mechanism averages action chunks in Euler-angle representation, which does not preserve the compositional structure of rotations. In contrast, the denoising-based formulation of Diffusion Policy is more robust to such issues, resulting in more kinematically feasible and stable trajectories.

*3) **Additional Trajectory Samples**:* We provide additional rollout visualizations across different scenes and object configurations in Fig. 8 to qualitatively assess trajectory diversity and stability. Compared to baselines, PaPA-generated trajectories exhibit consistent approach behaviors and smooth transitions between key waypoints across environments. These visualizations complement quantitative metrics by illustrating how PaPA maintains human-like motion patterns beyond the examples shown in the main paper.

We also include visualizations from ablation settings. Without global guidance, trajectories become noticeably less diverse and tend to follow uniform, repetitive approach patterns, indicating that global semantic guidance plays a key role in inducing motion diversity. In contrast, removing human supervision preserves the overall trajectory shape, but performance degrades due to unfiltered observation noise rather than changes in the motion itself.

*4) **Analysis on Cognitive Effort**:* The results on cognitive effort measurement are shown in Fig. 3. Participants achieve the highest mean responses rate when collecting data with PaPA (0.32 responses per second), followed by VoxPoser (0.30), Human Teleoperation (0.28), and CLIP-RT, which yields the lowest rate (0.17). These results indicate that PaPA imposes the lowest cognitive burden, whereas CLIP-RT requires the highest cognitive load during data collection. Notably, human teleoperation incurs relatively low cognitive load despite its high physical demands.

We attribute these differences to how much attention participants must allocate to robot operation. With PaPA, the trajectory generation pipeline is fully autonomous, allowing participants to devote minimal attention to the robot while performing the cognitive task. VoxPoser, although also autonomous, occasionally exhibits unstable or abrupt motions that require intermittent monitoring. Human teleoperation, while physically demanding, involves simple and predictable control, enabling participants to maintain focus on the secondary cognitive task. In contrast, CLIP-RT continuously requires experts to observe the environment and issue incremental language instructions, resulting in substantial and sustained cognitive load.

*5) **Analysis on Workload Surveys**:* Table III shows that PaPA consistently achieves the lowest perceived workload across all NASA-TLX subscales, resulting in the lowest average workload score (-0.71). Human teleoperation exhibits low cognitive demand but high physical demand, reflecting continuous manual control. In contrast, CLIP-RT imposes the highest workload across mental demand, effort, and frustration, due to iterative and attention-intensive language-based control. VoxPoser shows intermediate workload levels. Overall, these results demonstrate that PaPA substantially reduces both cognitive and physical workload during data collection compared to existing baselines.

TABLE III: **NASA-TLX Workload Evaluation.** The survey reports average normalized scores ($[-1, 1]$) for six workload subscales: mental demand, physical demand, temporal demand, performance, effort, and frustration. Lower values indicate lower perceived workload, and the rightmost column reports the mean score across all subscales.

Baseline	Mental	Physical	Temporal	Performance	Effort	Frustration	Average
Human Teleoperation	-0.16	0.56	-0.36	-0.72	0.32	-0.42	-0.13
CLIP-RT	0.92	0.58	0.74	0.20	0.90	0.90	0.71
VoxPoser	-0.08	-0.64	-0.28	0.10	-0.44	0.18	-0.19
PaPA (ours)	**-0.62**	**-0.74**	**-0.66**	**-0.84**	**-0.66**	**-0.74**	**-0.71**

TABLE IV: **Ablation Studies on Perception Module Combinations.** Data collection rate (episodes per minute) and data yield for different combinations of vision-language reasoning models and segmentation backends are provided. Results show that PaPA maintains comparable efficiency and yield across perception module choices.

VL Reasoning	Segmentation	Episode per Minute	Data Yield
InternVL3-8B	Grounded-SAM	1.981	90.9
	CLIPSeg	**2.226**	87.0
Qwen3-VL-8B	Grounded-SAM	1.619	**95.2**
	CLIPSeg	2.132	**95.2**
GPT-4	Grounded-SAM	1.492	**95.2**
	CLIPSeg	1.913	**95.2**

*6) **Ablation Studies on Perception Module**:* We conduct ablation studies on the perception stage of PaPA to evaluate its robustness to different combinations of vision-language reasoning models and segmentation backends. Specifically, we evaluate vision-language models including InternVL3-8B [11], Qwen3-VL-8B [9], and GPT-4 [7], paired with segmentation models such as Grounded-SAM [8] and CLIPSeg [6]. We assess each configuration using data collection efficiency metrics, including episode collection rate and the data yield. As shown in Table IV, PaPA achieves comparable data collection efficiency across different perception module combinations, without significant degradation in collection rate or yield. These results demonstrate that PaPA's data collection pipeline is robust to the choice of perception components and does not rely on a specific vision-language or segmentation model.

The results indicate that InternVL3-8B produces faster reasoning outputs than GPT-4 and more consistent formatting than Qwen3-VL-8B in our setting, leading to higher episode collection rates. In particular, we observe that Qwen3-based combinations more frequently produce unformatted outputs, resulting in higher rates of parsing failures and repeated inference calls. Although the InternVL3-8B and Grounded-SAM combination exhibits a slightly lower data yield compared to some alternatives, it provides more stable perception, higher-quality object grounding, and reduced engineering overhead. CLIPSeg additionally requires careful tuning of mask thresholds, introducing further complexity. Based on these trade-offs, we primarily adopt InternVL3-8B and Grounded-SAM in our experiments, while preserving the modularity of the PaPA framework.

TABLE V: **Data Collection Results under Open-Vocabulary Instruction Variations.** We report the number of successful trajectories out of five attempts for each task, using instruction modifications based on color, shape, and semantic properties. Performance remains stable for color- and shape-based variations, while semantic modifications exhibit lower success rates.

Modification	Task 1	Task 2	Task 3	Task 4
Color	**5 / 5**	**4 / 5**	4 / 5	**4 / 5**
Shape	**5 / 5**	3 / 5	**5 / 5**	3 / 5
Semantic	**5 / 5**	1 / 5	**5 / 5**	3 / 5

*7) **Sequential and Cyclic Data Collection**:* We evaluate PaPA in a sequential and cyclic data collection setting, where multiple pick-and-place instructions are executed consecutively within a shared environment. The set of tasks is designed to form a cyclic structure, such that the environment can theoretically return to the initial configuration after completing a sequence of instructions.

In this setup, PaPA receives a sequence of language instructions (e.g., *"pick the banana and place it into the pan"*, followed by *"pick the sausage and place it into the plate"*) and executes them sequentially without resetting the environment between instructions (see Fig. 4). The cyclic task design enables extended data collection with minimal human involvement, requiring intervention only in rare cases to address accumulated minor failures or to restore task feasibility when objects drift outside reachable regions. This setting therefore represents a step toward fully autonomous robotic data collection, while acknowledging the current limitations.

Using this setup, we collect a total of 100 episodes over extended operation. After post-hoc human verification, 70 episodes are identified as successful demonstrations, resulting in approximately 15–20 valid trajectories per instruction with minimal human supervision. These results demonstrate that PaPA also supports efficient sequential data collection under a cyclic task structure, significantly reducing reset overhead while maintaining data quality.

*8) **Open-Vocabulary Language Instruction-based Data Collection**:* We further evaluate PaPA under more diverse natural language instructions by modifying standard pick-and-place commands using generated paraphrases. These instructions vary along three axes—color, shape, and semantic properties (e.g., "banana" $\rightarrow$ {"yellow object", "bent, crescent-

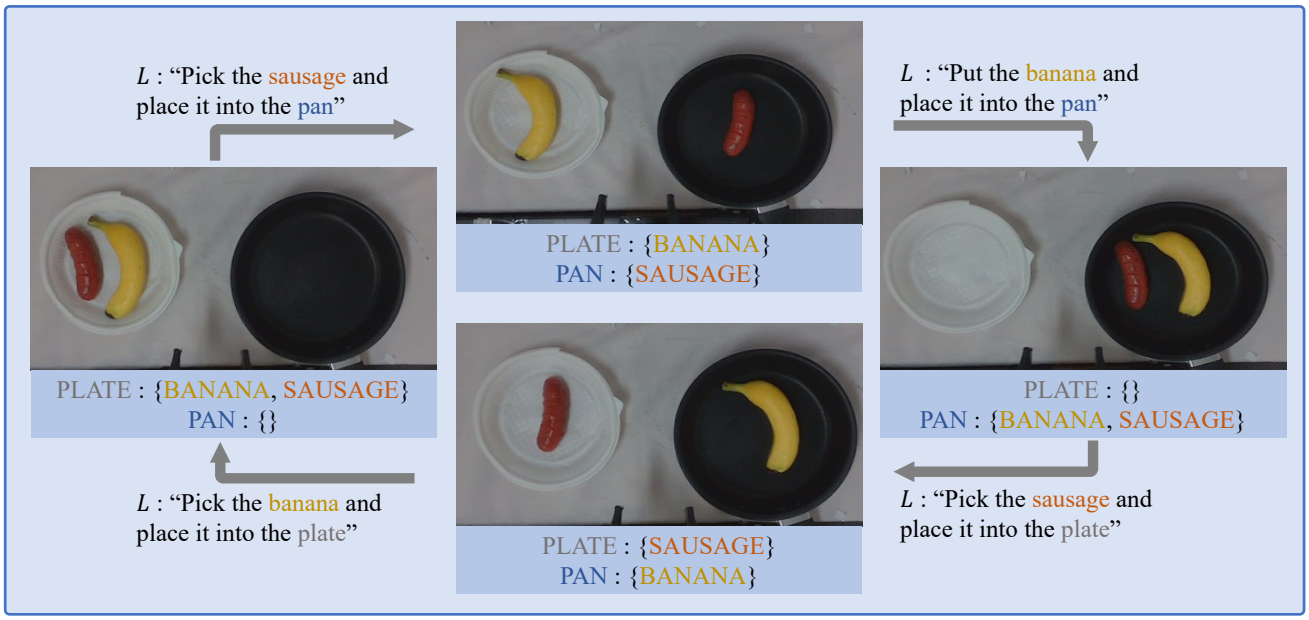


Fig. 4: **Sequential and Cyclic Data Collection with PaPA.** PaPA executes a sequence of language instructions L in a shared environment without intermediate resets. Each image corresponds to the environment state after completing a pick-and-place instruction, with object placements explicitly shown for the plate and pan. Arrows indicate the temporal order of execution, forming a cyclic task structure that allows the scene to return to the initial configuration. This setup enables continuous data collection over extended periods with minimal human supervision.

Modification	Task 1	Task 2	Task 3	Task 4
None	Banana	Sausage	Corn	Carrot
Color	Brightest yellow object in the scene	Brown food in the scene	Yellow object that has some green on it	Vegetable with a green top
Shape	The most bent, crescent-shaped object in the scene	Food with shallow diagonal cuts/slits along its surface	Food item that is more straighter, less curved	Vegetable that is longer and slender
Semantic	Peelable fresh fruit in the scene	Processed meat item in the scene	Food item with a grain/kernel-like texture	Root-vegetable item (the one with a stem end)

Fig. 5: **Open-Vocabulary Instruction Variants.** We show the modified pick-object descriptions for each task under different instruction types. Standard object names are replaced with open-vocabulary expressions based on color, shape, or semantic properties.

shaped object", "peelable fresh fruit"}). The full set of modified instructions is provided in Fig. 5.

We report data collection outcomes in Table V, where the number of successful trajectories out of five attempts is shown for each instruction type. The results indicate that PaPA remains robust to instruction variations based on color and shape, while performance degrades when instructions rely on higher-level semantic properties. This suggests that while PaPA can effectively ground visual attributes, instructions requiring abstract or functional knowledge introduce additional challenges for zero-shot perception.

You are a robotic manipulation planner for PICK-AND-PLACE tasks.
Your goal is to determine:
1) Which object to PICK,
2) Which object or location to PLACE onto,
3) And the best direction for the robot's end-effector to move next.

IMPORTANT DEFINITIONS

- <pick>:
The object that the robot is intended to grasp.
- <place>:
The object or location where the picked object should be placed.
- <action>:
The direction decided in the wrist (end-effector) coordinate frame, indicating how the end-effector should move.

INPUTS

You are given camera images and a task description:

[1] Camera Images:
- Global frontal view for overall scene layout and object arrangement.
- Wrist-mounted close-up view for action selection based on target position relative to the end-effector.

[2] Task Description:
<task>{TASK_DESCRIPTION}</task>

REASONING INSTRUCTIONS

Using the camera images and the task description, reason step-by-step about:

1) ROBOT STATE
- Where is the robot relative to the workspace and objects?
- Is the end-effector currently empty or holding something?
- What object(s) are visible in the cameras, and where are they relative to the wrist?

2) SCENE & OBJECT UNDERSTANDING
- Which objects are visible across different camera views?
- Where are the <pick> and <place> targets located relative to the wrist?

3) ACTION SELECTION (<action>)
- Decide the best direction for the next motion step in the WRIST coordinate frame, based on:
  - the relative position of the target and the wrist
  - spatial context from all camera views
- The action should move the end-effector closer to the current target.
- Allowed directions are: [front, back, left, right, up, down]

OUTPUT FORMAT

First, provide step-by-step reasoning:

<reasoning>
This includes detailed reasoning about:
- robot state
- multi-view scene understanding
- which target to pick and place
- which direction the end-effector should move

</reasoning>

Then output, on separate lines:
1) PICK object:
<pick>
The object to be picked
</pick>

2) PLACE object or location:
<place>
The object or location to place onto
</place>

3) Action direction (WRIST frame):
<action>
CHOSEN_DIRECTION
</action>

STRICT FORMAT RULES

- Always include exactly one <reasoning>... </reasoning> block.
- Always include exactly one <pick>... </pick> tag.
- Always include exactly one <place>... </place> tag.
- Do NOT output multiple actions.
- Do NOT add any commentary or text outside the specified tags.

Fig. 6: **Vision-Language Reasoning Prompt.** The full prompt template provided to the vision-language model is shown, including instructions for defining the task description, visual inputs, structured reasoning instructions, and output format used for downstream trajectory planning.

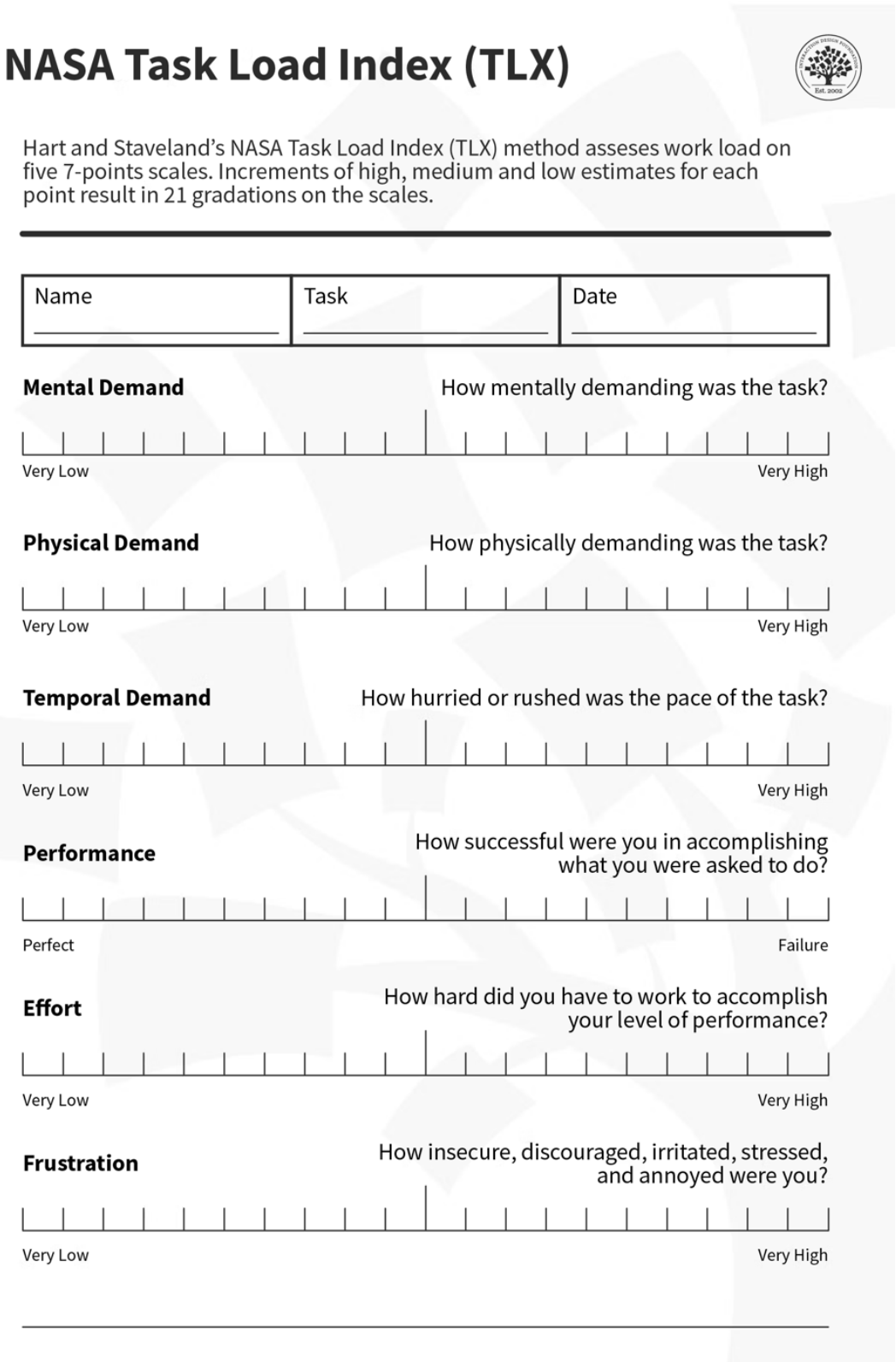
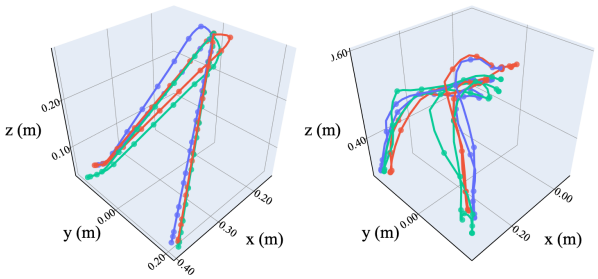


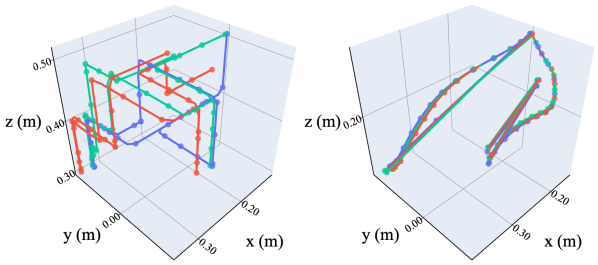
Fig. 7: **NASA-TLX Survey Form.** NASA-TLX survey form is used to collect self-reported workload ratings from operators in our user study.

TABLE VI: **Policy Architectures and Training Hyperparameters.** We report the architectural choices and training hyperparameters for Diffusion Policy and ACT used in all behavior cloning experiments. Unless otherwise noted, these settings are kept consistent across datasets and baselines to ensure fair comparison.

Parameter	**Diffusion Policy**	**ACT**
	Policy Type	
Policy type	Diffusion	ACT (Action Chunking Transformer)
Observation steps	2	1
Action dimension	7	7
Action horizon	16	100
Predicted action steps	8	100
	Visual Encoder	
Vision backbone	ResNet-18	ResNet-18
Pretrained backbone	None	ImageNet-1K
Separate RGB encoders	No	No
Group normalization	Yes	No
	Policy Architecture	
Hidden dimension	–	512
Attention heads	–	8
Encoder layers	–	4
Decoder layers	–	1
Dropout	–	0.1
UNet down dimensions	[512, 1024, 2048]	–
Diffusion embedding dim	128	–
	Diffusion / Generative Settings	
Noise scheduler	DDPM	–
Training timesteps	100	–
Beta schedule	Squared cosine	–
Prediction type	ϵ	–
	VAE / Latent Modeling	
VAE enabled	No	Yes
Latent dimension	–	32
KL weight	–	10.0
	Training Hyperparameters	
Optimizer	Adam	AdamW
Learning rate	1×10^{-4}	1×10^{-5}
Weight decay	1×10^{-6}	1×10^{-4}
Batch size	8	8
Training steps	100k	100k
Learning rate scheduler	Cosine	None

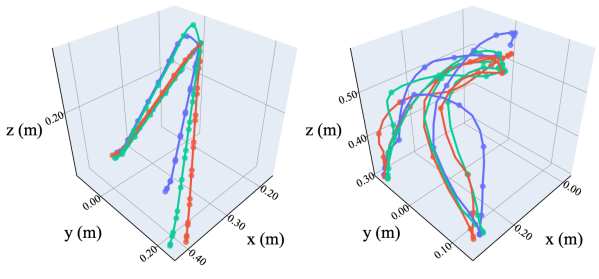


(a) PaPA (Ours) (b) Human Teleoperation

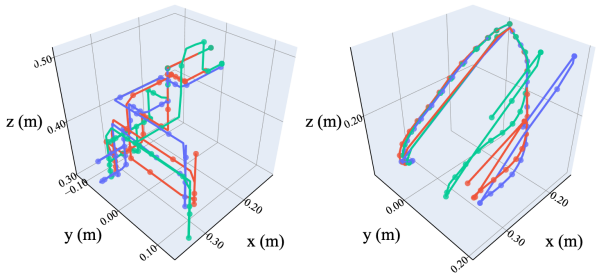


(c) CLIP-RT (d) VoxPoser

(i) Task 2

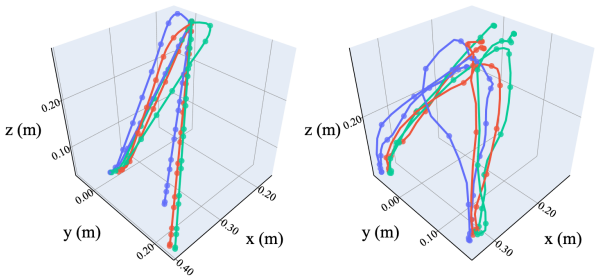


(a) PaPA (Ours) (b) Human Teleoperation

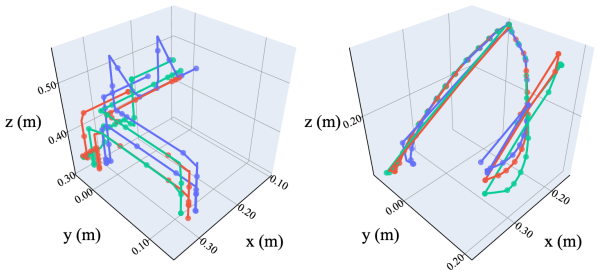


(c) CLIP-RT (d) VoxPoser

(ii) Task 3

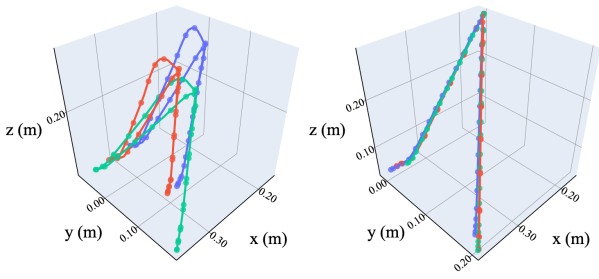


(a) PaPA (Ours) (b) Human Teleoperation

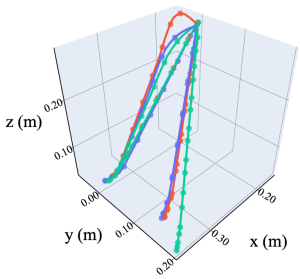


(c) CLIP-RT (d) VoxPoser

(iii) Task 4



(a) PaPA (Ours) (b) w/o global guidance



(c) w/o human supervsion

(iv) Ablation Studies

Fig. 8: **Additional Visualization of Sampled Trajectories from Other Tasks and Ablation Studies.** Additional end-effector trajectories are sampled across multiple tasks and methods, including (i) Task 2, (ii) Task 3, and (iii) Task 4. We also show trajectories from (iv) Ablation Studies.

REFERENCES

[1] Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 2024.

[2] D M Gronwall. Paced auditory serial-addition task: a measure of recovery from concussion. *Percept Mot Skills*, 44(2):367–373, April 1977.

[3] Sandra G. Hart and Lowell E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In Peter A. Hancock and Najmedin Meshkati, editors, *Human Mental Workload*, volume 52 of *Advances in Psychology*, pages 139–183. North-Holland, 1988. doi: https://doi.org/10.1016/S0166-4115(08)62386-9. URL https://www.sciencedirect.com/science/article/pii/S0166411508623869.

[4] Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. *arXiv preprint arXiv:2307.05973*, 2023.

[5] Gi Cheon Kang, Junghyun Kim, Kyuhwan Shim, Jun Ki Lee, and Byoung-Tak Zhang. Clip-rt: Learning language-conditioned robotic policies from natural language supervision. *arXiv preprint arXiv:2411.00508*, 2024.

[6] Timo Lüddecke and Alexander Ecker. Image segmentation using text and image prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7086–7096, June 2022.

[7] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[8] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024.

[9] Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[10] 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.

[11] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.