DexFlyWheel: A Scalable and Self-improving Data Generation Framework for Dexterous Manipulation

Anonymous Author(s)
Affiliation
Address
email

Figure 1: DexFlyWheel generates overall 2000 demonstrations from only one human demonstration per task, including various scenarios. The policies trained on our dataset are successfully deployed in the real-world.

Abstract

Dexterous manipulation is critical to advancing robot capabilities in real-world applications, yet diverse and high-quality datasets remain scarce. Existing data collection methods either rely on human teleoperation or require significant human engineering, or are merely limited to grasping, restricting their scalability and generalization. In this paper, we introduce DexFlyWheel, a scalable data generation framework that employs a self-improving cycle to iteratively expand data diversity. Starting from efficient seed demonstrations warmup, our framework expands data diversity via multiple iterations in the self-improving cycle. Each iteration follows a closed-loop pipeline that combines imitation learning, reinforcement learning, rollout trajectory collection, and data augmentation. At each iteration, we first use imitation learning to extract behavioral priors from demonstrations and employ reinforcement learning to enhance generalization. Based on our policy, we rollout trajectories in simulation and then augment these across different environments and objects positions. As iterations progress, our framework generates more diverse data, including various objects, environments, and object positions. Experimental results show that policies trained on our dataset achieve an average success rate of 81.9% on the challenge test sets, with a real-world transfer success rate of 78.3% on dual-arm lift tasks. Videos can be found on our project website https://DexFlyWheel.github.io.

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

1 Introduction

Learning from Demonstration (LfD) [1] has become increasingly prevalent in robotics. This approach typically requires collecting human demonstration data through teleoperation, where humans directly control robots to perform tasks. In dexterous manipulation particularly [2, 3, 4], the higher degrees of freedom and richer contact interactions demand substantially more data. Recent work has shown that training large models with extensive datasets can achieve more challenging tasks and better generalization [5, 6, 7, 8]. However, collecting sufficiently large and diverse demonstration datasets faces several critical limitations: it requires significant human resources, suffers from weak cross-embodiment transfer (where changes in robot embodiment or sensor placement can invalidate collected data), and typically constrains data collection to laboratory settings. While some researchers have explored using portable motion capture devices for in-the-wild data collection [9], this approach still requires substantial human involvement and remains limited by cross-embodiment gap. Recently, simulation has emerged as a promising solution to address data challenges in robotics [10, 11, 12, 13, 14, 15]. It offers numerous advantages: parallel data collection at scale, easy modification of robot embodiments and sensor configurations, and domain randomization for diverse data generation. Some researchers have employed optimization algorithms or heuristic planning methods [16] to generate demonstrations in simulation. However, these approaches often produce lower-quality trajectories compared to human demonstrations and face the limitations in significant optimization challenges, making them inadequate for dynamic dexterous tasks. While reinforcement learning (RL) has been widely adopted for training policies and collecting data in simulation [14, 17, 18, 19, 20], purely RL-trained models often exhibit non-human-like behaviors, leading to less robust manipulation and increased sim-to-real transfer challenges. Moreover, RL faces exploration difficulties and relies heavily on reward engineering challenges that are particularly acute in dexterous manipulation. In the context of high-quality dexterous manipulation data generation, these limitations make pure RL-based methods less suitable for generating high-quality data.

Given these simulation challenges, researchers have begun exploring the teleoperation with replay-mechanism [10, 21], where humans teleoperate simulated robots to collect training data and then use spatial transformations to synthesize new trajectories. Although this approach captures relatively high-quality data while leveraging simulation-based data augmentation, it typically allows only simple trajectory editing or domain randomization, which constrains the diversity of generated trajectories, especially when handling different objects. These limitations fundamentally restrict the collected data within the scope of collected human demonstrations, motivating us to rethink the role of human demonstrations in data generation pipelines. We observe that manipulating different objects only induces minor changes in the manipulation trajectories. This suggests that human demonstrations could serve as a strong behavioral prior, while RL could refine these priors to handle unseen scenarios through targeted exploration.

Building on this insight, we propose DexFlyWheel, a framework that integrates imitation learning (IL) with residual reinforcement learning (RL). IL captures human-like behaviors from demonstrations, while RL enables generalization by learning adaptive adjustments, enabling the generation of human-like and environmentally robust trajectories. We further enhance data diversity through data augmentation across spatial and environments. By embedding them in a self-improving cycle, we achieve continuous expansion of data diversity and enhancement of policy performance. Our main contributions include:

- We present DexFlyWheel, a novel framework that efficiently generates diverse dexterous manipulation data through iterative self-improving. Starting from a single human demonstrations per task, our framework efficiently generates diversity and high-quality demonstrations while maintaining human-like behavior patterns.
- We demonstrate the effectiveness of our framework across multiple dexterous manipulation tasks. We generate over 2,000+ successful demonstrations from just one human demonstration per task with 500+ diverse scenarios, including 80 objects, 12 environments and 15 spatial configurations. Our approach significantly outperforms baseline data generation methods in both diversity and robustness.
- We validate that policies trained with our generated data achieve over average 81.9% success rate in challenging test sets, significantly outperforming policies trained with data from alternative baselines. This superior performance demonstrates the framework's ability to enhance policy generalization across diverse scenarios.

2 Related Work

2.1 Dexterous Manipulation

Dexterous manipulation with multi-fingered robotic hands remains a significant challenge in robotics [22, 23, 24, 25, 26], largely constrained by high-quality demonstration data scarcity. While the prevailing approach employs reinforcement learning (RL) to develop manipulation skills, this method frequently encounters efficiency limitations and exploration challenges [17, 27, 28]. Researchers have explored human video demonstrations [29, 30, 31, 32, 33, 34, 35, 36], but morphological differences between human and robotic hands create substantial transfer barriers. Human teleoperation has emerged as a promising alternative for collecting expert trajectories for imitation learning (IL) [37, 21, 38, 10, 39, 40], effectively capturing expert actions in native robot morphology. Nevertheless, existing approaches still struggle with data collection efficiency or require extensive reward engineering, emphasizing the need for high-quality dexterous manipulation datasets.

2.2 Robotic Data Generation in Simulation

Current approaches for collecting robotic demonstrations in simulation face significant limitations when applied to dexterous manipulation. Motion planning-based methods, while effective for gripper-based systems [16, 41, 42, 11, 18], struggle with the high-dimensional action space and complex contact dynamics of multi-fingered manipulation. Large language models (LLMs) [15, 43, 44, 45] can effectively generate high-level command, but they demonstrate significant limitations when confronted with high-degree-of-freedom dexterous hands, unable to provide the fine-grained guidance necessary for coordinated finger-level control. Reinforcement learning-based approaches generate data through environmental interaction but are severely hampered by scaling challenges inherent to dexterous manipulation, where the high degrees of freedom in multi-fingered hands create exponentially large action spaces that make effective training extremely difficult [14, 42, 46]. Replay-based methods [21, 10, 47] attempt to edit existing demonstrations to new scenarios but face fundamental scalability constraints, as they merely implement spatial transformations of recorded trajectories without the ability to explore novel manipulation strategies beyond the original demonstrations. These collective limitations underscore the critical need for more efficient and scalable methods to generate diverse, high-quality dexterous manipulation data in simulation. Our proposed DexFlyWheel framework addresses these challenges by integrating the complementary strengths of these approaches within a cyclical policy learning paradigm, providing a more effective solution for dexterous skill acquisition that overcomes the individual shortcomings of existing methods.

3 Task Formulation

To address the challenge of generating high-quality synthetic data for robotic manipulation tasks, we train policy models for each manipulation task in simulation environments and use these policies to collect demostrations. We formulate each manipulation task as a Markov Decision Process (MDP) $\mathcal{M} = (\boldsymbol{S}, \boldsymbol{A}, \pi, \mathcal{T}, R, \gamma, \rho, G)$, where $\boldsymbol{S}$ is the state space, $\boldsymbol{A}$ is the action space, π is the agent's policy, $\mathcal{T}(s_{t+1}|s_t, a_t)$ is the transition distribution, R is the reward function, γ is the discount factor, and ρ is the initial state distribution. The policy π conditions on the current state s_t, and generates robot action distributions a_t to maximize the likelihood between the future object states $(s_{t+1}, s_{t+2}, \ldots, s_{t+T})$.

4 Method

In this section, we present DexFlyWheel, a framework for autonomous data generation that bootstraps dexterous manipulation through a self-improving policy learning cycle. We begin with an overview of the DexFlyWheel architecture (Section 4.1) and then detail the two-stage data generation pipeline (Sections 4.2 and 4.3). Together, these components enable scalable collection of diverse and high-quality demonstrations.

4.1 Overview

DexFlyWheel aims to generate diverse and high-quality demonstrations across various scenario configurations, providing broad coverage of objects, environments, and spatial variations, while only starting with minimal human demonstrations. Specifically, DexFlyWheel consists of two stages: the warm-up stage and the data FlyWheel stage. Figure 2 illustrates the overall architecture of our DexFlyWheel framework.

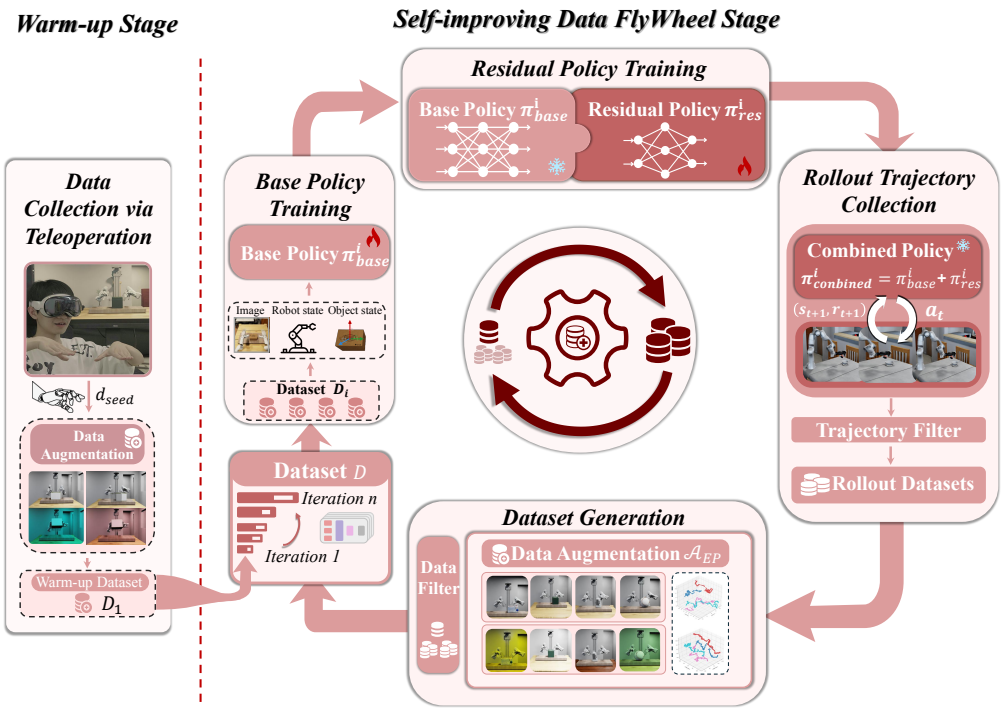

Figure 2: **DexFlyWheel Framework Overview.** Our method consists of two stages: a warm-up stage (left) and a self-improving data flywheel stage (right). In the warm-up stage, seed demonstrations from VR teleoperation are processed through data augmentation to create the initial dataset $\mathcal{D}_1$. The data flywheel stage operates as a closed-loop system with four key components: base policy training via imitation learning, residual policy training using reinforcement learning to enhance object generalization, rollout trajectory collection across various object configurations, and data augmentation to incorporate environment and spatial variations. As the flywheel iterates, both data diversity and policy performance progressively improve, enabling comprehensive generalization across objects, environments, and spatial configurations.

Warm-up stage. A single human demonstration d_{seed} is collected via a VR-based teleoperation system. This seed demonstration is then processed using a multi-dimensional data augmentation module $\mathcal{A}_{\text{EP}}$. Given the human demonstration d_{seed}, the augmentor generates new demonstrations with diverse environment and spatial variations to produce the initial dataset $\mathcal{D}_1$.

Self-improving Data FlyWheel stage. We design a data FlyWheel mechanism to progressively enhance both data diversity and policy performance. This stage comprises multiple iterations $i = \{1, 2, \ldots, n-1\}$, at each iteration i, the following steps are executed: *(1)* An imitation learning policy $\pi^{\text{i}}_{\text{base}}$ is trained on dataset $\mathcal{D}_i$ (with $\mathcal{D}_1$ used when $i = 1$). *(2)* To improve generalization to novel objects, a residual reinforcement learning policy $\pi^{\text{i}}_{\text{res}}$ is trained on top of the frozen $\pi^{\text{i}}_{\text{base}}$, yielding a combined policy $\pi^{\text{i}}_{\text{combined}} = \pi^{\text{i}}_{\text{base}} + \pi^{\text{i}}_{\text{res}}$. *(3)* The combined policy is deployed in simulation to generate demonstrations under various object configurations, forming a high-quality rollout dataset $\mathcal{D}^i_{\text{O}}$. *(4)* Finally, $\mathcal{D}^i_{\text{O}}$ is further augmented by $\mathcal{A}_{\text{EP}}$ with environment and spatial variations to produce the dataset $\mathcal{D}_{i+1}$ used in the next iteration.

4.2 Warm-up Stage

The first stage aims to generate an initial dataset $\mathcal{D}_1$ via data augmentation module $\mathcal{A}_{\text{EP}}$, starting from a single human demonstration d_{seed}. This warm-up stage including two operations:

Data Collection via Teleoperation: To bootstrap the framework with high-quality seed data, we design a VR-based teleoperation system—implemented in simulation using Apple Vision Pro [39]—to accurately track human hand, wrist, and head poses. Since large-scale data collection requires heavily human effort, we only need a single demonstration, denoted as d_{seed}. This demonstration serves as

the sole seed for subsequent data generation. This makes our method highly efficient in terms of human resources while maintaining the quality and diversity of the generated data.

Data Augmentation: To scale and diversify our dataset, we introduce data augmentation module $\mathcal{A}_{\text{EP}}$, which builds upon the MimicGen framework [21] and extends it to support multi-dimensional data augmentation across various environments and spatial configurations. data augmentation module $\mathcal{A}_{\text{EP}}$ takes a base demonstration dataset $\mathcal{D}$ and augmented scenario configurations $\mathcal{C}_{\text{aug}}$ as input, and outputs the augmented dataset $\mathcal{D}'$. In the warmup phase, this process is applied to the seed demonstration: $\mathcal{D}_1 = \mathcal{A}_{\text{EP}}(d_{\text{seed}}; \mathcal{C}_1)$. Through this warmup phase, we establish a foundation of diverse manipulation datasets that serves as the starting point for our iterative data FlyWheel mechanism.

4.3 Self-improving Data FlyWheel Stage

The second stage implements a closed-loop data FlyWheel mechanism that iteratively expands data diversity across objects, environments and spatial generalization dimensions. At each iteration $i \in \{1, 2, ..., n-1\}$, the FlyWheel performs four key operations:

Base Policy Training. Given the dataset $\mathcal{D}_i$ from the previous iteration, we employ diffusion-based policy as the base policy π_{base} to learning dexterous manipulation skill, obtaining a strong base policy for subsequent modules. At each step t, the base policy $\pi^{\text{i}}_{\text{base}}$ takes the state $s_t = \{s_t^{\text{vis}}, s_t^{\text{obj}}, s_t^{\text{prop}}\}$ as inputs, where s_t^{vis} represents visual input from camera, s_t^{obj} contains object state information including 6D pose (position and orientation) and velocities, and s_t^{prop} includes robot proprioception data consisting of joint positions, velocities, and end-effector poses. The policy outputs a sequence of robots actions $(a_t, a_{t+1}, \ldots, a_{t+H})$, where H represents the prediction horizon, and each action a_t consists of the end-effector 6D pose and target joint angles. Implementation details of the policy parameters are provided in Appendix A.1.

Residual Policy Training. Generalizing to novel objects remains a key challenge in imitation-based robotic manipulation, especially when training with limited data. We observe that manipulating different objects induces only small changes in the manipulation trajectories, suggesting that a well-initialized policy can only require fine-grained adjustments to adapt to new objects. Based on this observation, we propose a residual reinforcement learning framework that builds upon the base policy. Specifically, we train a residual policy $\pi^{\text{i}}_{\text{res}}$ that takes object state s_t^{obj} and robot proprioception s_t^{prop} as inputs and generates correction actions $\triangle a = (\triangle a_t, \triangle a_{t+1}, \ldots, \triangle a_{t+H})$. These corrections are applied with a scaling factor α to the trajectory generated by the base policy, forming a combined policy: $\pi^{\text{i}}_{\text{combined}} = \pi^{\text{i}}_{\text{base}} + \alpha \cdot \pi^{\text{i}}_{\text{res}}$. where $\tilde{a}_t = a_t + \alpha \cdot \triangle a_t$ represents the action output from the combined policy $\pi^{\text{i}}_{\text{combined}}$ at each timestep in the horizon. This approach allows the residual policy start from a reasonable robot actions from π_{base} and focus on learning the fine-grained refinements to generalize objects. The reward function R used to train π_{res} is detailed in Appendix A.2. To make the exploration more controllable, we utilize the progressive exploration schedule as proposed by [48]. Specifically, we define the combined policy during training as:

$$\pi_{\text{combined}}(s) = \begin{cases} \pi_{\text{base}}(s) + \alpha \cdot \pi_{\text{res}}(s) & \text{with probability } \epsilon \\ \pi_{\text{base}}(s) & \text{with probability } 1 - \epsilon \end{cases} \tag{1}$$

where ϵ serves as a mixing coefficient that linearly increases from 0 to 1 over T training steps, progressively shifting control from the base policy to the residual policy. This scheduling is designed to stabilize learning by relying on the base policy in early training and gradually incorporating the residual corrections as π_{res} becomes more reliable.

Rollout Trajectory Collection. In robotic manipulation, generating diverse demonstration data across different objects represents the most significant challenge for scaling data collection [10]. Trajectory editing methods cannot adapt to different objects due to their geometry-unaware [10], we use the combined policy π_{combined} to rollout in simulation with objects randomization: $\mathcal{D}^i_{\text{O}} = \{d_j = \{(s_t, a_t)\}_{t=0}^{T-1} | d_j \sim \pi^i_{\text{combined}}\}_{j=1}^{K}$, where we collect K high-quality trajectories by filtering based on task success.

Data Augmentation. In this module, we employ the previously introduced data augmentation module $\mathcal{A}_{\text{EP}}$ to efficiently augment data in various environment and spatial configurations. Taking the dataset $\mathcal{D}^i_{\text{O}}$ and augmented scenario configurations $\mathcal{C}^{i+1}_{\text{aug}}$ as input, the module produces an expanded dataset: $\mathcal{D}_{i+1} = \mathcal{A}(\mathcal{D}^i_{\text{O}}; \mathcal{C}^{i+1}_{\text{aug}})$. The expanded dataset $\mathcal{D}_{i+1}$ is used to train an improved imitation learning policy, which serves as the foundation for the next iteration of the FlyWheel.

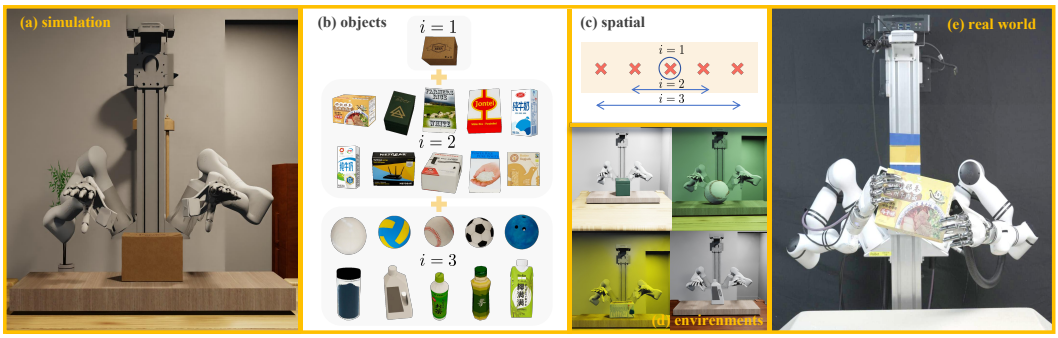

Figure 3: **Experiment Setup.** Taking the dual-arm robot system as an example, **(a)** Our simulation environment. **(b)** Object diversity expansion across iterations, progressing from a single object (i=1) to geometrically similar objects (i=2) and diverse geometries and physical properties objects (i=3). **(c)** Spatial diversity, showing the spatial arrangements. **(d)** Environment diversity, including variations in lighting conditions and tabletop appearances. **(e)** Real-world environment.

5 Experiments

The experiments section are desgin to answer the following questions: **Q1**: How DexFlyWheel effectively expand data diversity and enhance policy generalization? **Q2**: How does DexFlyWheel data generation compare to alternatives? **Q3**:What are the strengths of our method, and how does our system design quantitatively contribute to overall performance? **Q4**: How DexFlyWheel enables real-world deployment on bimanual dexterous robot system?

5.1 Experimental Setup

Tasks and Robots. We evaluate our framework on four dexterous manipulation tasks that assess both fine-grained control in single-arm settings and coordinated motion in dual-arm scenarios: (1) Grasp: The robot must grasp the target object and lift it to a height greater than 0.2 meters from the tabletop. (2) Pour: The robot manipulates a source container to transfer its contents into a target container, requiring controlled pouring of the contained objects from one receptacle to another. (3) Lift: The robot performs collaborative manipulation using its two arms to synchronously lift a object to a minimum height of 15 cm. (4) Handover: The robot performs an intra-agent handover by transferring an object from one hand to the other through a stable, coordinated motion.

For the single-arm Grasp and Pour tasks, we use a Franka Emika Panda robot arm equipped with an Inspire robotic hand. For the dual-arm Lift and Handover tasks, we use a 7-DoF Real-Man RM75-6F arm paired with a 6-DoF PsiBot G0-R hand. The robot is equipped with a RealSense D435 camera mounted on its head, which provides a first-person perspective.

Data Collection and Environment. For each manipulation task, we collected only a single demonstration trajectory using VR teleoperation as our minimal seed data. To ensure the generation of high-quality data, we employed OmniGibson [49] as our simulation platform, leveraging its realistic rendering to generates high-quality data. We prepared 80 distinct objects across various categories and 12 different environments with varying lighting conditions, tabletop appearances. we set the number of iterations to $i = \{1, 2, 3\}$. For each task, we generated 20, 100, and 500 trajectories in the three iterations, respectively. Simuation setup is visualized in Figure 3. More detailed data collection and environment setup are provided in Appendix A.3

Evaluation Design We evaluate our method using two criteria: (1) **data diversity**, the number of object variations O, environment variations E and spatial variations P in our generated dataset $D_{\mathbf{i}}$ in each iteration, and the total number of possible scenario configurations ($O \times E \times P$) these combinations can form from our pipeline; (2) **generalization performance**, the Success Rate (SR) of task execution when policies trained on our generated datasets D_i. It is calculated as the ratio of successful task completion to the total attempts. A better policy will lead to a higher Success Rate. Specifically, we construct two types of test sets. First, the multi-factor generalization test set (T_{OEP}) contains 40 unseen scenario configurations that simultaneously incorporate all three types of

variations: object, environment, and spatial arrangements. Second, the object generalization test set (T_O^i) evaluates the success rate of a robot manipulating different objects when the scenario is fixed. This test set contains all the objects introduced during the data generation process of the i-th iteration. A higher value of this metric not only indicates better object generalization performance of the policy but also implies a better capability to enhance the diversity of objects in data generation. All success rates reported as mean values from 5 independent runs. Detailed compositions of all evaluation sets and success rate calculation method are provided in Appendix A.4.

Baselines. We compared our approach against the following methods: (1) Human Demo (Default): 20 human demonstrations per task in a fixed scenario; (2) Human Demo (Enhanced): 20 demonstrations collected across diverse environments; (3) DexMimicGen(Default) [10]: generates data from a single demonstration via trajectory replay and editing; and (4) DexMimicGen (Enhanced): To create the strongest possible baseline, we provided DexMimicGen with 10 diverse human demonstrations collected across different scenarios—a 10× resource advantage compared to our method's starting point. (4) w/o Res: An ablation model maintaining the base policy and residual policy while removing the $\mathcal{A}_{\text{EP}}$ module. (5) w/o $\mathcal{A}_{\text{EP}}$: An ablation model featuring the base policy with $\mathcal{A}_{\text{EP}}$ but excluding the Residual Reinforcement Learning component. (6) w/o Res. + w/o $\mathcal{A}_{\text{EP}}$: The minimal baseline configuration consisting solely of the base network, without either residual policy or $\mathcal{A}_{\text{EP}}$ components. All methods are evaluated under identical conditions: same source demonstrations, simulation setups, model architectures, and test sets, ensuring a fair and rigorous comparison.

5.2 Data Diversity and Policy Generalization Performance Evaluation

As shown in the mid columns of Table 1, DexFlyWheel successfully expands data diversity with each iteration. In the final iteration (i=3), our method generates an average of 2,250 various scenario configurations spanning 21 different objects per task— *all starting from just a single human demonstration per task.* These results underscore our method's exceptional capability for automated generation of large-scale, diverse, and high-quality demonstration data, significantly reducing the dependency on extensive human data collection. The object diversity results (shown in the O column of Table 1) demonstrate our framework's capacity for object-level data generation. Across all tasks, we expand from a single object per task in iteration 1 to an average of 20 distinct objects per task in iteration 3. This improvement stems from our residual reinforcement learning approach, which enhances adaptation to novel objects during the data generation process. As shown in the SR in T_O^i column, our residual policies consistently improve performance on object generalization by 28.8% on average (comparing base policies to combined policies with residual components). For multi-factor generalization performance (presented in the SR in T_{OEP} column of Table 1), we observe that as dataset diversity increases across iterations, the generalization capabilities of trained policies correspondingly improve, achieving an average success rate of 81.9% in iteration 3—a substantial improvement from the initial 16.5% in iteration 1.

5.3 Performance Comparison Between DexFlyWheel and Baselines

As shown in Table 2, DexFlyWheel significantly outperforms baseline methods across all four tasks. Our findings can be summarized as follows: (a) Our method consistently outperforms all baselines, with an average 28.7% absolute improvement over DexMimicGen (Enhanced), the strongest baseline approach. This substantial performance gap can be attributed to our systematic enhancement of demonstration diversity through the iterative data FlyWheel mechanism. (b) Replay-based data generation methods demonstrate high sensitivity to task complexity, often failing in sophisticated scenarios involving multi-arm coordination such as lift and handover tasks. This limitation likely stems from their offline replay nature, which operates without awareness of changing environmental dynamics and robot behavioral patterns. In contrast, DexFlyWheel maintains consistently high success rates even in these challenging tasks, highlighting the critical importance of its interactive data generation process in enhancing generalization and robustness.(detailed analysis in Section 5.4.) (b) Collecting 20 trajectories in identical scenarios (Human Demo) enables basic skill acquisition but results in poor generalization performance (below 17% on most tasks). While introducing limited diversity can moderately improve performance on simpler tasks, it proves insufficient or even counterproductive for complex manipulation tasks involving dual-arm coordination, such as handover and lift, where Human Demo (Enhanced) achieves merely 2.5% or 0% success rates. This

Table 1: **Self-improving Data Generation Process.** *Left)* Evaluated tasks and iterations. *Middle)* Diversity metrics of our generated datasets. *Right)* Generalization performance of policies trained on our datasets.

Settings		Data Diversity					Generalization Performance	
Task	**Iter.**	O	E	P	**Configs**	**Traj.**	*SR in* T_O^i	*SR in* T_{OEP}
Grasp	$i=1$	1	3	5	15	20	100.0%$_{\pm 1.2\%}$	15.0%$_{\pm 2.1\%}$
	$i=2$	11	8	10	880	100	71.0%$_{\pm 4.3\%}$→ **84.0%**$_{\pm 3.5\%}$	58.0%$_{\pm 4.8\%}$
	$i=3$	**22**	**12**	**15**	**3960**	**500**	35.0%$_{\pm 5.2\%}$→ **89.1%**$_{\pm 3.9\%}$	**90.0%**$_{\pm 3.2\%}$
Pour	$i=1$	1	7	3	21	20	100.0%$_{\pm 3.1\%}$	36.1%$_{\pm 3.3\%}$
	$i=2$	4	9	8	288	100	58.3%$_{\pm 5.1\%}$→ **75.0%**$_{\pm 4.0\%}$	55.6%$_{\pm 4.5\%}$
	$i=3$	**12**	**12**	**10**	**1440**	**500**	58.0%$_{\pm 4.8\%}$→ **80.7%**$_{\pm 3.7\%}$	**85.8%**$_{\pm 3.5\%}$
Lift	$i=1$	1	1	1	1	20	90.0%$_{\pm 2.2\%}$	13.9%$_{\pm 2.8\%}$
	$i=2$	6	5	2	60	100	50.0%$_{\pm 5.3\%}$→ **83.3%**$_{\pm 3.8\%}$	44.4%$_{\pm 4.6\%}$
	$i=3$	**26**	**12**	**5**	**1560**	**500**	68.8%$_{\pm 4.4\%}$→ **98.0%**$_{\pm 2.1\%}$	**79.4%**$_{\pm 7.9\%}$
Handover	$i=1$	1	1	1	1	20	28.8%$_{\pm 4.7\%}$	0.8%$_{\pm 1.1\%}$
	$i=2$	6	5	2	60	100	28.6%$_{\pm 5.8\%}$→ **85.7%**$_{\pm 4.2\%}$	17.5%$_{\pm 3.4\%}$
	$i=3$	**20**	**12**	**5**	**1200**	**500**	32.1%$_{\pm 5.5\%}$→ **62.5%**$_{\pm 4.3\%}$	**72.5%**$_{\pm 4.1\%}$
Avg. $i=1$		1.0	3.0	2.5	9.5	20	79.7%	16.5%
Avg. $i=2$		6.8	6.8	5.5	322.0	100	52.0%→ **82.0%**	43.9%
Avg. $i=3$		**20.0**	**12.0**	**8.8**	**2040.0**	**500**	48.5%→ **82.6%**	**81.9%**
Improvement $(i=1 \rightarrow 3)$		**20.0×**	**4.0×**	**3.5×**	**214.7×**	**25.0×**	—	**+396.4%**

Notes: O: Number of objects, E: Number of environments, P: Number of poses. Configs: Total scenarios configurations ($O \times E \times P$). Rightward arrows (→) show improvement from base policy to combined policy via residual policy.

Table 2: **Comparison with Baselines.** Success rates of policies trained on datasets generated by different methods when tested on multi-factor generalization test set (T_{OEP}).

Method	Grasp	Pour	Lift	Handover	Avg.
Human Demo (Default)	6.1%$_{\pm 1.2\%}$	16.7%$_{\pm 2.5\%}$	13.9%$_{\pm 2.1\%}$	0.8%$_{\pm 1.1\%}$	9.4%
Human Demo (Enhanced)	15.0%$_{\pm 2.1\%}$	36.1%$_{\pm 3.3\%}$	2.5%$_{\pm 1.1\%}$	0%$_{\pm 0.0\%}$	13.4%
DexMimicGen (Default)	30.3%$_{\pm 3.8\%}$	38.9%$_{\pm 4.2\%}$	28.2%$_{\pm 3.5\%}$	28.3%$_{\pm 4.7\%}$	31.4%
DexMimicGen (Enhanced)	50.3%$_{\pm 4.5\%}$	44.4%$_{\pm 3.8\%}$	43.7%$_{\pm 3.6\%}$	42.5%$_{\pm 4.9\%}$	45.2%
Ours	**90.0%**$_{\pm 3.2\%}$	**85.8%**$_{\pm 3.5\%}$	**79.4%**$_{\pm 7.9\%}$	**72.5%**$_{\pm 4.1\%}$	**81.9%**

underscores the necessity of our approach which balances diversity expansion with data quality maintenance.

5.4 Performance Analysis

In this section, we conduct an in-depth analysis to uncover the underlying reasons behind DexFlyWheel's superior performance.

To illustrate our framework's object generalization advantages, we analyze the number of successfully handled objects in the data generation process across different tasks for our method, Ours w/o Res, and DexMimicGen in Figure 4. This figure demonstrates that DexFlyWheel achieves substantially greater object diversity compared to DexMimicGen. This superior performance can be attributed primarily to the residual reinforcement learning module, as evidenced by the significant drop in performance when this component is removed (w/o Res vs. ours: from 8.25 to 20 objects on average). Replay-based methods like DexMimicGen are geometry-unaware, typically operating effectively only on geometrically similar objects (same categories and shapes), which severely limits their ability to handle diverse object sets. To further investigate the robustness of different data generation approaches, we evaluated the data collection success rate across all tasks. In Table 3, we find that DexMimicGen's success rate decreases as task complexity increases, achieving only 14.8% success rate when generating handover demonstrations, whereas our method still maintains 85.7%. The underlying reason is that objects experience varying dynamics during manipulating. Despite same

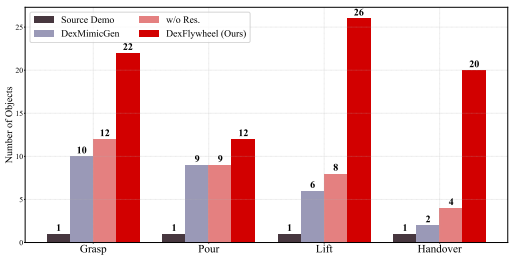

Figure 4: **Comparison of Object Diversity.** Our method successfully handles objects with diverse geometries, sizes, and categories.

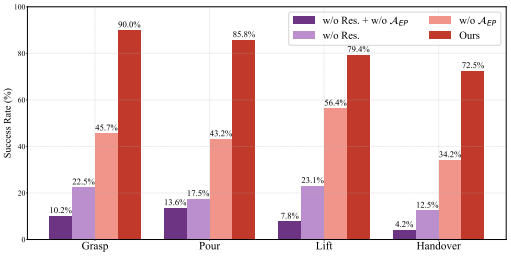

Figure 5: **Ablation Study.** Quantitative contribution of each module in DexFlyWheel across four manipulation tasks.

Table 3: **Data Collection Success Rate Comparison.** Success rates of generating valid demonstrations using different methods across tasks.

Method	Grasp	Pour	Lift	Handover
Human Demo	100.0% ±0.0%	100.0% ±0.0%	100.0% ±0.0%	100.0% ±0.0%
DexMimicGen	87.3% ±3.1%	81.5% ±4.3%	68.2% ±5.6%	**14.8%** ±6.2%
DexFlyWheel (Ours)	**93.6%** ±2.2%	**90.2%** ±2.8%	**89.5%** ±3.5%	**85.7%** ±5.1%

initial states, inherent task dynamics always exist, yet offline replay trajectories fail due to their dynamics-unaware characitcs.

We ablate the core components of DexFlyWheel across four manipulation tasks in Figure 5 We find that the residual policy contributes most significantly to generalization performance.

5.5 Deployment on Real Robots

We transfer our trained policy in simulation into real-world scenarios following a real-to-sim-to-real pipeline. Specifically, we collect one real-world demonstration via teleoperation and replay it in simulation as the initial seed data. Then we generate hundreds of augmented successful demonstrations leveraging DexFlyWheel and distill them into a state-based diffusion policy. The diffusion policy takes the robot proprioception information and the object pose as input and output the robot joint action. Finally, we directly transfer our trained policy back to the real-world scenarios. We set up identical hardware settings both in simulation and real-world as shown in Figure 3, which includes two Realman RM75-6F arms paired with two PsiBot G0-R hands. We employ an egocentric-view RealSense D455 camera to obtain the real-world object pose leveraging FoundationPose [50]. We evaluate the performance of our pipeline on the Dual-arm Lift and Handover Task. The experimental results demonstrate that our simulation-trained policy can be successfully applied to real-world scenarios, achieving average success rates of 78.3% for the Dual-arm Lift task and 63.3% for the Handover task across three trials (20 trajectories per trial).

6 Limitations and Future Work

There are several limitations of our work. First, our reinforcement learning process still requires manually designed rewards, which can be time-consuming and may introduce biases. Future research could explore leveraging large language models to automate reward generation for diverse manipulation tasks. Second, our policies and simulations currently lack tactile feedback due to the immaturity of tactile sensing and simulation technologies. We plan to explore the potential of sensor-based tactile signals for contact-rich tasks in future work.

7 Conclusion

We present DexFlyWheel, a self-improving framework for generating diverse, high-quality dexterous manipulation data from minimal seed demonstrations. Our two-stage pipeline combines imitation learning's behavioral priors with reinforcement learning's generalization capabilities to expand data distribution across diverse environments. Experiments show policies trained on our generated data achieve 75% success rates in challenging scenarios, outperforming baselines, with successful transfer to real-world hardware.

References

[1] Stefan Schaal. Learning from demonstration. Advances in neural information processing systems, 9, 1996.

[2] Yunfei Bai and C. Karen Liu. Dexterous manipulation using both palm and fingers. In International Conference on Robotics and Automation (ICRA), pages 1560–1565, 2014.

[3] S. Gruber. Robot hands and the mechanics of manipulation. IEEE Journal on Robotics and Automation, 2(1):59–59, 1986.

[4] Vikash Kumar, Yuval Tassa, Tom Erez, and Emanuel Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In International Conference on Robotics and Automation (ICRA), pages 6808–6815, 2014.

[5] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, Szymon Jakubczak, Tim Jones, Liyiming Ke, Sergey Levine, Adrian Li-Bell, Mohith Mothukuri, Suraj Nair, Karl Pertsch, Lucy Xiaoyang Shi, James Tanner, Quan Vuong, Anna Walling, Haohuan Wang, and Ury Zhilinsky. π_0: A vision-language-action flow model for general robot control. CoRR, abs/2410.24164, 2024.

[6] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, Tianli Ding, Danny Driess, Avinava Dubey, Chelsea Finn, Pete Florence, Chuyuan Fu, Montse Gonzalez Arenas, Keerthana Gopalakrishnan, Kehang Han, Karol Hausman, Alexander Herzog, Jasmine Hsu, Brian Ichter, Alex Irpan, Nikhil Joshi, Ryan Julian, Dmitry Kalashnikov, Yuheng Kuang, Isabel Leal, Lisa Lee, Tsang-Wei Edward Lee, Sergey Levine, Yao Lu, Henryk Michalewski, Igor Mordatch, Karl Pertsch, Kanishka Rao, Krista Reymann, Michael Ryoo, Grecia Salazar, Pannag Sanketi, Pierre Sermanet, Jaspiar Singh, Anikait Singh, Radu Soricut, Huong Tran, Vincent Vanhoucke, Quan Vuong, Ayzaan Wahid, Stefan Welker, Paul Wohlhart, Jialin Wu, Fei Xia, Ted Xiao, Peng Xu, Sichun Xu, Tianhe Yu, and Brianna Zitkovich. Rt-2: Vision-language-action models transfer web knowledge to robotic control, 2023.

[7] Songming Liu, Lingxuan Wu, Bangguo Li, Hengkai Tan, Huayu Chen, Zhengyi Wang, Ke Xu, Hang Su, and Jun Zhu. Rdt-1b: a diffusion foundation model for bimanual manipulation. arXiv preprint arXiv:2410.07864, 2024.

[8] Minjie Zhu, Yichen Zhu, Jinming Li, Junjie Wen, Zhiyuan Xu, Ning Liu, Ran Cheng, Chaomin Shen, Yaxin Peng, Feifei Feng, and Jian Tang. Scaling diffusion policy in transformer to 1 billion parameters for robotic manipulation. arXiv preprint arXiv:2409.14411, 2024.

[9] Chen Wang, Haochen Shi, Weizhuo Wang, Ruohan Zhang, Li Fei-Fei, and C. Karen Liu. Dexcap: Scalable and portable mocap data collection system for dexterous manipulation, 2024.

[10] Zhenyu Jiang, Yuqi Xie, Kevin Lin, Zhenjia Xu, Weikang Wan, Ajay Mandlekar, Linxi Fan, and Yuke Zhu. Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning, 2024.

[11] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. Rlbench: The robot learning benchmark learning environment, 2019.

[12] 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, 2023.

[13] Soroush Nasiriany, Abhiram Maddukuri, Lance Zhang, Adeet Parikh, Aaron Lo, Abhishek Joshi, Ajay Mandlekar, and Yuke Zhu. Robocasa: Large-scale simulation of everyday tasks for generalist robots, 2024.

[14] Yufei Wang, Zhou Xian, Feng Chen, Tsun-Hsuan Wang, Yian Wang, Katerina Fragkiadaki, Zackory Erickson, David Held, and Chuang Gan. Robogen: Towards unleashing infinite data for automated robot learning via generative simulation, 2024.

[15] Yao Mu, Tianxing Chen, Shijia Peng, Zanxin Chen, Zeyu Gao, Yude Zou, Lunkai Lin, Zhiqiang Xie, and Ping Luo. Robotwin: Dual-arm robot benchmark with generative digital twins (early version), 2024.

[16] Hao-Shu Fang, Chenxi Wang, Minghao Gou, and Cewu Lu. Graspnet-1billion: A large-scale benchmark for general object grasping. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[17] Fengshuo Bai, Yu Li, Jie Chu, Tawei Chou, Runchuan Zhu, Ying Wen, Yaodong Yang, and Yuanpei Chen. Retrieval dexterity: Efficient object retrieval in clutters with dexterous hand. arXiv preprint arXiv:2502.18423, 2025.

[18] 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.

[19] Yuanpei Chen, Yiran Geng, Fangwei Zhong, Jiaming Ji, Jiechuang Jiang, Zongqing Lu, Hao Dong, and Yaodong Yang. Bi-dexhands: Towards human-level bimanual dexterous manipulation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(5):2804–2818, 2023.

[20] Yuanpei Chen, Chen Wang, Li Fei-Fei, and C Karen Liu. Sequential dexterity: Chaining dexterous policies for long-horizon manipulation. arXiv preprint arXiv:2309.00987, 2023.

[21] 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.

[22] Matthew T Mason and J Kenneth Salisbury Jr. Robot hands and the mechanics of manipulation. 1985.

[23] Antonio Bicchi. Hands for dexterous manipulation and robust grasping: A difficult road toward simplicity. IEEE Transactions on robotics and automation, 16(6):652–662, 2002.

[24] Igor Mordatch, Zoran Popović, and Emanuel Todorov. Contact-invariant optimization for hand manipulation. In Proceedings of the ACM SIGGRAPH/Eurographics symposium on computer animation, pages 137–144, 2012.

[25] Vikash Kumar, Yuval Tassa, Tom Erez, and Emanuel Todorov. Real-time behaviour synthesis for dynamic hand-manipulation. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 6808–6815. IEEE, 2014.

[26] Yunfei Bai and C Karen Liu. Dexterous manipulation using both palm and fingers. In 2014 IEEE International Conference on Robotics and Automation (ICRA), pages 1560–1565. IEEE, 2014.

[27] Yuanpei Chen, Tianhao Wu, Shengjie Wang, Xidong Feng, Jiechuang Jiang, Stephen Marcus McAleer, Yiran Geng, Hao Dong, Zongqing Lu, Song-Chun Zhu, and Yaodong Yang. Towards human-level bimanual dexterous manipulation with reinforcement learning, 2022.

[28] Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes, 2024.

[29] Shikhar Bahl, Abhinav Gupta, and Deepak Pathak. Human-to-robot imitation in the wild. arXiv preprint arXiv:2207.09450, 2022.

[30] Priyanka Mandikal and Kristen Grauman. Dexvip: Learning dexterous grasping with human hand pose priors from video. In Conference on Robot Learning, pages 651–661. PMLR, 2022.

[31] 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.

[32] Kenneth Shaw, Shikhar Bahl, and Deepak Pathak. Videodex: Learning dexterity from internet videos. In Conference on Robot Learning, pages 654–665. PMLR, 2023.

[33] Sumedh Sontakke, Jesse Zhang, Séb Arnold, Karl Pertsch, Erdem Bıyık, Dorsa Sadigh, Chelsea Finn, and Laurent Itti. Roboclip: One demonstration is enough to learn robot policies. Advances in Neural Information Processing Systems, 36:55681–55693, 2023.

[34] Robert McCarthy, Daniel CH Tan, Dominik Schmidt, Fernando Acero, Nathan Herr, Yilun Du, Thomas G Thuruthel, and Zhibin Li. Towards generalist robot learning from internet video: A survey. arXiv preprint arXiv:2404.19664, 2024.

[35] Arpit Bahety, Priyanka Mandikal, Ben Abbatematteo, and Roberto Martín-Martín. Screwmimic: Bimanual imitation from human videos with screw space projection. arXiv preprint arXiv:2405.03666, 2024.

[36] Hanzhi Chen, Boyang Sun, Anran Zhang, Marc Pollefeys, and Stefan Leutenegger. Vidbot: Learning generalizable 3d actions from in-the-wild 2d human videos for zero-shot robotic manipulation. arXiv preprint arXiv:2503.07135, 2025.

[37] Lai Sum Yim, Quang TN Vo, Ching-I Huang, Chi-Ruei Wang, Wren McQueary, Hsueh-Cheng Wang, Haikun Huang, and Lap-Fai Yu. Wfh-vr: Teleoperating a robot arm to set a dining table across the globe via virtual reality. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 4927–4934. IEEE, 2022.

[38] Yuzhe Qin, Wei Yang, Binghao Huang, Karl Van Wyk, Hao Su, Xiaolong Wang, Yu-Wei Chao, and Dieter Fox. Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system. arXiv preprint arXiv:2307.04577, 2023.

[39] Xuxin Cheng, Jialong Li, Shiqi Yang, Ge Yang, and Xiaolong Wang. Open-television: Teleoperation with immersive active visual feedback. arXiv preprint arXiv:2407.01512, 2024.

[40] Runyu Ding, Yuzhe Qin, Jiyue Zhu, Chengzhe Jia, Shiqi Yang, Ruihan Yang, Xiaojuan Qi, and Xiaolong Wang. Bunny-visionpro: Real-time bimanual dexterous teleoperation for imitation learning. arXiv preprint arXiv:2407.03162, 2024.

[41] Jialiang Zhang, Haoran Liu, Danshi Li, Xinqiang Yu, Haoran Geng, Yufei Ding, Jiayi Chen, and He Wang. Dexgraspnet 2.0: Learning generative dexterous grasping in large-scale synthetic cluttered scenes, 2024.

[42] Pu Hua, Minghuan Liu, Annabella Macaluso, Yunfeng Lin, Weinan Zhang, Huazhe Xu, and Lirui Wang. Gensim2: Scaling robot data generation with multi-modal and reasoning llms, 2024.

[43] Huy Ha, Pete Florence, and Shuran Song. Scaling up and distilling down: Language-guided robot skill acquisition. In Conference on Robot Learning, pages 3766–3777. PMLR, 2023.

[44] Wenlong Huang, Chen Wang, Yunzhu Li, Ruohan Zhang, and Li Fei-Fei. Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation. arXiv preprint arXiv:2409.01652, 2024.

[45] Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. Smart-llm: Smart multi-agent robot task planning using large language models, 2024.

[46] Yunfei Li, Ying Yuan, Jingzhi Cui, Haoran Huan, Wei Fu, Jiaxuan Gao, Zekai Xu, and Yi Wu. Robot generating data for learning generalizable visual robotic manipulation. In 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5813–5820. IEEE, 2024.

[47] Zhengrong Xue, Shuying Deng, Zhenyang Chen, Yixuan Wang, Zhecheng Yuan, and Huazhe Xu. Demogen: Synthetic demonstration generation for data-efficient visuomotor policy learning. arXiv preprint arXiv:2502.16932, 2025.

[48] Xiu Yuan, Tongzhou Mu, Stone Tao, Yunhao Fang, Mengke Zhang, and Hao Su. Policy decorator: Model-agnostic online refinement for large policy model. arXiv preprint arXiv:2412.13630, 2024.

[49] Chengshu Li, Ruohan Zhang, Josiah Wong, Cem Gokmen, Sanjana Srivastava, Roberto Martín-Martín, Chen Wang, Gabrael Levine, Michael Lingelbach, Jiankai Sun, et al. Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation. In Conference on Robot Learning, pages 80–93. PMLR, 2023.

[50] Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. arXiv preprint arXiv:2312.08344, 2023.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The claims in the abstract and introduction match the core contributions and are supported by the method and results sections.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper includes a dedicated discussion of limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: The paper does not include formal theoretical results or proofs, as it focuses on algorithm design and empirical evaluation.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides detailed descriptions of the model architecture, training procedure, dataset, and evaluation metrics, enabling reproduction of the main results without requiring access to the code.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code are provided in supplementary materials for reproduction of all key experiments, as detailed in the supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper specifies all relevant training details, including hyperparameters, optimizer settings, and training schedules, with additional configurations provided in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification:We report the standard deviation over multiple runs to reflect variability, and clearly indicate it in both the main text and figure captions.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper specifies the use of NVIDIA RTX4090 GPUs and 4 CPU cores, along with details on training time and batch sizes for all experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research adheres to the NeurIPS Code of Ethics, with no ethical concerns related to data usage, human subjects, or potential misuse identified.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper considers positive impacts in assistive and robotics, while also noting possible misuse in surveillance and unintended physical harm from inaccurate predictions.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not release models or data with high risk for misuse; thus, no special safeguards are necessary.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use third-party assets that require attribution or license disclosure.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The core method development in this research does not involve LLMs.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.