HANDFUL: Sequential Grasp-Conditioned Dexterous Manipulation with Resource Awareness

Anonymous Authors

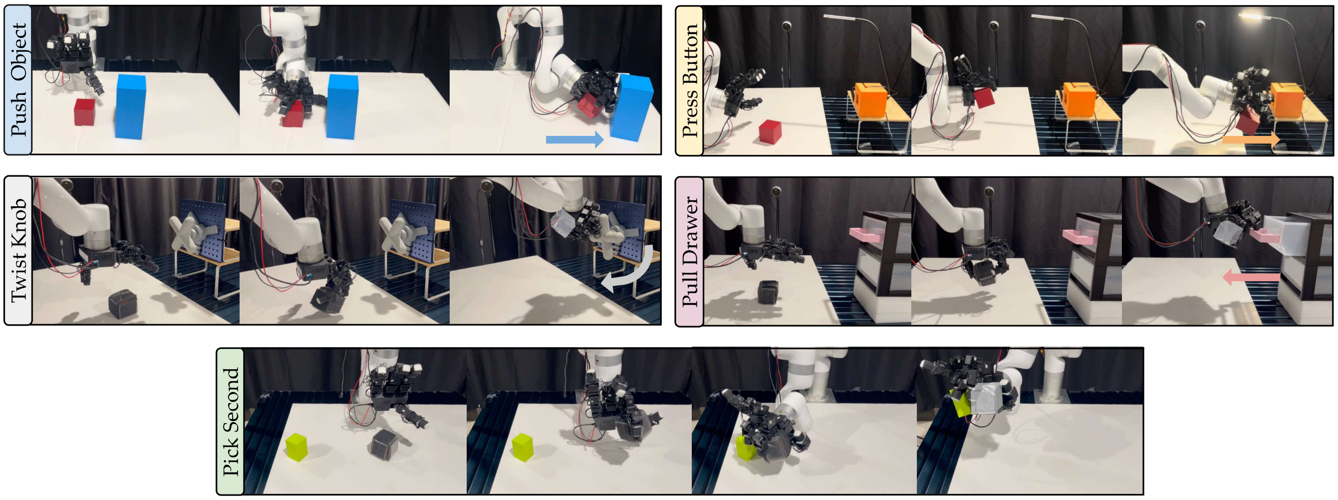

Fig. 1: HANDFUL enables sequential dexterous manipulation by learning resource-aware grasps that preserve specific fingers for downstream subtasks. For each task, the robot first selects an appropriate initial grasp of the object (a block), and then executes a second subtask using the remaining fingers. We show real-world rollouts of HANDFUL using a LEAP Hand [11] for tasks that involve pushing (top left), pressing (top right), twisting (middle left), pulling (middle right), and grasping a second object (bottom).

***Abstract*— Dexterous robot hands offer rich opportunities for multifunctional manipulation, where a robot must execute multiple skills in sequence while maintaining control over previously grasped objects. Most prior work in dexterous manipulation focuses on single-object, single-skill tasks. In contrast, our insight is that many sequential tasks require resource-aware grasps that conserve fingers for future actions. We introduce HANDFUL, a learning framework that models finger usage as a limited resource and encourages exploration of resource-aware grasps through finger-level contact rewards. These grasps are subsequently selected for downstream tasks via curriculum-based policy learning. We further propose HANDFUL-Bench, a simulation benchmark for sequential dexterous manipulation across multiple second-subtask objectives including pushing, pulling, and pressing. Simulation results demonstrate that resource-aware grasps improve second-subtask success over stability-focused baselines. We additionally validate our approach on a real dexterous LEAP Hand. Together, this work establishes resource-aware grasp planning as a key principle for multifunctional dexterous manipulation. Supplementary material is available at: *handful-dex.github.io*.**

I. Introduction

Dexterous multi-fingered hands offer manipulation capabilities that extend beyond parallel-jaw grippers, enabling tasks that require great dexterity such as object reorientation [5]. Despite their versatility, much prior work focuses on isolated skills: grasping [14], [17], pushing [4], or in-hand rotation [6], [7], [15] applied to a single object and optimized for immediate completion.

In contrast, many real-world scenarios are inherently sequential. A robot must often first acquire an object and then perform a second task while continuing to hold it: after grasping a container, pushing a button, pulling a drawer, or reaching for another object. These scenarios require maintaining force closure while simultaneously executing large finger reconfiguration motions with other fingers. Critically, we notice that grasps optimized purely for force closure can occupy fingers needed for subsequent actions, rendering downstream tasks infeasible. These settings therefore require resource-aware grasps that conserve fingers and contact surfaces for future subtasks

We propose HANDFUL (whole-**HAN**d **D**exterity **F**or seq**U**ential task **L**earning), which treats finger usage as a limited resource to be allocated across subtasks. Rather than optimizing grasps for immediate stability, HANDFUL learns resource-aware grasping policies that preserve unused fingers for future actions, then selects the best grasp modalities for each subsequent subtask via curriculum-based policy learning. We also introduce HANDFUL-Bench, a new benchmark built on ManiSkill [12] with the LEAP Hand [11] consisting of tasks that share a common first subtask (grasping), while varying the second subtask objective, such as pushing, pulling, or acquiring a second object, enabling systematic evaluation of multifunctional dexterous policies.

Our main contributions are: (1) a framework for sequential dexterous manipulation that treats finger usage as a limited

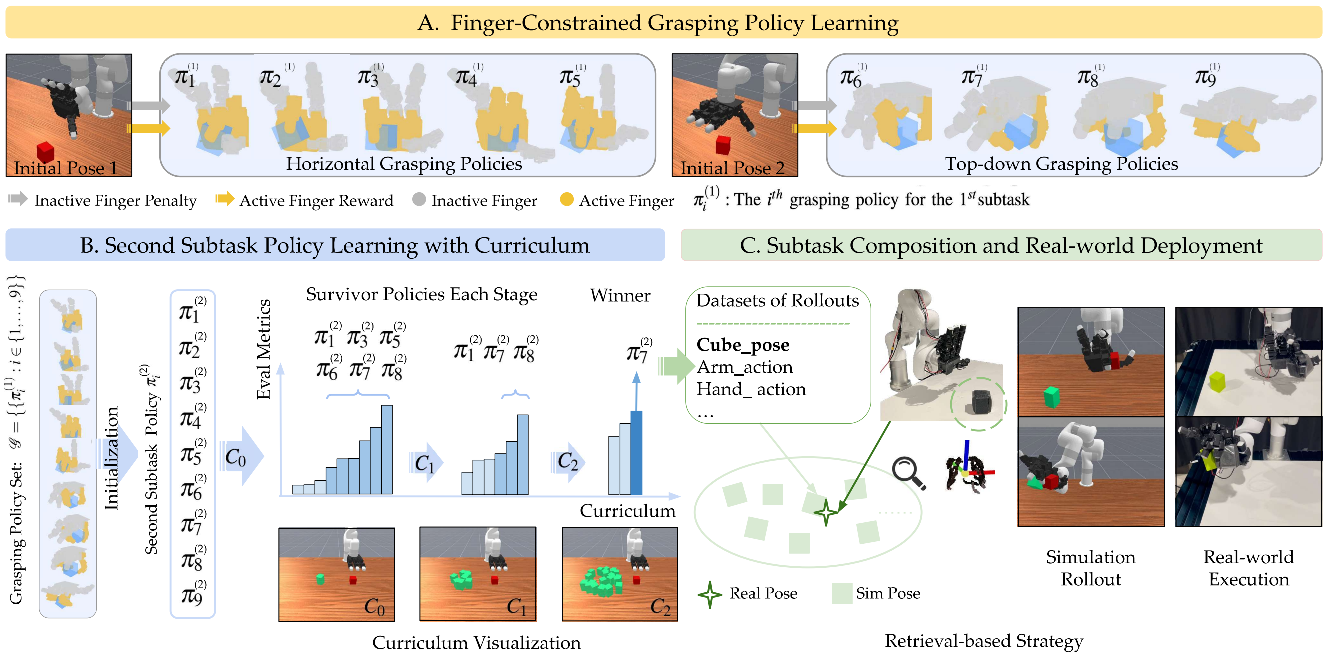

Fig. 2: Overview of HANDFUL. (**A, Sec. II-A**): Multiple grasping policies are trained with active and inactive finger constraints to encourage resource-aware grasps that preserve fingers for future manipulation. **(B, Sec. II-B)**: For each grasp, a second-stage manipulation policy is trained via a multi-stage curriculum, where survivor policies are selected at increasing environment difficulty levels. **(C, Sec. II-C)**: Successful policies for grasping and manipulation are composed sequentially. Real-world object poses are matched to simulated rollouts via retrieval-based execution, enabling sim-to-real transfer.

resource and learns grasp policies that preserve resources for future subtasks; (2) a curriculum-based training approach for efficiently identifying which grasping modalities best support downstream tasks; (3) HANDFUL-Bench, a simulation benchmark for multi-step dexterous manipulation; and (4) experiments demonstrating improved success rates over stability-focused baselines, along with a retrieval-based execution strategy that enables robust real-world deployment.

II. Method: HANDFUL

HANDFUL learns two subpolicies mapping state-based inputs to continuous actions: a grasping policy $\pi^{(1)}$ that produces resource-aware grasps, followed by a second-subtask policy $\pi^{(2)}$ that completes the task while preserving the initial grasp, following the task-decomposition paradigm of Chen et al. [1]. Our method has three main parts: (1) learning a diverse set of dexterous finger-constrained grasping policies (Sec. II-A); (2) learning dexterous finger-constrained second-subtask policies with curriculum learning (Sec. II-B), and (3) deploying composed subtask policies in the real-world via a retrieval-based execution strategy (Sec. II-C).

A. Finger-Constrained Grasping Policy Learning

Because the grasping subtask competes with subsequent subtasks for finger allocation, we define the notion of active fingers, which interact with the current objective, and inactive fingers, which are deliberately conserved for the following subtask. In the first subtask, active fingers grasp the object. In the second subtask, roles reverse: previously active fingers stabilize the grasp while previously inactive fingers execute the new objective. We define the reward r_t during grasping as:

$$r_t = w_g r_g + w_v p_v + w_c p_c + w_a r_a + w_{ina} p_{ina} \tag{1}$$

where we design the following new reward components:

$$r_a = \frac{1}{M}\sum_{i=1}^{M}\exp(-\lambda_a d_i) \qquad \text{Active Finger Reward} \tag{2}$$

$$p_{ina} = \mathrm{clip}\Big(-\sum_{j\in J} f_j, -\beta_{ina}, 0\Big) \quad \text{Inactive Finger Penalty} \tag{3}$$

Here, d_i is the distance from the i-th active contact point (fingertip or palm) to the object, $\lambda_a \in \mathbb{R}$ is a scaling factor, f_j is the contact force between the block and inactive finger $j \in J$, and $\beta_{ina} \in \mathbb{R}$ caps the penalty magnitude. Together, Eq. 2 pulls active contacts (palm and fingertips) toward the object while Eq. 3 discourages inactive fingers from touching it, preserving them for the next subtask. r_g is a standard grasping reward with reaching and lifting components, adopted from [3], [8], [12]; p_v penalizes excessive arm velocity; p_c penalizes table collisions. The $w_g, w_v, w_c, w_a, w_{ina}$ are scalar weights used to form a weighted sum of the reward components.

To provide diverse starting states for second subtasks, we train grasping policies for every one- and two-finger combination on the four-finger hand, from two initial hand poses: palm facing downward (above object) and palm facing horizontally (level with object). This yields nine unique, viable finger configurations (see Fig. 2), forming the grasping policy set $\mathscr{G} = \{\pi_i^{(1)} : i \in \{1,\ldots,9\}\}$. We train the grasping policies using SAC [2], but HANDFUL is compatible with other reinforcement learning algorithms such as PPO [10].

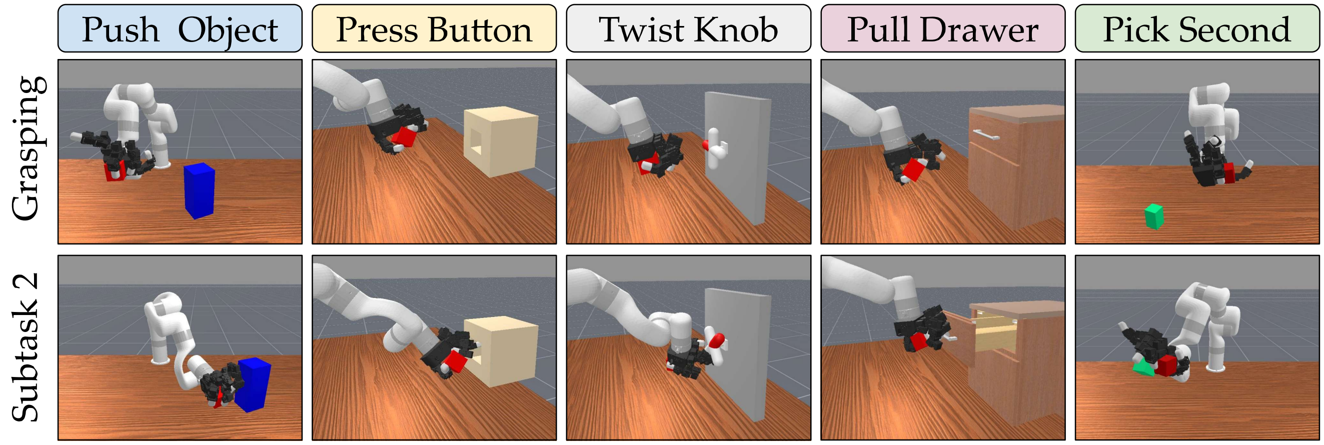

Fig. 3: Overview of HANDFUL-Bench (Sec. III). The top row shows the outcome after grasping the block with resource-aware grasping policies. The bottom row shows successful configurations after performing the second subtask while preserving the initial grasp. The examples above are test-time executions of HANDFUL.

B. Second Subtask Policy Learning with Curriculum

Following the forward initialization scheme of [1], each second-subtask policy $\pi_i^{(2)}$ is initialized from terminal states generated by the corresponding grasping policy $\pi_i^{(1)}$. Different grasps induce distinct hand configurations that directly affect downstream task feasibility, so grasp choice matters significantly. However, training all nine $\pi_i^{(2)}$ to completion is computationally expensive. We therefore introduce a curriculum-based grasp selection strategy that identifies grasps best suited for a specific second subtask and progressively improves policy performance. We construct a curriculum $\{C_0, C_1, C_2\}$ of increasing environment randomization. All nine policies train under C_0; the top six "survivors" (ranked by terminal success rate p_{st}, with any-time success rate p_{sa} and average return p_{ar} as tie-breakers) advance to C_1; the top three advance to C_2. The highest-performing policy in C_2 is the winner. Denoting its index as i^*, we obtain:

$$\pi^{(1)} = \pi_{i^*}^{(1)}, \quad \pi^{(2)} = \pi_{i^*}^{(2)}$$

Second-subtask rewards prioritize the task objective (see Sec. III) while maintaining the initial grasp through finger-role reversal. Fingers active during grasping transition to an inactive, stabilizing role, while previously inactive fingers execute the new objective. We consequently retain the formulations from Eq. 2-3 but apply them to these reversed sets to encourage simultaneous force closure and manipulation. To prevent interference, fingertip-based metrics in these subtasks are computed solely using the newly active set, allowing the inactive, stabilizing fingers to prioritize grasp maintenance.

C. Subtask Composition and Real-World Deployment

For real-world deployment, we adopt a retrieval-based strategy to bridge the sim-to-real gap. To do so, we compose the learned policies in simulation by first executing $\pi^{(1)}$ to obtain a terminal grasp state, followed by $\pi^{(2)}$ to complete the downstream objective while preserving the grasp. Using this strategy, we collect N=50 collision-free simulated trajectories per task under randomized initial object poses, forming dataset $\mathscr{D} = \{\tau_k\}$. During real-world execution, we segment the first object to grasp using SAM2 [9] and estimate its 3D position $\mathbf{p}^{\text{real}} \in \mathbb{R}^3$ from fused point clouds captured by two camera views. We then retrieve the trajectory τ_{k^*} whose initial state best matches the observed pose:

$$k^* = \arg\min_k \|\mathbf{p}_0^{(k)} - \mathbf{p}^{\text{real}}\|_2, \quad k \in \{1, \dots, N\} \tag{4}$$

where $\mathbf{p}_0^{(k)}$ denotes the initial first object position in τ_k. We execute the selected trajectory τ_{k^*} in the real world. This retrieval-based strategy improves robustness compared to directly deploying learned RL policies, reducing the sim-to-real gap by grounding execution in pre-validated trajectories.

III. Benchmark: HANDFUL-Bench

HANDFUL-Bench is an open-source ManiSkill-based framework for sequential dexterous manipulation. Each task shares a common first subtask (grasping an object) but varies the second objective: `Push Object`, `Press Button`, `Twist Knob`, `Pull Drawer`, and `Pick Second` (see Fig. 3). These downstream tasks impose competing demands on finger allocation and in-hand space. Initial grasps optimized solely for force closure can occupy regions of the hand or fingers required for subsequent tasks. As a result, effective manipulation in HANDFUL-Bench requires resource-aware grasps to accommodate diverse downstream objectives.

The robot uses a 23-DOF action space (16 hand, 7 arm) via PD joint delta control. Observations include joint positions/velocities, fingertip/palm poses, and task-specific object or articulation features. To ensure grasp stability, episodes run for their full duration rather than terminating on first success. In turn, we use terminal success rate (p_{st}), success at the final timestep, as our primary metric.

IV. Experiments

A. Simulation Results

We compare HANDFUL against two ablations and one baseline over five seeds (Table I). **w/o Curriculum** skips curriculum selection and trains each second-subtask policy

Method	Push Object	Press Button	Twist Knob	Pull Drawer	Pick Second
HANDFUL (Ours)	$\mathbf{69.90 \pm 5.54}$	$\mathbf{77.75 \pm 2.15}$	$\mathbf{61.52 \pm 5.47}$	$\mathbf{78.94 \pm 1.77}$	76.54 ± 3.63
w/o Curriculum	69.47 ± 1.18	76.80 ± 7.32	58.50 ± 9.48	71.38 ± 20.16	$\mathbf{80.42 \pm 6.72}$
w/o Finger Constraint	66.69 ± 5.66	44.26 ± 40.58	49.44 ± 5.73	58.99 ± 17.36	0.00 ± 0.00
Phase-based Reward	32.38 ± 6.36	10.08 ± 19.23	0.00 ± 0.00	0.00 ± 0.00	0.00 ± 0.00

TABLE I: Success rate across tasks in simulation over five seeds. For each task, only the selected grasp is evaluated for both curriculum and non-curriculum training. Here, "w/o Curriculum" and "w/o Finger Constraint" are ablations of our method (HANDFUL) with the identified component removed (see Sec. IV-A for details). For each task, we bold the metric with the highest mean value.

Curriculum	$\pi_1^{(2)}$	$\pi_2^{(2)}$	$\pi_3^{(2)}$	$\pi_4^{(2)}$	$\pi_5^{(2)}$	$\pi_6^{(2)}$	$\pi_7^{(2)}$	$\pi_8^{(2)}$	$\pi_9^{(2)}$
C_0	51.8	0.0	42.8	0.0	0.0	29.4	36.4	18.4	1.6
C_1	65.2	0.0	31.8	0.0	11.8	27.4	41.2	22.5	3.6
C_2	77.8	0.0	41.8	0.1	13.2	39.3	66.0	52.2	33.3

TABLE II: Curriculum Predictivity for Pick Second. For each grasp at each curriculum stage, we report the average maximum training "success at end" rate p_{st} over 3 seeds. Highlighted are the 1st, 2nd, and 3rd best performing grasps in each stage.

Tasks	Push Object	Press Button	Twist Knob	Pull Drawer	Pick Second
Success (Both)	10	8	6	6	4
Fail Subtask 1 Only	4	4	6	6	7
Fail Subtask 2 Only	0	0	0	0	1
Fail Both	1	3	3	3	3

TABLE III: Real-world experiment results. For each task, we report the number of trials (out of 15) categorized by success and failure types. See Sec. IV-B for details.

directly under C_2 from the same terminal grasp states as the HANDFUL winning grasp. **w/o Finger Constraint** designates all fingers as active in both subtasks, removing resource-aware allocation entirely. **Phase-based Reward** trains the full two-stage task in a unified environment using time-indexed rewards, without task decomposition.

Finger constraints are essential. In Table I, policies with finger constraints (HANDFUL and w/o Curriculum) consistently outperform those without, with the gap widening for tasks requiring greater finger availability such as `Press Button`, `Pull Drawer`, and `Pick Second`. Notably, w/o Finger Constraint fails on `Pick Second` (0%): without explicit allocation between subtasks, the policy cannot free sufficient fingers and in-hand space to pick up the second block, highlighting the importance of promoting and maintaining grasp pose diversity in these manipulation tasks.

Task decomposition is critical. Phase-based Reward fails to solve three of five tasks on any seed. These policies struggle with the shifting reward landscape during rollouts, often prioritizing second-subtask objectives without first learning a stable grasp, underscoring the importance of explicit task decomposition for multi-step manipulation.

Curriculum improves efficiency and stability. HANDFUL achieves comparable success rates to w/o Curriculum across all five tasks while reducing total training by 40% (54M vs. 90M steps), by eliminating underperforming grasp candidates early. We also observe that the curriculum may stabilize training with early stages providing a more consistent environment from which to build successful strategies, though this warrants further investigation.

Grasp selection validity. We validate our curriculum selection strategy through a case study on `Pick Second`, which is the strongest test for our criteria: training is volatile due to tight constraints on in-hand space and finger placement, therefore, it is imperative that early curriculum stages reliably reflect final stage performance to avoid premature elimination of promisng policies. To assess this, we train all nine candidate grasps through C_0, C_1, and C_2 without selection, recording second-subtask policy performance at each stage. As shown in Table II, the grasps that perform best in C_2 (highlighted) consistently rank among the top performers in C_0 and C_1 as well, confirming that early-stage success rate is a reliable signal for grasp selection.

B. Real-World Results

We deploy HANDFUL on a 7-DOF xArm7 with a LEAP Hand, using SAM2 [9] and two Intel RealSense L515 cameras for object pose estimation to execute retrieved trajectories open-loop. Over 15 trials per task (Table III), HANDFUL achieves full two-stage success rates of 66.7% (`Push Object`), 53.3% (`Press Button`), 40.0% (`Twist Knob`), 40.0% (`Pull Drawer`), and 26.7% (`Pick Second`), despite being trained entirely in simulation. Most failures occur during the second subtask and isolated first-subtask-only failures are rare, indicating that learned grasps are stable, but that sequential multifunctional execution increases task difficulty.

Comparison with DP3. We additionally evaluate all tasks on 3D Diffusion Policy (DP3) [16] trained on 30 DexCap [13] teleoperated demonstrations per task, achieving 38.6% average success over 75 rollouts and underperforming HANDFUL on four of five tasks. Notably, while human demonstrations exhibit a bias toward index-finger use in the second task, HANDFUL more explicitly leverages the palm for grasping and explores hand poses outside of the typical human demonstration distribution.

V. Conclusion

We propose HANDFUL, a framework for sequential dexterous manipulation that treats finger usage as a limited resource, along with HANDFUL-Bench, a simulation benchmark to accelerate future research. Our results suggest that in multi-step, multi-object manipulation tasks, effective subtask policy learning may require explicitly accounting for shared physical resources, such as finger availability and in-hand space, that are easy to overlook when learning skills in isolation. We hope that this perspective inspires future work in multi-step and multi-object dexterous hand manipulation.

REFERENCES

[1] Y. Chen, C. Wang, L. Fei-Fei, and C. K. Liu, "Sequential dexterity: Chaining dexterous policies for long-horizon manipulation," in Conference on Robot Learning (CoRL), 2023.

[2] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," in International Conference on Machine Learning (ICML), 2018.

[3] H. Jiang, Y. Wang, H. Zhou, and D. Seita, "Learning to Singulate Objects in Packed Environments using a Dexterous Hand," in International Symposium on Robotics Research (ISRR), 2024.

[4] Y. Li, Y. Ling, G. S. Sukhatme, and D. Seita, "Learning Geometry-Aware Non-prehensile Pushing and Pulling with Dexterous Hands," in IEEE International Conference on Robotics and Automation (ICRA), 2026.

[5] OpenAI, I. Akkaya, M. Andrychowicz, M. Chociej, M. Litwin, B. McGrew, A. Petron, A. Paino, M. Plappert, G. Powell, R. Ribas, J. Schneider, N. Tezak, J. Tworek, P. Welinder, L. Weng, Q. Yuan, W. Zaremba, and L. Zhang, "Solving rubik's cube with a robot hand," arXiv preprint arXiv:1910.07113, 2019.

[6] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik, "In-Hand Object Rotation via Rapid Motor Adaptation," in Conference on Robot Learning (CoRL), 2022.

[7] H. Qi, B. Yi, S. Suresh, M. Lambeta, Y. Ma, R. Calandra, and J. Malik, "General In-Hand Object Rotation with Vision and Touch," in Conference on Robot Learning (CoRL), 2023.

[8] Y. Qin, B. Huang, Z.-H. Yin, H. Su, and X. Wang, "DexPoint: Generalizable Point Cloud Reinforcement Learning for Sim-to-Real Dexterous Manipulation," in Conference on Robot Learning (CoRL), 2022.

[9] N. Ravi, V. Gabeur, Y.-T. Hu, R. Hu, C. Ryali, T. Ma, H. Khedr, R. Rädle, C. Rolland, L. Gustafson, E. Mintun, J. Pan, K. V. Alwala, N. Carion, C.-Y. Wu, R. Girshick, P. Dollár, and C. Feichtenhofer, "SAM 2: Segment Anything in Images and Videos," arXiv preprint arXiv:2408.00714, 2024.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal Policy Optimization Algorithms," arXiv:1707.06347, 2017.

[11] K. Shaw, A. Agarwal, and D. Pathak, "Leap hand: Low-cost, efficient, and anthropomorphic hand for robot learning," in Robotics: Science and Systems (RSS), 2023.

[12] S. Tao, F. Xiang, A. Shukla, Y. Qin, X. Hinrichsen, X. Yuan, C. Bao, X. Lin, Y. Liu, T. kai Chan, Y. Gao, X. Li, T. Mu, N. Xiao, A. Gurha, Z. Huang, R. Calandra, R. Chen, S. Luo, and H. Su, "ManiSkill3: GPU Parallelized Robotics Simulation and Rendering for Generalizable Embodied AI," arXiv preprint arXiv:2410.00425, 2024.

[13] C. Wang, H. Shi, W. Wang, R. Zhang, L. Fei-Fei, and C. K. Liu, "DexCap: Scalable and Portable Mocap Data Collection System for Dexterous Manipulation," in Robotics: Science and Systems (RSS), 2024.

[14] R. Wang, J. Zhang, J. Chen, Y. Xu, P. Li, T. Liu, and H. Wang, "Dexgraspnet: A large-scale robotic dexterous grasp dataset for general objects based on simulation," in IEEE International Conference on Robotics and Automation (ICRA), 2023.

[15] Z.-H. Yin, B. Huang, Y. Qin, Q. Chen, and X. Wang, "Rotating without Seeing: Towards In-hand Dexterity through Touch," in Robotics: Science and Systems (RSS), 2023.

[16] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3D Diffusion Policy: Generalizable Visuomotor Policy Learning via Simple 3D Representations," in Robotics: Science and Systems (RSS), 2024.

[17] J. Zhang, H. Liu, D. Li, X. Yu, H. Geng, Y. Ding, J. Chen, and H. Wang, "DexGraspNet 2.0: Learning Generative Dexterous Grasping in Large-scale Synthetic Cluttered Scenes," in Conference on Robot Learning (CoRL), 2024.