INSIGHT Bench: Towards Grounded IN-SItu Guidance for Robotic ManipulaTion

Supplementary Material

7. Benchmark details

In this section, we describe the details of INSIGHT Bench including success criteria and demo generation framework. Fig. 12 shows the variations of scene with domain randomization.

7.1. Success criteria

Success for each task is determined by whether a predefined target joint reaches its specified threshold, δ, at any point during an episode. The thresholds are deliberately set to be achievable within the limited workspace of the fixed-base Franka arm.

- **Cabinet:** Success is defined by a significant state change in the target component. For prismatic joints (drawers), the position must exceed $\delta_{\text{slide}} = 0.1\,\text{m}$. For revolute joints (doors), the angle must exceed $\delta_{\text{rotate}} = 0.5\,\text{rad}$.
- **Door:** Success is achieved when the door hinge opens beyond $\delta_{\text{rotate}} = 0.1\,\text{rad}$. While this threshold is small, it is a robust indicator of success because the hinge is mechanically locked until the agent successfully resolves the prerequisite handle-turn constraint.
- **Bottle:** Success is measured by the vertical travel of the cap along its screw axis. Let q_{limit} denote the maximum travel distance. For the opening task, the cap position must exceed $0.6 \times q_{\text{limit}}$. We select this threshold rather than the theoretical maximum to account for the fixed-base robot's limited workspace. Since the primary challenge lies in identifying the correct direction and overcoming the initial lock (in squeeze tasks), 60% travel is sufficient to verify that the correct unfastening action has been sustained. For the closing task, the cap must be fully tightened, defined as a position less than $0.1 \times q_{\text{limit}}$.

7.2. Demo generation framework details

The detailed descriptions of each skill primitive used in demonstration generation are provided below, and Fig. 8 visualizes each primitive.

- **GRASP**(p, q): Moves the end-effector to a target pose defined by position $p \in \mathbb{R}^3$ and orientation quaternion $q \in \mathbb{H}$. This primitive automatically handles the approach phase and gripper actuation.
- **ROTATE**(ϕ): Rotates the end-effector about its local z-axis by ϕ radians. To address joint limits, the primitive employs a recovery strategy: if a limit is reached, the arm executes a 360° unwind motion in the opposite direction to reset the joint configuration before resuming the remaining rotation.

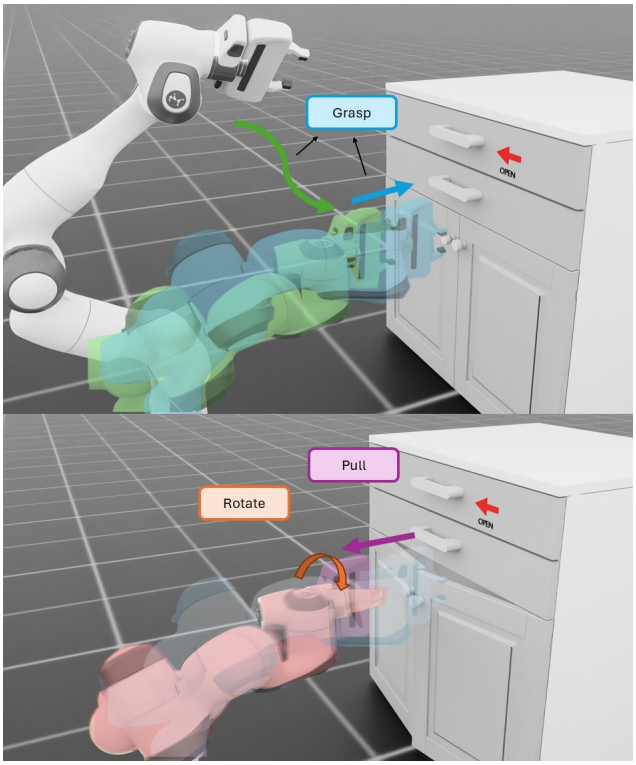
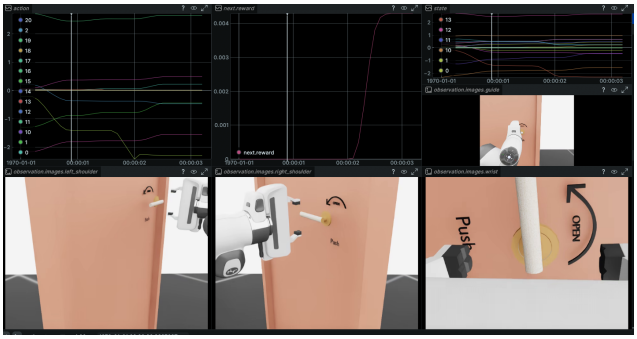


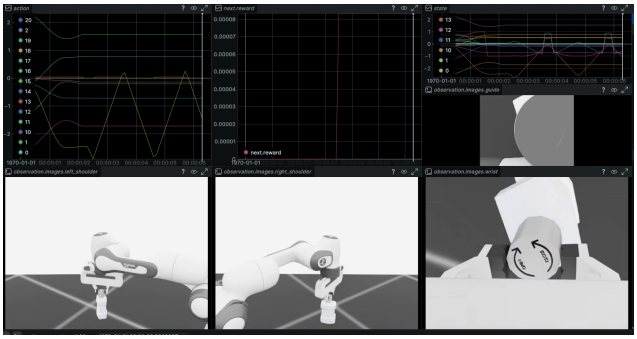
Figure 8. Primitive skills for trajectory generation

- **PULL**(d): Translates the end-effector by distance d along its forward axis. The sign of d determines the direction, enabling both pushing ($d > 0$) and pulling ($d < 0$) interactions within a single primitive.

Our automated generation pipeline enabled the efficient creation of this large-scale dataset. Using a single NVIDIA RTX 4090 GPU, our framework is capable of generating approximately 200 complete episodes per hour. The programmatic expert achieved an average success rate of approximately 52% during generation, with failures primarily occurring when the motion planner could not find a collision-free solution in randomized scenes. Considering the entire process operates without any human intervention, this framework presents a highly efficient and scalable solution for robotics data creation. Fig. 9 shows the example of the dataset.



(a) `Door` task data.



(b) `Bottle` task data.

Figure 9. Visualizations of the dataset in LeRobot format. (a) shows the Door task and (b) shows the Bottle task.

8. Experimental setup details

8.1. Model details

8.1.1. Training details

We assess the performance of three baseline policies: π_0 [4], SmolVLA [34], and GR00T [29]. All policies are trained as multi-task agents, learning the `Cabinet`, `Door`, and `Bottle` tasks simultaneously. Across all baselines, we employed the `AdamW` optimizer with a `cosine decay` learning rate scheduler and trained in `bfloat16` precision with an action chunk size (horizon) of $H = 50$.

Pi0 (π_0). Based on the PaliGemma-3B backbone, Pi0 employs a flow matching objective with a Beta time-sampling distribution. We fine-tuned the model for 50,000 steps using a batch size of 16 and a learning rate of 2.5×10^{-5}. The vision encoder remained frozen during training to preserve pretrained representations.

SmolVLA. Utilizing the SmolVLM2-500M backbone with a specialized Action Expert, we performed full fine-tuning of the model except vision encoder as in π_0. Training was conducted for 30,000 steps with a batch size of 128 and a learning rate of 1.0×10^{-4}.

GR00T N1.5. Groot (GR00T-N1.5) generates actions via a diffusion head. We fine-tuned the projector and diffusion head for 50,000 steps with a batch size of 128 and a learning rate of 1.0×10^{-4}, while keeping the primary vision and language backbones frozen.

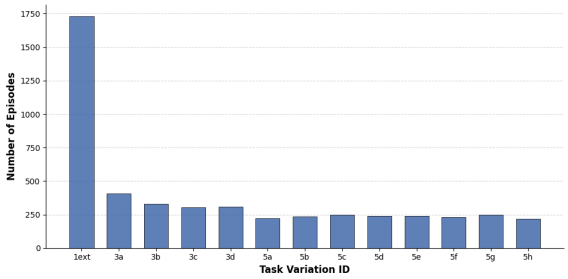


Figure 10. Balanced task distribution for training. To ensure a fair comparison, we aligned the number of episodes between the *Guide* and *Guideless* datasets. For each task variation, the episode count was clipped to the minimum available number across both datasets, discarding excess data.

For the experiments detailed in Sec. 5, we prepared a separate dataset where the positions of object assets were fixed. However, we retained all other domain randomizations, including the position of the guides. We collected 5,647 episodes for the *Guide* dataset and 5,564 episodes for the *Guideless* dataset. To ensure a fair comparison, we balanced the datasets by excluding specific episodes, resulting in an identical number of episodes for each task variation. The detailed task distribution is presented in Fig. 10.

8.1.2. Instructions

In this section, we provide the exact text prompts utilized for each experimental setting: NG-VLA, G-VLA, and LI-VLA in Sec. 5. For the NG-VLA setting, we employ generic prompts that specify the high-level goal without referencing external guides. Conversely, for the G-VLA setting, the phrase "following the guide" is appended to implicitly condition the model to attend to the in-situ visual cues. For the LI-VLA setting, we provide comprehensive text descriptions that explicitly state the physical constraints and action primitives (*e.g.* rotation direction, grasping points) conveyed by the visual guides.

Listing 8.1: Prompts for NG-VLA

```
SCENE_TASK_PROMPT = {
    "cabinet": "Open the cabinet.",
    "CW-Push": "Open the door.",
    "CCW-Push": "Open the door.",
    "CW-Pull": "Open the door.",
    "CCW-Pull": "Open the door.",
    "Open-CW (Sqz)": "Open the bottle.",
    "Open-CCW": "Open the bottle.",
    "Open-CW": "Open the bottle.",
    "Close-CW": "Close the bottle.",
    "Close-CCW": "Close the bottle.",
    "Close-CW": "Close the bottle.",
    "Open-CCW (Sqz)": "Open the bottle.",
    "Close-CCW": "Close the bottle cap.",
}
```

Listing 8.2: Prompts for G-VLA

```
SCENE_TASK_PROMPT_GUIDE = {
    "cabinet": "Open the cabinet following the
        guide.",
    "CW-Push": "Open the door following the guide.",
    # ... (Other keys follow the same pattern)
    "Open-CCW (Sqz)": "Open the bottle following the
        guide.",
    "Close-CCW": "Close the bottle cap following the
        guide.",
}
```

Listing 8.3: Prompts for LI-VLA

```
SCENE_TASK_PROMPT_INSTRUCTION = {
    "cabinet": "Find the arrow guide and open the
        indicated drawer",
    "CW-Push": "Open the door, rotate clockwise and
        push.",
    "CCW-Push": "Open the door, rotate
        counter-clockwise and push.",
    "CW-Pull": "Open the door, rotate clockwise and
        pull.",
    "CCW-Pull": "Open the door, rotate
        counter-clockwise and pull.",
    "Open-CW (Sqz)": "Grip the cap on the sides
        indicated by the 'squeeze' arrow and open
        the bottle in counter-clockwise direction.",
    "Open-CCW": "Open the bottle in
        counter-clockwise direction.",
    "Open-CW": "Open the bottle in clockwise
        direction.",
    # ... (Detailed instructions continued)
    "Open-CCW (Sqz)": "Grip the cap on the sides
        indicated by the 'squeeze' arrow and open
        the bottle in clockwise direction.",
    "Close-CCW": "Close the bottle in
        counter-clockwise direction.",
}
```

8.1.3. Reversed Instructions

To evaluate instruction grounding robustness, we utilize Reversed Instructions, where the directional or procedural constraints contradict the actual physical constraints indicated by the visual guides.

Listing 8.4: Prompts for Reversed Instructions (Adversarial)

```
SCENE_TASK_PROMPT_INSTRUCTION_REVERSE = {
    "cabinet": "Find the arrow guide and open the
        indicated drawer",
    "CW-Push": "Open the door, rotate
        counter-clockwise and pull.",
    "CCW-Push": "Open the door, rotate clockwise and
        pull.",
    # ... (Reversed logic)
    "Open-CW (Sqz)": "Grip the cap on the sides
        indicated by the 'squeeze' arrow and open
        the bottle in clockwise direction.",
}
```

8.1.4. Prompts

As described in Sec. 5.1.1, during the evaluation of the LI-VLA setting, we employ an external VLM to parse visual guides into text instructions when ground truth is unavailable. The following system prompt and few-shot examples are utilized to ensure the generated instructions strictly adhere to the format required by the policy.

Listing 8.5: System Instruction for VLM API

```
prompt_system_instructions = f"""
You are an expert AI assistant specializing in
     robotics and human-robot interaction. Your task
     is to look at an image and generate a clear,
     concise, and actionable instruction for a robot
     (VLA model) to perform a task.

Follow these rules strictly:
1. **Image-Grounded:** The instruction must be based
     ONLY on the visual information in the image. Do
     not invent objects or actions.
2. **Language:** The instruction must be in English.
3. **Format:** The instruction should be a single,
     imperative sentence.
    There are one of 3 objects in the image:
         cabinet, door, bottle.
    For cabinet, use format "Find the arrow guide
         and open the indicated drawer. The indicated
         drawer is ..."
    For bottle without squeeze guide, use format
         "{bottle_action_verb} the bottle in
         (counter-clockwise or clockwise) direction."
    For door, use format "Open the door, rotate
         (counter-clockwise or clockwise) and (push
         or pull)."
    For bottle with squeeze guide, use format "Grip
         the cap on the sides indicated by the
         'squeeze' arrow and
         {bottle_action_verb.lower()} the bottle in
         clockwise direction. The arrows are at
         (number)-degrees counter-clockwise from
         horizontal axis."
4. **Coordinate for grasping:** A horizontal line
     from left to right represents 0 degrees. A
     vertical line from bottom to top represents 90
     degrees. Angles are measured counter-clockwise
     from the 0-degree horizontal axis.
"""
```

9. Additional Analysis and Results

9.1. Effect of instruction on LI-VLA

To validate the reliability of the VLM-based instruction generation module and to verify whether the VLA models effectively ground the semantic content of the provided instructions, we conducted an ablation study under three distinct inference settings:

- LI-VLA-GT (without VLM): Instructions are generated using ground-truth instruction according to the task ID.
- LI-VLA-GPT: Instructions are generated by the external VLM [31] based solely on visual observations of the in-situ guides, as proposed in our main experiment Sec. 5.
- Reverse-VLA: Instructions are intentionally inverted relative to the visual guide (*e.g.* commanding "Counter-Clockwise" when the guide indicates "Clockwise") to act as a negative control for evaluating semantic adherence.

The quantitative results are presented in Fig. 11. We observe that the *Reverse-VLA* setting causes a catastrophic performance drop across all evaluated models compared to the valid instruction settings.

The *Reverse-VLA* setting caused a catastrophic performance drop across all models (*e.g.* π_0 success rate dropped to 3.7%). This confirms that guide information acts as a

Table 2. Detailed success rates (%) for each task variation across all models. For the Cabinet task, we report the aggregated success rate as variations share similar structures. For the Bottle task, variations requiring a squeezing action are marked with (Sqz).

	π_0			**SmolVLA**			**GR00T N1.5**		
Task Variation	NG	G	LI	NG	G	LI	NG	G	LI
Cabinet									
Total	6.0	9.8	9.2	6.5	8.7	2.7	8.7	27.2	24.5
Door									
CW-Push	31.9	8.3	52.8	48.6	26.4	36.1	58.3	44.4	69.4
CCW-Push	18.1	0.0	34.7	6.9	5.6	12.5	13.9	15.3	22.2
CW-Pull	8.3	15.3	34.7	5.6	5.6	22.2	13.9	25.0	30.6
CCW-Pull	5.6	15.3	15.3	0.0	0.0	4.2	1.4	16.7	9.7
Bottle-Standard									
Open-CW	31.2	43.8	36.2	22.5	16.2	20.0	41.2	13.8	40.0
Open-CCW	41.2	53.8	47.5	23.8	13.8	36.2	31.2	23.8	31.2
Close-CW	0.0	6.2	5.0	2.5	3.8	7.5	21.2	10.0	11.2
Close-CCW	10.0	12.5	3.8	11.2	6.2	15.0	7.5	15.0	16.2
Close-CW w/Sqz guide	1.2	5.0	5.0	5.0	3.8	6.2	22.5	15.0	8.8
Close-CCW w/Sqz guide	1.2	0.0	1.2	1.2	2.5	0.0	7.5	22.5	21.2
Bottle-Squeeze									
Open-CW (Sqz)	5.0	11.2	12.5	1.2	2.5	0.0	3.8	2.5	18.8
Open-CCW (Sqz)	7.5	6.2	1.2	2.5	0.0	0.0	5.0	1.2	15.0

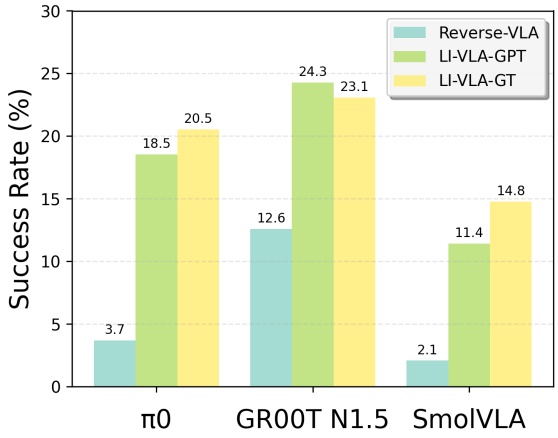


Figure 11. The success rate of total tasks with different instruction setting

hard physical constraint essential for task success; current VLA models cannot visually override incorrect text instructions to solve these constraints. Conversely, the performance gap between *LI-VLA-GPT* and the oracle *LI-VLA-GT* was marginal, with GR00T N1.5 even showing a slight advantage using VLM-generated instructions (24.3% vs. 23.1%).

9.2. Training with noisy language signals in LI-VLA

To address concerns about clean language signals, we train a Noisy-VLA model using VLM-generated instructions with inference-matched noise (Tab. 3). Noisy-VLA achieves a higher success rate (17.5%) than G-VLA (14.1%), indicating that the bottleneck lies in the visual perception and grounding. The VLM-based instruction extractor achieves 73% accuracy overall, with lower performance on tasks requiring global spatial reasoning or fine-grained angle estimation.

Table 3. Ablation Studies. Guide cam ablation, Noisy LI-VLA on π_0 and Imitation Learning baseline.

Task	**Baseline**	**Guide Cam**		**LI-VLA**		**Diffusion Policy**	
	NG-VLA	*w/o*	*G-VLA*	*Noisy*	*Clean*	*Maskout*	*Guide*
Cabinet	6.0	2.7	9.8	4.4	9.2	-	-
Door	16.0	15.6	9.7	27.4	34.4	12.5	17.5
Bottle-Std	14.2	20.9	20.2	21.0	16.5	22.9	50.0
Bottle-Sqz	6.2	6.2	8.8	3.8	6.9	6.25	18.75

9.3. Impact of visual resolution via Guide camera

To separate visual resolution limits from multi-modal fusion failures, we introduce a dedicated guide camera that acts as an embedded symbol-detection module. As shown in Tab. 3, removing this camera creates a clear resolution bottleneck in the Cabinet task, while Door performance reverts to the NG-VLA baseline, indicating that low-resolution guides no longer mislead the model. Bottle tasks remain robust due to close-up wrist views. These results suggest that resolution alone does not explain the failures, and that grounding guide semantics remains the core challenge.

9.4. Necessity of In-Situ guides

We conducted additional experiments using a standard imitation learning baseline, Diffusion Policy [8]. We compare training with access to in-situ visual guides against a guide-masked setting. We perform single-task learning on the Door and Bottle tasks under a constrained evaluation setting. Results are reported in Tab. 3. The performance gap between the two settings indicates that these tasks cannot be reliably solved by affordances alone, and that in-situ guides provide meaningful supervision, supporting the utility of the proposed benchmark.

9.5. Detailed results for each task variation

Tab. 2 presents the success rates for each task variation across all evaluated models and settings. We observe that the impact of in-situ guides and language instructions varies significantly depending on the specific physical constraints of the task variation. For instance, in the Bottle task, variations involving the "squeeze" constraint generally show lower success rates compared to standard rotation tasks, highlighting the difficulty of grounding complex force-interaction constraints. Similarly, `Door` and `Cabinet` tasks exhibit performance fluctuations across variations, indicating that certain geometric configurations or articulation types pose greater challenges for current VLA models.

References

[1] Ayush Agrawal, Joel Loo, Nicky Zimmerman, and David Hsu. Sign language: Towards sign understanding for robot autonomy. *arXiv preprint arXiv:2506.02556*, 2025. 2

[2] Shikhar Bahl, Russell Mendonca, Lili Chen, Unnat Jain, and Deepak Pathak. Affordances from human videos as a versatile representation for robotics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13778–13790, 2023. 2

[3] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, et al. Paligemma: A versatile 3b vlm for transfer. *arXiv preprint arXiv:2407.07726*, 2024. 6

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024. 1, 2, 6

[5] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Joseph Dabis, Chelsea Finn, Keerthana Gopalakrishnan, Karol Hausman, Alex Herzog, Jasmine Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022. 4

[6] Remi Cadene, Simon Alibert, Alexander Soare, Quentin Gallouedec, Adil Zouitine, and Thomas Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. `https://github.com/huggingface/lerobot`, 2024. 6

[7] Guo Chen, Zhiqi Li, Shihao Wang, Jindong Jiang, Yicheng Liu, Lidong Lu, De-An Huang, Wonmin Byeon, Matthieu Le, Tuomas Rintamaki, et al. Eagle 2.5: Boosting long-context post-training for frontier vision-language models. *arXiv preprint arXiv:2504.15271*, 2025. 6, 8

[8] 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. 6, 5

[9] Michael Danielczuk, Andrey Kurenkov, Ashwin Balakrishna, Matthew Matl, David Wang, Roberto Martín-Martín, Animesh Garg, Silvio Savarese, and Ken Goldberg. Mechanical search: Multi-step retrieval of a target object occluded by clutter. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1614–1621. IEEE, 2019. 2

[10] Caelan Reed Garrett, Ajay Mandlekar, Bowen Wen, and Dieter Fox. Skillgen: Automated demonstration generation for efficient skill learning and deployment. In *2nd CoRL Workshop on Learning Effective Abstractions for Planning*, 2024. 4

[11] Haoran Geng, Ziming Li, Yiran Geng, Jiayi Chen, Hao Dong, and He Wang. Partmanip: Learning cross-category generalizable part manipulation policy from point cloud observations. *arXiv preprint arXiv:2303.16958*, 2023. 3, 4

[12] Yiran Geng, Boshi An, Haoran Geng, Yuanpei Chen, Yaodong Yang, and Hao Dong. Rlafford: End-to-end affordance learning for robotic manipulation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5880–5886, 2023. 2

[13] Ran Gong, Jiangyong Huang, Yizhou Zhao, Haoran Geng, Xiaofeng Gao, Qingyang Wu, Wensi Ai, Ziheng Zhou, Demetri Terzopoulos, Song-Chun Zhu, et al. Arnold: A benchmark for language-grounded task learning with continuous states in realistic 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20483–20495, 2023. 2

[14] Jiayuan Gu, Sean Kirmani, Paul Wohlhart, Yao Lu, Montserrat Gonzalez Arenas, Kanishka Rao, Wenhao Yu, Chuyuan Fu, Keerthana Gopalakrishnan, Zhuo Xu, et al. Rt-trajectory: Robotic task generalization via hindsight trajectory sketches. In *The Twelfth International Conference on Learning Representations*. 2

[15] Jiayuan Gu, Fanbo Xiang, Xuanlin Li, Zhan Ling, Xiqiang Liu, Tongzhou Mu, Yihe Tang, Stone Tao, Xinyue Wei, Yunchao Yao, et al. Maniskill2: A unified benchmark for generalizable manipulation skills. *arXiv preprint arXiv:2302.04659*, 2023. 2

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 6

[17] Alexander Khazatsky, Karl Pertsch, Suraj Nair, Ashwin Balakrishna, Sudeep Dasari, Siddharth Karamcheti, Soroush Nasiriany, Mohan Kumar Srirama, Lawrence Yunliang Chen, Kirsty Ellis, et al. Droid: A large-scale in-the-wild robot manipulation dataset. *arXiv preprint arXiv:2403.12945*, 2024. 1

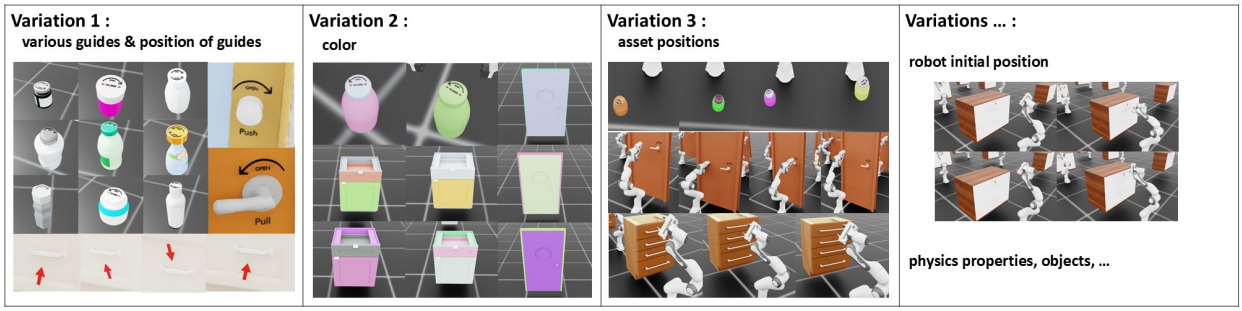


Figure 12. Variations of Scene

[18] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024. 1, 2

[19] Olivia Y Lee, Annie Xie, Kuan Fang, Karl Pertsch, and Chelsea Finn. Affordance-guided reinforcement learning via visual prompting. *arXiv preprint arXiv:2407.10341*, 2024. 2

[20] Yu Li, Xiaojie Zhang, Ruihai Wu, Zilong Zhang, Yiran Geng, Hao Dong, and Zhaofeng He. Unidoormanip: Learning universal door manipulation policy over large-scale and diverse door manipulation environments. *arXiv preprint arXiv:2403.02604*, 2024. 4

[21] Yanbang Li, Ziyang Gong, Haoyang Li, Xiaoqi Huang, Haolan Kang, Guangping Bai, and Xianzheng Ma. Robotic visual instruction. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 12155–12165, 2025. 2

[22] Claire Liang, Cheng Perng Phoo, Laasya Renganathan, Yingying Yu, Bharath Hariharan, and Hadas Kress-Gazit. Perceiving signs for navigation guidance in spaces designed for humans. In *Workshop on Closing the Academia to Real-World Gap in Service Robotics at Robotics Science and Systems (RSS)*, 2020. 2

[23] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023. 6

[24] Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. Libero: Benchmarking knowledge transfer for lifelong robot learning. *arXiv preprint arXiv:2306.03310*, 2023. 2

[25] 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. *arXiv preprint arXiv:2310.17596*, 2023. 4

[26] Andrés Marafioti, Orr Zohar, Miquel Farré, Merve Noyan, Elie Bakouch, Pedro Cuenca, Cyril Zakka, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, et al. Smolvlm: Redefining small and efficient multimodal models. *arXiv preprint arXiv:2504.05299*, 2025. 6

[27] Oier Mees, Lukas Hermann, Erick Rosete-Beas, and Wolfram Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3): 7327–7334, 2022. 2

[28] Mayank Mittal, Calvin Yu, Qinxi Yu, Jingzhou Liu, Nikita Rudin, David Hoeller, Jia Lin Yuan, Ritvik Singh, Yunrong Guo, Hammad Mazhar, Ajay Mandlekar, Buck Babich, Gavriel State, Marco Hutter, and Animesh Garg. Orbit: A unified simulation framework for interactive robot learning environments. *IEEE Robotics and Automation Letters*, 8(6): 3740–3747, 2023. 3

[29] NVIDIA, Nikita Cherniadev Johan Bjorck andFernando Castañeda, Xingye Da, Runyu Ding, Linxi "Jim" Fan, Yu Fang, Dieter Fox, Fengyuan Hu, Spencer Huang, Joel Jang, Zhenyu Jiang, Jan Kautz, Kaushil Kundalia, Lawrence Lao, Zhiqi Li, Zongyu Lin, Kevin Lin, Guilin Liu, Edith Llontop, Loic Magne, Ajay Mandlekar, Avnish Narayan, Soroush Nasiriany, Scott Reed, You Liang Tan, Guanzhi Wang, Zu Wang, Jing Wang, Qi Wang, Jiannan Xiang, Yuqi Xie, Yinzhen Xu, Zhenjia Xu, Seonghyeon Ye, Zhiding Yu, Ao Zhang, Hao Zhang, Yizhou Zhao, Ruijie Zheng, and Yuke Zhu. GR00T N1: An open foundation model for generalist humanoid robots. In *ArXiv Preprint*, 2025. 1, 2, 6

[30] Octo Model Team, Dibya Ghosh, Homer Walke, Karl Pertsch, Kevin Black, Oier Mees, Sudeep Dasari, Joey Hejna, Charles Xu, Jianlan Luo, Tobias Kreiman, You Liang Tan, Lawrence Yunliang Chen, Pannag Sanketi, Quan Vuong, Ted Xiao, Dorsa Sadigh, Chelsea Finn, and Sergey Levine. Octo: An open-source generalist robot policy. In *Proceedings of Robotics: Science and Systems*, Delft, Netherlands, 2024. 1, 2

[31] OpenAI. Introducing gpt-5. https://openai.com/index/introducing-gpt-5/, 2025. 7, 3

[32] Abby O'Neill, Abdul Rehman, Abhiram Maddukuri, Abhishek Gupta, Abhishek Padalkar, Abraham Lee, Acorn Pooley, Agrim Gupta, Ajay Mandlekar, Ajinkya Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903. IEEE, 2024. 1

[33] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of lan-

guage, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR, 2023. 2

[34] Mustafa Shukor, Dana Aubakirova, Francesco Capuano, Pepijn Kooijmans, Steven Palma, Adil Zouitine, Michel Aractingi, Caroline Pascal, Martino Russi, Andres Marafioti, et al. Smolvla: A vision-language-action model for affordable and efficient robotics. *arXiv preprint arXiv:2506.01844*, 2025. 1, 2, 6

[35] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*. 6

[36] Balakumar Sundaralingam, Siva Kumar Sastry Hari, Adam Fishman, Caelan Garrett, Karl Van Wyk, Valts Blukis, Alexander Millane, Helen Oleynikova, Ankur Handa, Fabio Ramos, et al. Curobo: Parallelized collision-free robot motion generation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8112–8119. IEEE, 2023. 6

[37] Priya Sundaresan, Quan Vuong, Jiayuan Gu, Peng Xu, Ted Xiao, Sean Kirmani, Tianhe Yu, Michael Stark, Ajinkya Jain, Karol Hausman, et al. Rt-sketch: Goal-conditioned imitation learning from hand-drawn sketches. In *8th Annual Conference on Robot Learning*. 2

[38] Homer Rich Walke, Kevin Black, Tony Z Zhao, Quan Vuong, Chongyi Zheng, Philippe Hansen-Estruch, Andre Wang He, Vivek Myers, Moo Jin Kim, Max Du, et al. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning*, pages 1723–1736. PMLR, 2023. 1

[39] Yuanfei Wang, Xiaojie Zhang, Ruihai Wu, Yu Li, Yan Shen, Mingdong Wu, Zhaofeng He, Yizhou Wang, and Hao Dong. Adamanip: Adaptive articulated object manipulation environments and policy learning. In *International Conference on Learning Representations*, 2025. 2, 4

[40] Nicky Zimmerman, Louis Wiesmann, Tiziano Guadagnino, Thomas Läbe, Jens Behley, and Cyrill Stachniss. Robust onboard localization in changing environments exploiting text spotting. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 917–924. IEEE, 2022. 2