Local Policies Enable Zero-shot Long-horizon Manipulation

Murtaza Dalal*[1] Min Liu*[1] Walter Talbott[2] Chen Chen[2]
Deepak Pathak[1] Jian Zhang[2] Ruslan Salakhutdinov[1]
Carnegie Mellon University[1], Apple[2]

Abstract: Sim2real for robotic manipulation is difficult due to the challenges of simulating complex contacts and generating realistic task distributions. To tackle the latter problem, we introduce ManipGen, which leverages a new class of policies for sim2real transfer: local policies. Locality enables a variety of appealing properties including invariances to absolute robot and object pose, skill ordering, and global scene configuration. We combine these policies with foundation models for vision, language and motion planning and demonstrate SOTA zero-shot performance of our method to Robosuite benchmark tasks in simulation (97%). We transfer our local policies from simulation to reality and observe they can solve unseen long-horizon manipulation tasks with up to 8 stages with significant pose, object and scene configuration variation. ManipGen outperforms SOTA approaches such as SayCan, OpenVLA and LLMTrajGen across 50 real-world manipulation tasks by 36%, 76% and 62% respectively. All code, models and datasets will be released. Video results at `manipgen.github.io`

Keywords: Sim-to-real Transfer, Long-horizon Manipulation

1 Introduction

How can we develop generalist robot systems that plan, reason, and interact with the world like humans? Tasks that humans solve during their daily lives are incredibly challenging for existing robotics approaches. Cleaning the table, organizing the shelf, putting items away inside drawers, etc. are complex, long-horizon problems that require the robot to act capably and consistently over an extended period of time. Furthermore, such a generalist robot should be able to do so without requiring task-specific engineering effort or demonstrations. Although large-scale data-driven learning has produced generalists for vision and language [1], such models don't yet exist in robotics due to the challenges of scaling data collection. It often takes significant manual labor cost and years of effort to just collect datasets on the order of 100K-1M trajectories [2, 3, 4, 5]. Consequently, generalization is limited, often to within centimeters of an object's pose for complex tasks [6, 7].

Instead, we seek to use a large-scale approach via simulation-to-reality (sim2real) transfer, a cost-effective technique for generating vast datasets that has enabled training generalist policies for locomotion which can traverse complex, unstructured terrain [8, 9, 10, 11, 12, 13]. While sim2real transfer has shown success in industrial manipulation tasks [14, 15, 16], including with high-dimensional hands [17, 18, 19, 20], these efforts often involve training and testing on the same task in simulation. Can we extend sim2real to open-world manipulation, where robots need to solve any task from text instruction? The core bottlenecks are: 1) accurately simulating contact dynamics [21] - for which strategies such as domain randomization [17, 22], SDF contacts [23, 14, 15], and real world corrections [16] have shown promise, 2) generating all possible scene and task configurations to ensure trained policies generalize and 3) acquiring long-horizon behaviors themselves, which may require potentially intractable amounts of data for as the horizon grows.

To address points 2) and 3), our solution is to note that for many manipulation tasks of interest, the skill can be simplified to two steps: achieving a pose near a target object, then performing manipulation. The key idea is that of *locality of interaction*. Policies that observe and act in a region

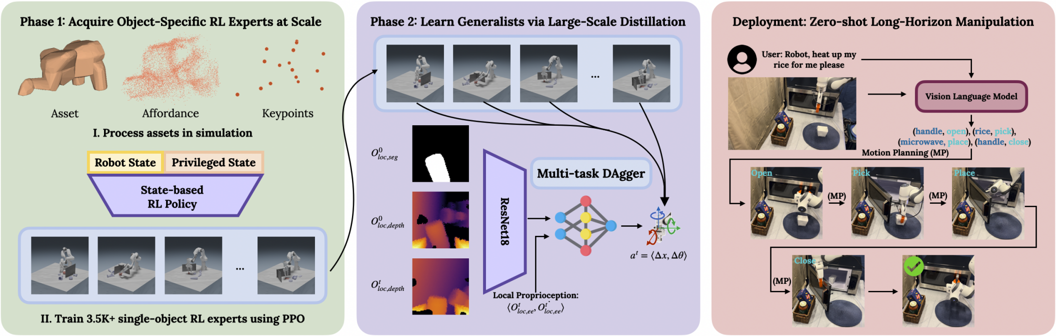
Phase 1: Acquire Object-Specific RL Experts at Scale
Asset
Affordance
Keypoints
I. Process assets in simulation
Robot State
Privileged State
State-based RL Policy
...
II. Train 3.5K+ single-object RL experts using PPO
Phase 2: Learn Generalists via Large-Scale Distillation
...
$O^0_{loc,seg}$
$O^0_{loc,depth}$
$O^t_{loc,depth}$
ResNet18
Multi-task DAgger
$a^t = \langle \Delta x, \Delta\theta \rangle$
Local Proprioception:
$\langle O^t_{loc,ee}, O^t_{loc,ee} \rangle$
Deployment: Zero-shot Long-Horizon Manipulation
User: Robot, heat up my rice for me please
Vision Language Model
(handle, open), (rice, pick), (microwave, place), (handle, close)
Motion Planning (MP)
Open
(MP)
Pick
(MP)
Place
Close
(MP)

Figure 1: **ManipGen Method Overview** (*left*) Train 1000s of RL experts in simulation using PPO (*middle*) Distill single-task RL experts into generalist visuomotor policies via DAgger (*right*) Text-conditioned long-horizon manipulation via task decomposition (VLM), pose estimation and goal reaching (Motion Planning) and sim2real transfer of local policies

local to the target object of interest are by construction: **Absolute pose invariant**: they reason over a smaller set of relative poses between objects and robot. **Skill order invariant**: transition from the termination to initiation of policies via motion planning. **Scene configuration invariant**: they solely observe the local region around the point of interaction.

We propose a novel approach that leverages the strong generalization capabilities of existing foundation models such as Visual Language Models (VLMs) for decomposing tasks into sub-problems [1], processing and understanding scenes [24] and planning collision-avoidant motions [25]. Specifically, given a text prompt, our approach outputs a plan to solve the task (using VLM), estimates where to go and moves the robot accordingly (using motion planning) and deploys local policies for interaction. As a result, a simple scene generation approach can produce strong transfer results across many tasks.

Our contribution is an approach to training agents at scale solely in simulation that are capable of solving a vast set of long-horizon manipulation tasks in the real world *zero-shot*. Our method generalizes to unseen objects, poses, receptacles and skill order configurations. To do so, our method, ManipGen, 1) introduces a novel policy class for sim2real transfer 2) proposes techniques for training policies at scale in simulation 3) and deploys policies via integration with VLMs and motion planners. We perform a thorough, real world evaluation of ManipGen on **50** long-horizon manipulation tasks in **five** environments with up to **8** stages, achieving a success rate of **76%**, outperforming SayCan, OpenVLA and LLMTrajGen by **36%**, **76%** and **62%**.

2 Methods

To build agents capable of generalizing to a wide class of long-horizon robotic manipulation tasks, we propose a novel approach (ManipGen) that hierarchically decomposes manipulation tasks, takes advantage of the generalization capabilities of foundation models for vision and language and uses large-scale learning with our proposed policy class to learn manipulation skills.

Framework We can decompose any task the robot needs to complete into a problem of learning a set of temporally abstracted actions (skills) as well as a policy over those skills [26]. Given a language goal g, and observation O, we can select our policy over skills, $p_\theta(g_k|g, O)$ to be a pre-trained VLM, where g_k is skill k. State-of-the-art VLMs can decompose robotics tasks into language subgoals [27, 28, 29, 30] because they are trained using a vast corpus of internet-scale data and have captured powerful, visually grounded semantic priors for what various real world tasks look like. Any policy class can be used to define the skills, denoted as $p_{\phi_k}(a^t|g_k, O^t)$, which take in the kth sub-goal g_k and current observation O^t. However, note that many manipulation skills (*e.g.* picking, pushing, turning, etc.) can be decomposed into a policy π_{reach} to achieve target poses near objects $X_{targ,k}$ followed by policy π_{loc} for contact-rich interaction. Accordingly, $p_{\phi_k}(a^t|g_k, O^t) = \pi_{reach}(\tau_{reach}|g_k, O^t)\pi_{loc}(a^t_{loc}|O^t_{loc})$. To implement π_{reach}, we need to interpret language sub-goals g_k to take the robot from its current configuration $q_{k,i}$ to some target configuration $q_{k,f}$ such that X_{ee} (the end-effector pose) is close to $X_{targ,k}$. Thus, we structure the VLM's sub-goal predictions, g_k, as tuples containing the following information (object, skill). We then interpret these plans into robot poses by pairing any language conditioned pose estimator or affordance model (to predict $X_{targ,k}$) with an inverse kinematics routine (to compute $q_{k,f}$). Motion planning can predict actions τ_{reach} to achieve the target configuration $q_{k,f}$ while avoiding collisions. Finally, we instantiate local policies (π_{loc}) to be invariant to robot and object poses, order of skill execution and scene configurations with: 1) initialization region s_{init} near a target region/object of interest which has pose $X_{targ,k}$, 2) local observations O^t_{loc}, independent of the absolute configuration of the robot and scene and only observing the environment around the interaction region and 3) actions a^t_{loc}relative to the local observations. Overall: $\pi_{loc}(a^t_{loc}|O^t_{loc})$, $s_{init} = \{s \mid ||X_{ee} - X_{targ,k}||^2 < \epsilon\}$.

Training Local Policies for Sim2Real Manipulation. To train local policies, we adapt the standard two-phase training approach [31, 12, 11, 32, 19, 16] in which we first train state-based expert policies using RL, then distill them into visuomotor policies for transfer. Although local policies can generalize automatically across scene arrangements, robot configurations, and object poses, they must be trained across a wide array of objects to foster object-level generalization. To do so, we train a vast array of *single-object* state-based experts and then distill them into *generalist* visuomotor policies per skill. While such local policies can cover a broad set of manipulation skills (pick and place, articulated/deformable object manipulation, assembly, etc.), in this work, we focus on training the following skills π_{loc}: **pick**, **place**, **grasp handle**, **open** and **close** as a minimal skill library to demonstrate generalist manipulation capabilities for a specific class of tasks. **Pick** grasps any free rigid objects. **Place** sets the object down near the initial pose. **Grasp Handle** grasps the handle of any door or drawer. **Open and Close** pull or push doors and drawers to open or close them. We describe details of the design in data generation, observations, actions, and rewards in **??**.

Generalist Policies via Distillation In order to convert single-object, privileged policies into real world deployable skills, we distill them into multi-object, generalist visuomotor policies using DAgger [33]. For local policies to transfer effectively to the real robot, the observation space and augmentations must be designed with transfer in mind. We use wrist camera depth maps for local observations. Depth maps transfer well from sim2real for locomotion [10, 11, 12, 32], and wrist views are inherently local and improve manipulation performance [34, 35, 36]. To further enforce locality, we clamp depth values and normalize them. Since local wrist-views often get extremely close to the object during execution, it can become difficult for the agent to understand the overall object shape. Thus, we include the initial local observation $O^0_{loc,depth}$ at every step with a segmentation mask of the target object ($O^0_{loc,seg}$) so that the local policy is aware of which object to manipulate. We transform absolute proprioception into local by computing observations relative to the first time-step ($O_{loc,ee} = [X^0_{ee,t} - X^0_{ee}]$) and incorporate velocity information ($O_{lo\dot{c},ee,t}$), which improves transfer. Our observation space is $\mathbf{O^t_{loc}} = \langle O^t_{loc,depth}, O^0_{loc,seg}, O^0_{loc,depth}, O^t_{loc,ee}, O^t_{lo\dot{c},ee} \rangle$. We also apply data augmentation to enable robustness to noisy real world observations, which is detailed in **??**

Zero-shot Text Conditioned Manipulation To enable our system to solve long-horizon tasks, $p_\theta(g_k|g, O)$, ManipGen decomposes the task into a skill chain to execute given goal g. We implement p_θ as GPT-4o. Given the task prompt g, descriptions of the pre-trained local skills and how they operate, and images of the scene O, we prompt GPT-4o to give a plan for the task structured as a list of (object, skill) tuples. We then need a language conditioned pose estimator (to compute $X_{targ,k}$) that generalizes broadly; we opt to use Grounded SAM [24] due to its strong open-set segmentation capabilities. To estimate $X_{targ,k}$, we can segment the object pointcloud, average it to get a position and use its surface normals to select a collision-free orientation. For predicting τ_{reach}, while any motion planner can be used, we select Neural MP [25] due to its fast planning time (3s) and strong real-world planning performance. Given $X_{targ,k}$, we compute target joint state $q_{k,f}$, plan with Neural MP open-loop and execute the predicted τ_{reach} on the robot using a PID joint controller. We then execute the appropriate local policy (as predicted by the VLM) on the robot to perform manipulation. We alternate between motion planning and local policies until the task is complete. Finally, we note that the particular choice of models is orthogonal to our method.

3 Experimental Results

3.1 Simulation Comparisons and Analysis

Robosuite Benchmark Results We first evaluate against the long-horizon manipulation tasks used in PSL [37] from the Robosuite benchmark [38] in simulation. We compare to end-to-end RL methods [39], hierarchical RL [40, 37], TAMP [41] and LLM planning [27]. In these experiments, we *zero-shot* transfer our trained policies to Robosuite and evaluate their performance against methods that use task specific data (Tab. 1). ManipGen outperforms or matches PSL, the SOTA method on these tasks, across the board, achieving an average success rate of 97.33% compared to 95.83%.

	Bread	Can	Milk	Cereal	CanBread	CerealMilk	Average
Stages	*2*	*2*	*2*	*2*	*4*	*4*	
DRQ-v2	52%	32%	2%	0%	0%	0%	14%
RAPS	0%	0%	0%	0%	0%	0%	0%
TAMP	90%	100%	85%	100%	72%	71%	86%
SayCan	93%	100%	90%	63%	63%	73%	80%
PSL	100%	100%	100%	100%	90%	85%	96%
Ours	100%	100%	99%	97%	97%	91%	**97%**

Table 1: **Robosuite Benchmark Results.** ManipGen zero-shot transfers to Robosuite, outperforming end-to-end and hierarchical RL methods as well as traditional and LLM planning methods.

ManipGen Analysis and Ablations. We provide analyses of design decisions in **??**.

3.2 Real World Evaluation

FurnitureBench Results We evaluate the sim2real capabilities of local policies on FurnitureBench [42]. ManipGen achieves an average success of 90%, matching or outperforming end-to-end direct transfer methods (75%, 53.3%), imitation methods (55%, 82.7%, 65%, 75%, 86.7%) and sim2real methods that leverage additional correction data [16]. Detailed analyses are available in **??**.

Tasks	Ours	Transic	Direct Transfer	DR. & Data Aug. [43]	HG-Dagger [44]	IWR [45]	BC [46]
Stabilize	95%	**100%**	10%	35%	65%	65%	40%
Reach and Grasp	**95%**	**95%**	35%	60%	30%	40%	25%
Insert	**80%**	45%	0%	15%	35%	40%	10%
Avg	**90%**	80%	15%	36.7%	43.3%	48.3%	25%

Table 2: **Transic Benchmark Results** ManipGen achieves SOTA results in terms of task success rate without any real world data, outperforming direct transfer, imitation learning and human-in-the-loop methods.

Zero-shot Long-horizon Manipulation To test the generalization capabilities of our method, we propose 5 diverse long-horizon manipulation tasks which involve pick and place, obstacle avoidance and articulated object manipulation. Detailed task descriptions and baselines are provided in **??**.

	Cook	Replace	CabinetStore	DrawerStore	Tidy	Avg
Stages	*2*	*4*	*4*	*6*	*8*	*4.8*
OpenVLA	0% (0.1)	0 (0.0)	0% (0.0)	0 (0.0)	0 (0.0)	0% (.02)
SayCan	80% (1.7)	10% (1.3)	70% (3.5)	20% (3.6)	20% (4.8)	40% (3.0)
LLMTrajGen	70% (1.5)	0% (0.6)	0% (0.6)	0% (1.0)	0% (2.6)	14% (1.3)
Ours	**90% (1.9)**	**80% (3.7)**	**90% (3.9)**	**60% (4.7)**	**60% (7.2)**	**76% (4.3)**

Table 3: **Zero-shot Long Horizon Manipulation** We report task success rate and average number of stages completed per real world task. ManipGen outperforms all methods on each task, achieving 76% with 4.28/4.8 stages completed on average.

Across all 5 tasks (Tab. 3), we find that ManipGen outperforms all methods, achieving 76% **zero-shot success rate** overall. ManipGen is able to avoid obstacles while performing manipulation of unseen objects in arbitrary poses and configurations. Failure cases for our method resulted from 1) vision failures as open-set detection models such as Grounding Dino [47] detected the wrong object, 2) imperfect motion planning, resulting in collisions with the environment during execution which dropped objects sometimes and 3) local policies failing to manipulate from sub-optimal initial poses. In general, DrawerStore and Tidy are the most challenging tasks due to their horizon, and consequently all methods, including our own perform worse (60% for ours, 20% for best baseline).

Acknowledgments

Apple, ONR MURI N00014-22-1-2773 and ONR N00014-22-1-2096.

References

[1] R. OpenAI. Gpt-4 technical report. *arXiv*, pages 2303–08774, 2023.

[2] O.-X. E. Collaboration, A. Padalkar, A. Pooley, A. Jain, A. Bewley, A. Herzog, A. Irpan, A. Khazatsky, A. Rai, A. Singh, et al. Open x-embodiment: Robotic learning datasets and rt-x models. *arXiv preprint arXiv:2310.08864*, 2023.

[3] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, P. D. Fagan, J. Hejna, M. Itkina, M. Lepert, Y. J. Ma, P. T. Miller, J. Wu, S. Belkhale, S. Dass, H. Ha, A. Jain, A. Lee, Y. Lee, M. Memmel, S. Park, I. Radosavovic, K. Wang, A. Zhan, K. Black, C. Chi, K. B. Hatch, S. Lin, J. Lu, J. Mercat, A. Rehman, P. R. Sanketi, A. Sharma, C. Simpson, Q. Vuong, H. R. Walke, B. Wulfe, T. Xiao, J. H. Yang, A. Yavary, T. Z. Zhao, C. Agia, R. Baijal, M. G. Castro, D. Chen, Q. Chen, T. Chung, J. Drake, E. P. Foster, J. Gao, D. A. Herrera, M. Heo, K. Hsu, J. Hu, D. Jackson, C. Le, Y. Li, K. Lin, R. Lin, Z. Ma, A. Maddukuri, S. Mirchandani, D. Morton, T. Nguyen, A. O'Neill, R. Scalise, D. Seale, V. Son, S. Tian, E. Tran, A. E. Wang, Y. Wu, A. Xie, J. Yang, P. Yin, Y. Zhang, O. Bastani, G. Berseth, J. Bohg, K. Goldberg, A. Gupta, A. Gupta, D. Jayaraman, J. J. Lim, J. Malik, R. Martín-Martín, S. Ramamoorthy, D. Sadigh, S. Song, J. Wu, M. C. Yip, Y. Zhu, T. Kollar, S. Levine, and C. Finn. Droid: A large-scale in-the-wild robot manipulation dataset. 2024.

[4] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine. Bridge data: Boosting generalization of robotic skills with cross-domain datasets. *arXiv preprint arXiv:2109.13396*, 2021.

[5] H. Walke, K. Black, A. Lee, M. J. Kim, M. Du, C. Zheng, T. Zhao, P. Hansen-Estruch, Q. Vuong, A. He, V. Myers, K. Fang, C. Finn, and S. Levine. Bridgedata v2: A dataset for robot learning at scale. In *Conference on Robot Learning (CoRL)*, 2023.

[6] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[7] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv: Arxiv-2401.02117*, 2024.

[8] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 5(47):eabc5986, 2020.

[9] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. *arXiv preprint arXiv:2107.04034*, 2021.

[10] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In *Conference on Robot Learning*, pages 403–415. PMLR, 2023.

[11] Z. Zhuang, Z. Fu, J. Wang, C. Atkeson, S. Schwertfeger, C. Finn, and H. Zhao. Robot parkour learning. *arXiv preprint arXiv:2309.05665*, 2023.

[12] X. Cheng, K. Shi, A. Agarwal, and D. Pathak. Extreme parkour with legged robots. *arXiv preprint arXiv:2309.14341*, 2023.

[13] D. Hoeller, N. Rudin, D. Sako, and M. Hutter. Anymal parkour: Learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566, 2024.

[14] B. Tang, M. A. Lin, I. Akinola, A. Handa, G. S. Sukhatme, F. Ramos, D. Fox, and Y. S. Narang. Industreal: Transferring contact-rich assembly tasks from simulation to reality. In K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, editors, *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi:10.15607/RSS.2023.XIX.039. URL `https://doi.org/10.15607/RSS.2023.XIX.039`.

[15] B. Tang, I. Akinola, J. Xu, B. Wen, A. Handa, K. Van Wyk, D. Fox, G. S. Sukhatme, F. Ramos, and Y. Narang. Automate: Specialist and generalist assembly policies over diverse geometries. *arXiv preprint arXiv:2407.08028*, 2024.

[16] Y. Jiang, C. Wang, R. Zhang, J. Wu, and L. Fei-Fei. Transic: Sim-to-real policy transfer by learning from online correction. *arXiv preprint arXiv: Arxiv-2405.10315*, 2024.

[17] I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[18] A. Handa, A. Allshire, V. Makoviychuk, A. Petrenko, R. Singh, J. Liu, D. Makoviichuk, K. Van Wyk, A. Zhurkevich, B. Sundaralingam, et al. Dextreme: Transfer of agile in-hand manipulation from simulation to reality. *arXiv preprint arXiv:2210.13702*, 2022.

[19] T. G. W. Lum, M. Matak, V. Makoviychuk, A. Handa, A. Allshire, T. Hermans, N. D. Ratliff, and K. Van Wyk. Dextrah-g: Pixels-to-action dexterous arm-hand grasping with geometric fabrics. *arXiv preprint arXiv:2407.02274*, 2024.

[20] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal. Visual dexterity: In-hand dexterous manipulation from depth. *arXiv preprint arXiv:2211.11744*, 2022.

[21] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 5026–5033. IEEE, 2012.

[22] O. M. Andrychowicz, B. Baker, M. Chociej, R. Jozefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, et al. Learning dexterous in-hand manipulation. *The International Journal of Robotics Research*, 39(1):3–20, 2020.

[23] Y. S. Narang, K. Storey, I. Akinola, M. Macklin, P. Reist, L. Wawrzyniak, Y. Guo, Á. Moravánszky, G. State, M. Lu, A. Handa, and D. Fox. Factory: Fast contact for robotic assembly. In K. Hauser, D. A. Shell, and S. Huang, editors, *Robotics: Science and Systems XVIII, New York City, NY, USA, June 27 - July 1, 2022*, 2022. doi:10.15607/RSS.2022.XVIII.035. URL `https://doi.org/10.15607/RSS.2022.XVIII.035`.

[24] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks, 2024.

[25] M. Dalal, J. Yang, R. Mendonca, Y. Khaky, R. Salakhutdinov, and D. Pathak. Neural mp: A generalist neural motion planner. *arXiv preprint arXiv:2409.05864*, 2024.

[26] R. S. Sutton, D. Precup, and S. Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999. ISSN 0004-3702. doi:https://doi.org/10.1016/S0004-3702(99)00052-1. URL `https://www.sciencedirect.com/science/article/pii/S0004370299000521`.

[27] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, D. Ho, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, E. Jang, R. J. Ruano, K. Jeffrey, S. Jesmonth, N. J. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, K.-H. Lee, S. Levine, Y. Lu, L. Luu, C. Parada, P. Pastor, J. Quiambao, K. Rao, J. Rettinghouse, D. Reyes, P. Sermanet, N. Sievers, C. Tan, A. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, S. Xu, and M. Yan.

Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv: Arxiv-2204.01691*, 2022.

[28] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. *arXiv preprint arXiv:2207.05608*, 2022.

[29] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *International Conference on Machine Learning*, pages 9118–9147. PMLR, 2022.

[30] J. Wu, R. Antonova, A. Kan, M. Lepert, A. Zeng, S. Song, J. Bohg, S. Rusinkiewicz, and T. Funkhouser. Tidybot: Personalized robot assistance with large language models. *arXiv preprint arXiv:2305.05658*, 2023.

[31] A. Agarwal, S. Uppal, K. Shaw, and D. Pathak. Dexterous functional grasping. In *7th Annual Conference on Robot Learning*, 2023.

[32] S. Uppal, A. Agarwal, H. Xiong, K. Shaw, and D. Pathak. Spin: Simultaneous perception interaction and navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18133–18142, 2024.

[33] S. Ross, G. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635. JMLR Workshop and Conference Proceedings, 2011.

[34] K. Hsu, M. J. Kim, R. Rafailov, J. Wu, and C. Finn. Vision-based manipulators need to also see from their hands. *arXiv preprint arXiv:2203.12677*, 2022.

[35] M. Dalal, A. Mandlekar, C. Garrett, A. Handa, R. Salakhutdinov, and D. Fox. Imitating task and motion planning with visuomotor transformers. 2023.

[36] 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. In *arXiv preprint arXiv:2108.03298*, 2021.

[37] M. Dalal, T. Chiruvolu, D. Chaplot, and R. Salakhutdinov. Plan-seq-learn: Language model guided rl for solving long horizon robotics tasks. In *International Conference on Learning Representations (ICLR)*, 2024.

[38] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, S. Nasiriany, and Y. Zhu. robosuite: A modular simulation framework and benchmark for robot learning. *arXiv preprint arXiv: Arxiv-2009.12293*, 2020.

[39] D. Yarats, R. Fergus, A. Lazaric, and L. Pinto. Mastering visual continuous control: Improved data-augmented reinforcement learning. *arXiv preprint arXiv:2107.09645*, 2021.

[40] M. Dalal, D. Pathak, and R. R. Salakhutdinov. Accelerating robotic reinforcement learning via parameterized action primitives. *Advances in Neural Information Processing Systems*, 34: 21847–21859, 2021.

[41] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 440–448, 2020.

[42] M. Heo, Y. Lee, D. Lee, and J. J. Lim. Furniturebench: Reproducible real-world benchmark for long-horizon complex manipulation. In K. E. Bekris, K. Hauser, S. L. Herbert, and J. Yu, editors, *Robotics: Science and Systems XIX, Daegu, Republic of Korea, July 10-14, 2023*, 2023. doi:10.15607/RSS.2023.XIX.041. URL `https://doi.org/10.15607/RSS.2023.XIX.041`.

[43] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. *arXiv preprint arXiv: Arxiv-1710.06537*, 2017.

[44] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. *arXiv preprint arXiv: Arxiv-1810.02890*, 2018.

[45] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv: Arxiv-2012.06733*, 2020.

[46] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 1. Morgan-Kaufmann, 1988. URL `https://proceedings.neurips.cc/paper_files/paper/1988/file/812b4ba287f5ee0bc9d43bbf5bbe87fb-Paper.pdf`.

[47] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, C. Li, J. Yang, H. Su, J. Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.