arXiv:2606.22471v1 [cs.RO] 21 Jun 2026

Scalable Multi-Task Data Generation via Reinforcement Learning for Language-Conditioned Bimanual Dexterous Manipulation

Zechu Li[1,4], Yufeng Jin[1,2], Puze Liu[3], Jan Peters[1,3,4], Georgia Chalvatzaki[1,4]

***Abstract*— A key bottleneck in training generalist policies for bimanual dexterous manipulation is the lack of large-scale, high-quality datasets. Synthetic data generation in simulation provides a scalable alternative to human video demonstrations by overcoming challenges such as morphology mismatch, missing physical interactions, and the generation of robot actions. However, existing approaches based on human teleoperation offer limited task diversity, as object-centric trajectory matching often neglects the feasibility of robot execution. Reinforcement learning (RL) enables broader scalability but is often constrained by handcrafted, task-specific rewards. In this work, we propose a systematic RL-based data generation pipeline that integrates generalizable reward design, effective domain randomization, and language-conditioned task annotations. This pipeline synthesizes diverse, high-quality datasets for dexterous bimanual manipulation and enables training of language-conditioned multi-task policies. Our experiments show that the generated data significantly improves generalization across three representative manipulation tasks.**

I. INTRODUCTION

Bimanual dexterous manipulation is critical for enabling robots to perform complex real-world tasks. Recent advances in robotic foundation models, driven by Internet-scale pretraining and grounding with robot actions [1]–[5], have shown impressive capabilities in understanding and executing diverse manipulation skills. However, these models are typically constrained to single-arm setups or bimanual systems with simple parallel grippers, falling short of controlling dexterous hands. This gap largely stems from the absence of large-scale, high-quality, diverse, and multimodal datasets for bimanual dexterous manipulation.

To bridge this gap, multiple data sources have been explored. Human teleoperation offers natural demonstrations [6]–[8], but scaling it across diverse tasks and objects is prohibitively expensive and time-consuming. Human videos provide an abundant Internet-scale resource, yet robots face embodiment mismatches, partial observability, and retargeting imperfections that hinder robust policy learning [9]–[12]. On the other hand, simulation provides powerful data augmentation capabilities in terms of the visual appearance, objects' diversity, scene complexity, and physics parameters, enabling large-scale dataset collection [13]–[16], but its diversity is still limited and heavily dependent on the quality of expert demonstrations. Prior data augmentation approaches relying on human demonstrations [12], [13], [17] with the assumption of equivariance of object-centric trajectories, often resulting in infeasibility for the robot. For example, when a pot is rotated, the corresponding trajectory is rotated to grasp the handle without considering the complex cluster scene for collision-free execution.

Instead, leveraging Reinforcement Learning (RL) to generate synthetic data offers an attractive alternative: agents can autonomously generate diverse, high-quality trajectories by optimizing rewards and adapting to varied objects, poses, gripper types, and cluttered scenes. Yet, two major challenges hinder its practicality. First, reward engineering requires extensive human effort and must be tailored to each task, limiting scalability on the task level. Second, many RL-based methods rely on explicit state estimation (e.g., object tracking), which makes it challenging for smooth deployment in real-world settings.

To tackle these challenges, we propose Reinforcement Learning as Data Collector (RLDC), a simple yet effective method that leverages reinforcement learning to generate high-quality training data for bimanual dexterous manipulation. RLDC adopts a teacher–student paradigm: we first train per-task state-based teacher policies and then distill them into a single multi-task, diffusion-based student policy.

While conceptually straightforward, RLDC offers a scalable pipeline for constructing large-scale, multimodal datasets, paving the way for generalist robotic policies in bimanual dexterous manipulation. Our pipeline includes several innovations: (1) in the teacher stage, we **design a generalizable reward function tailored to bimanual dexterous tasks**, reducing engineering overhead and accelerating policy training; (2) for data collection, we augment the dataset with **extensive domain randomization together with grounded language task description** over objects' geometry, appearance, physical parameters, poses, lighting, and camera settings, exploiting the fully controlled simulation. The language description of each environment is further expanded using LLM-based augmentation, enriching the language label diversity; (3) finally, we distill the dataset into a **language-conditioned student policy using diffusion model** to achieve robust bimanual dexterous manipulation. To better ground the vision encoder, we introduce an auxiliary loss that anchors visual representations to the underlying state information—an advantage not possible with human teleoperation or video-only data. We evaluate RLDC both

Corresponding author: Zechu Li (zechu.li@tu-darmstadt.de).

This work was supported by the German Research Foundation (DFG) Emmy Noether Programme (CH 2676/1-2) and funded by the European Union. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

[1]Department of Computer Science, TU Darmstadt, Germany.
[2]Honda Research Institute Europe GmbH, Offenbach, Germany.
[3]DFKI, Research Department SAIROL, Darmstadt, Germany.
[4]Hessian.AI, Darmstadt, Germany

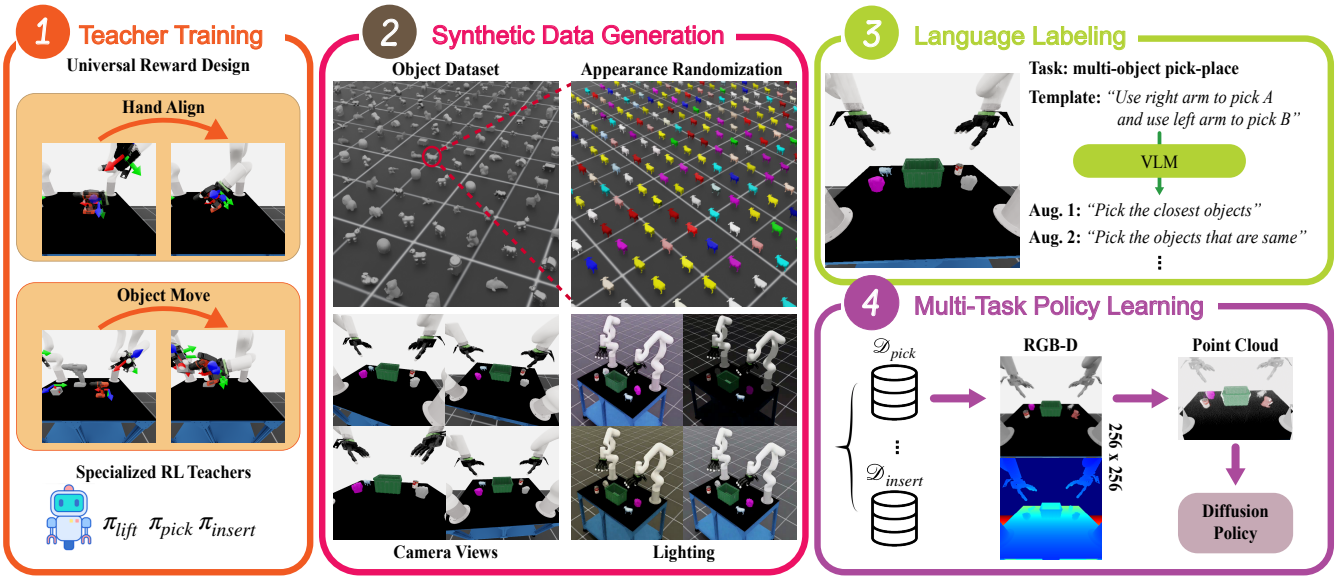


Fig. 1. We propose Reinforcement Learning as Data Collector (RLDC), a scalable pipeline for generating synthetic datasets with task-specialized RL teachers and training a language-conditioned multi-task policy for real-world bimanual dexterous manipulation. (1) A universal reward design reduces reward-engineering effort, enabling quick task scaling. (2) Diverse datasets are collected through extensive randomization of objects, physics, and visual conditions. (3) Language labels are automatically generated and further augmented with a VLM for greater diversity. (4) Finally, the collected data are distilled into a visual-based multi-task policy that transfers zero-shot to the real world.

in simulation and on real robots and demonstrate that the generalizable reward design achieves success rates comparable to finely tuned task-specific rewards, and that multi-task policies can be learned entirely from synthetic data with strong transfer to the real world.

II. RELATED WORK

A. Data Generation for Robot Manipulation

Automated data generation has emerged as a compelling approach for building large-scale robotic datasets [12], [13], [17], [18]. Typically, this paradigm starts with a small set of expert demonstrations, augments them through domain randomization, and validates the synthesized trajectories in simulation. For instance, DexMimicGen [17] decomposes tasks into subtasks and generates trajectories for each of them by applying transformations based on the relative pose between the end-effector and the object. RoboTwin 2.0 [13] leverages LLM agents for policy generation and evaluation, creating a closed-loop architecture for automated data collection. However, the diversity and quality of data produced by these methods remain heavily dependent on the original demonstrations or predefined skills, which significantly constrains their scalability and optimality.

An alternative is to exploit reinforcement learning (RL) policies as teachers in a teacher–student paradigm. Unlike augmented demonstrations, RL teacher policies are optimized under extensive randomization, allowing them to generate more diverse and higher-quality trajectories. Many sim-to-real RL methods adopt this paradigm, but they are typically restricted to specific tasks [19]–[23]. In contrast, our work investigates its effectiveness in a multi-task setting, while further enriching the data by incorporating language labels to support generalization. Another direction, exemplified by RLDG [24], explores directly training RL policies in the real world, thereby circumventing the sim-to-real gap. Although promising for precise manipulation, such real-world RL data generation approaches suffer from low training efficiency and limited scalability compared to simulation-based data generation.

B. Generalizable Reward Design

RL often requires extensive reward engineering for every single task [25]. Given the complexity of dexterous manipulation, such manual tuning is tedious and time-consuming, severely limiting scalability across diverse tasks. A natural solution is to design generalizable reward functions that can be applied broadly, thus reducing the need for task-specific engineering. Recent work has explored extracting information from human–object interaction videos, such as human hand poses and object reference trajectories [11], [12], [26], [27]. This enables rewards that generalize across different videos by encouraging robots to imitate hand poses and track object trajectories. However, simply enforcing object trajectory tracking often leads to suboptimal behaviors. For example, when placing an object into a box with randomized object and box poses, naively following the original trajectory may cause collisions with the box's sides.

A more flexible approach is to define rewards based on contact goals and object goals, allowing the agent to discover feasible trajectories while still achieving the desired final pose [28]. Our method also builds on this idea: instead of relying on contact points, we use reference hand poses as guidance, which ensures functional manipulation while maintaining generalization across tasks.

C. Sim-to-real Transfer for Dexterous Manipulation

A major challenge in sim-to-real transfer arises from visual perception, and three main approaches have been explored for deploying policies in the real world. The first is the state estimation, where object states are inferred through vision pipelines such as segmentation, pose estimation, and tracking, enabling zero-shot execution of state-based policies trained in simulation [19], [20]. While effective in principle, these pipelines are often bottlenecked by high inference latency and suffer from compounded errors due to sensor noise and model inaccuracies, especially in cluttered scenes. The second approach is to use images (RGB and/or depth) as input. Depth images typically exhibit a smaller domain gap than RGB and are therefore widely adopted [12], [28], [29]. To tackle the harder case of RGB input, prior works apply strong augmentation and multi-view learning: e.g., DextrAH-RGB [30] leverages stereo RGB with extensive visual randomization, while ManiWhere [31] employs a multi-view contrastive loss on RGB-D to achieve robust view generalization. Point cloud is an alternative to depth image, which is less constrained by camera-view discrepancies between simulation and reality and allows efficient multi-view fusion, although it requires accurate calibration [21]. Note that, in this work, we adopt point clouds with RGB information as input, combining geometric fidelity with appearance cues to improve robustness in dexterous bimanual manipulation.

III. METHOD

We propose Reinforcement Learning as Data Collector (RLDC), a scalable pipeline that leverages task-specialized RL policies to generate diverse, multimodal datasets, which are then distilled into a single multi-task diffusion policy. As illustrated in Fig. 1, RLDC consists of four stages: (1) training task-specific RL teachers using a generalizable reward design; (2) rolling out these teachers in simulation with extensive domain randomization across physical and visual factors; (3) automatically generating corresponding language labels to enrich multimodality; and (4) distilling the collected trajectories into a unified multi-task diffusion policy implemented with a decoupled transformer architecture featuring separate action heads for robust bimanual control.

A. Tasks

We evaluate our approach on three representative bimanual dexterous manipulation tasks in both simulation and real-world settings: (1) box lifting, (2) multi-object pick-and-place, and (3) drawer insertion. These tasks span different levels of coordination complexity and manipulation patterns, covering a broad range of rigid-body manipulation scenarios. Box lifting represents a single-stage, single-object task that requires precise two-arm coordination and force balance to achieve stable lifting. Multi-object pick-and-place introduces additional challenges in perception and scene understanding, including object detection and stable grasping in cluttered environments and sequential placement while avoiding inter-arm collisions. Drawer insertion further increases complexity by involving articulated object interaction, along with precise timing and bimanual cooperation for successful object insertion. Collectively, these tasks encompass high inter-arm coordination, long-horizon planning, cluttered scene reasoning, and interaction with articulated objects, which provide a comprehensive and challenging testbed for validating the effectiveness and generalization capability of our method.

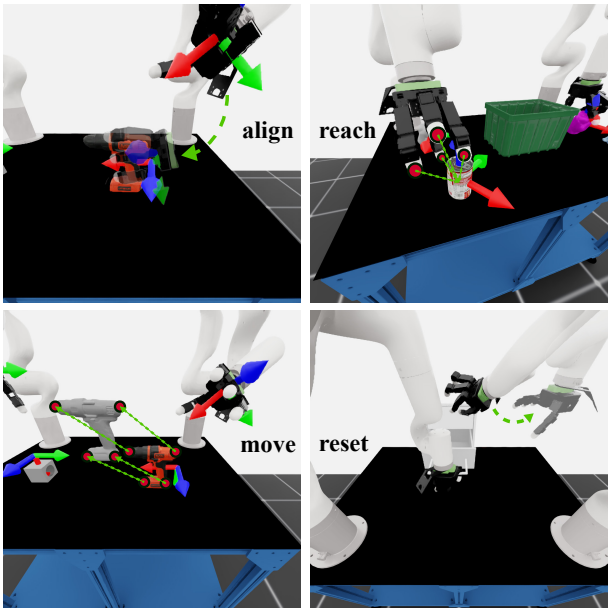


Fig. 2. Four reward terms. (Top-Left) A predefined hand pose associated with the drill guides the hand to align accordingly. (Top-Right) The distance between the fingertips, marked by red points, and the object center is minimized to encourage contact. (Bottom-Left) The object is rewarded for moving toward a defined goal pose in the air. (Bottom-Right) After task completion, e.g., placing the object into the drawer, the robot is encouraged to return to a reset configuration.

B. Generalizable Reward Design

RL often relies on carefully designed reward functions that are specific to each task, which hinders its scalability especially for challenging bimanual manipulation with multi-fingered hands. It is critical to construct a set of generalizable reward terms that can be reused and adapted, reducing engineering overhead and accelerating the prototyping cycle. Thus, we introduce the following four reward terms that are generalizable enough to solve the aforementioned three tasks with the same set of scaling factors.

a) Initial manipulation hand pose: The initial hand pose relative to the object strongly impacts functional success [32], [33]. We define a task-specific initial hand pose for each hand and attach the pose onto the object, as shown in Fig. 2 (top-left). We thus encourage the pose alignment:

$$r_{\text{align}} = -\alpha \left\| p_{\text{palm}} - p_{\text{def}} \right\|^2 - \beta d_{\text{quat}}(q_{\text{palm}}, q_{\text{def}}), \quad (1)$$

where p_{palm} and q_{palm} are the current robot hand pose and p_{def} and q_{def} are the predefined initial hand pose depending on the object's initialization. d_{quat} is the distance between two quaternions, and α and β are scaling factors.

b) Object contact: After aligning the robot hand with the initial manipulation pose, we encourage the hand to establish contact with the object. Specifically, we minimize the distance between the fingertips and the object's center of mass. This term ensures reliable contact initiation, providing a stable starting point for subsequent manipulation:

$$r_{\text{contact}} = \sum_{i=1}^{n} \frac{1}{1 + \alpha^i \left\| p_{\text{tip}}^i - p_{\text{com}} \right\|^2}, \tag{2}$$

where n is the number of fingertips, p_{tip}^i is the position of the i-th fingertip, p_{com} denotes the object center of mass, and α^i is the corresponding fingertip weight.

c) Object goal poses with lifting gate: Objects typically have designated target poses associated with task completion, e.g., placing an object into a box. We therefore define a goal pose for each object, where orientation constraints can be optionally included depending on task requirements. Prior works enforce full object pose trajectories, which are prone to errors when retargeting from human videos and often suboptimal under extensive pose randomization [11], [12], [26], [27]. Instead, we reward progress toward flexible goal poses, allowing the RL agent to explore diverse strategies for achieving the target. To prevent undesirable behaviors such as pushing objects rather than manipulating them in the air, we further introduce a lifting condition: the reward is only activated once the object is lifted above a height threshold:

$$r_{\text{goal}} = \begin{cases} \max(d_z, 0), & \text{if } d_z < \tau, \\ \exp\left(-\alpha \| p_{\text{o}} - p_{\text{g}} \|^2 - \beta \, d_{\text{quat}}(q_{\text{o}}, q_{\text{g}})\right), & \text{otherwise.} \end{cases} \tag{3}$$

where $d_z = |p_{\text{o},z} - \hat{p}_{\text{o},z}|$ and $\hat{p}_{\text{o},z}$ is the height of initial object position and τ is the lifting threshold.

d) Robot reset: We further include a reset term that encourages the robot arm to return to a predefined configuration once the sub-task is completed, e.g., the object is already inside the drawer (Fig. 2 (bottom-right)). This term not only prevents the robot from executing irrelevant or unsafe behaviors during bimanual interaction but also provides a consistent starting state that facilitates chaining of tasks and smooth integration into long-horizon tasks:

$$r_{\text{reset}} = \exp\left(-\alpha \| \theta - \theta_{\text{reset}} \|^2\right), \tag{4}$$

where θ_{reset} is the reset joint configuration.

C. Policy Training and Multimodal Data Collection

We obtain expert policies for each simulation task by optimizing the proposed generalizable rewards. Learning even a single bimanual dexterous manipulation task is challenging due to the high-dimensional observation and action spaces and the inherent difficulty of credit assignment. To address this, we formulate the bimanual problem as a multi-agent setting and adopt independent PPO (IPPO) [34], following [19]. Each arm is controlled by a dedicated policy conditioned on its local observations, with its own value function defined over the local observations and rewards. This decomposition reduces exploration space, simplifies credit assignment, and improves learning efficiency. During teacher policy training, we apply domain randomization to factors that directly affect state-based learning, such as object initialization and physical parameters. Visual attributes are randomized only during subsequent data collection described below.

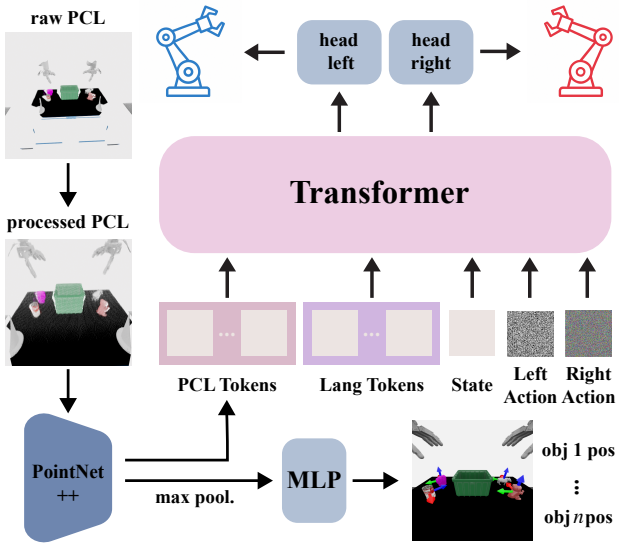


Fig. 3. Network architecture. The raw point cloud is cropped to the region of interest and downsampled. Point cloud features, language tokens, robot states, and noisy actions are processed by a transformer, after which separate heads predict the action noises for the right and left hands. An additional MLP predicts the ground-truth object states.

Once the RL teacher policies converge, we construct a diverse multimodal dataset by rolling them out in simulation. To maximize diversity, we employ extensive domain randomization as shown in Fig. 1(2). For objects, we draw from the dataset in [21], which includes a wide range of complex and nonconvex geometries such as cars, shoes, and toys. Each object is randomized in pose, physical parameters (e.g., static and dynamic friction, mass, restitution), and visual attributes (e.g., color, reflectivity, roughness). Specifically, we select 100 different objects and process them into similar scale that are suitable for grasping, and 20 color categories. We also randomize perception-related factors including camera viewpoints, intrinsic parameters, and lighting conditions. This combination of physical and visual perturbations exposes the policy to a broad distribution of scenarios, enhancing its robustness, and improving the transfer to real-world settings.

D. Automated Language Labeling

One major advantage of simulation-based data generation is the ability to automatically generate language labels for every trajectory [13]. We predefine language annotations for objects, colors, and goals, and then instantiate task-specific descriptions at each environment reset using a template format. For instance, in the multi-object pick-and-place task in Fig. 1(3), we generate the instruction: "use the right arm to place the red car into the bin, and the left arm to place the blue shoe into the bin." To increase linguistic diversity, we employ GPT [35] to paraphrase the templates:

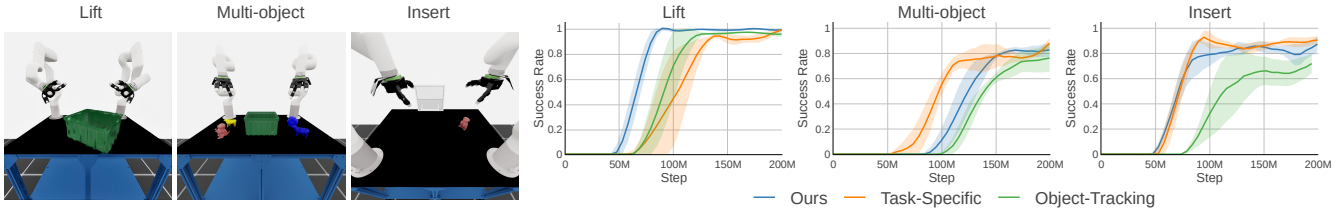


Fig. 4. (Left) Visualization of three bimanual dexterous manipulation tasks in simulation. (Right) Performance of the proposed reward design compared to baselines. Our rewards achieve performance comparable to task-specific designs and outperform the object reference trajectory baseline.

"pick the nearest objects with both arms and place them respectively." The augmentation enriches the dataset with natural language variability, improving the generalization ability to unseen instructions. The language input is primarily used to disambiguate task goals, object choices, and arm-object assignments within the shared multi-task policy, rather than to improve single-task imitation in isolation.

E. Multi-task Policy Distillation

After constructing a large-scale multimodal dataset, our goal is to train a multi-task policy capable of executing diverse tasks conditioned on language instructions. Exploiting such large-scale synthetic data is challenging, so we model the policy as a transformer-based diffusion model due to its expressive capacity. The denoising network ϵ_θ takes as input a 6-channel point cloud O along with the robots' state S and language tokens L and predicts the noise ϵ^k required for the diffusion process, where ϵ^k is the noise at k iteration by doing forward diffusion process at a data point a^0. The action loss is defined as $\ell_{\text{action}} = \left\|\epsilon^k - \epsilon_\theta(\bar{\alpha}_k a^0 + \bar{\beta}_k \epsilon^k, k, O, S, L)\right\|^2$, where $\bar{\alpha}_k$ and $\bar{\beta}_k$ are noise schedulers. In practice, we train with 20 denoising steps and adopt 5 inference steps using the DDIM [36] sampler.

a) Network architecture (Fig.3).: We first preprocess the raw point cloud by cropping it to the workspace region of interest (e.g., the tabletop) and randomly downsampling it to $2,048$ points. The processed points are encoded using a PointNet++ [37] encoder without max pooling to preserve spatial structure, and the resulting point features are projected onto a 256-dimensional embedding space. In parallel, the low-level robot state, including joint positions and velocities, is embedded using Fourier positional encodings to improve temporal discriminability across steps.

Bimanual manipulation often requires the two arms to perform actions that are partially independent yet coordinated [38]. To reflect this structure, we decouple the action representation into separate queries for each side. Through cross-attention, these action queries interact with the conditioning tokens, which consist of point cloud features, robot state features, and language embeddings encoded via SigLip [39], at linear complexity $O(n)$ rather than quadratic $O(n^2)$. Finally, two output heads decode the refined action tokens to predict the diffusion noise for each arm.

b) Auxiliary loss for state prediction.: A unique advantage of simulation-based data is the availability of ground-truth object states, which can serve as anchors for perception. We leverage this opportunity by introducing an auxiliary prediction head that maps the point cloud features to object states $\ell_{\text{aux}} = \left\|\hat{x}_{\text{obj}} - x_{\text{obj}}\right\|^2$, where x_{obj} is the object state including positions and orientations and $\hat{x}_{\text{obj}}$ is the prediction. Note that for the continuity of the rotation and smoothness of learning, we adopt 6D representations [40]. This auxiliary supervision regularizes the visual encoder, encouraging it to extract task-relevant geometric information and improving policy robustness in vision-conditioned settings.

IV. EXPERIMENTS

A. Real-World and Simulation Setup

Our robotic platform consists of two 6-degree-of-freedom (DOF) xArm UF850 arms, each equipped with a 16-DOF Allegro Hand V4, yielding a total of 44 degrees of freedom. The system operates over a 1.2×0.8m tabletop workspace under standard safety constraints. A low-level joint-level PD controller runs at 120Hz, while policy inference is executed at 20Hz. Perception is provided by a single egocentric ZED2i RGB-D camera mounted in-between the arms. We use the NVIDIA IsaacLab [41] as our simulator and construct a digital twin that accurately replicates the real-world setup. We follow [28] to identify the physics parameters in simulation to ensure smooth sim-to-real transfer.

B. Evaluation of Generalizable Reward Design

To construct datasets across diverse tasks, the first question to answer is whether the proposed generalizable reward design can achieve high success rates and transfer effectively across different manipulation tasks. We compare our reward design against two baselines: (i) task-specific rewards manually designed for each task as in [19], and (ii) the object reference trajectory reward from [27], a widely adopted approach in recent work (see Section II-B). Note that we use the **same reward terms and scaling factors** for all tasks, emphasizing the scalability for reward tuning.

Learning efficiency and success rate. Our proposed reward design achieves comparable learning efficiency and overall success rates compared to the task-specific reward. As shown in Fig. 4, it performs comparably on the *Insert* task and outperforms it on *Lift*. We attribute this improvement to the fact that overly constrained task-specific rewards may hinder the learning, whereas simpler rewards allow more flexible exploration. On the *Multi-Object* task, however, our performance is lower than with task-specific rewards because

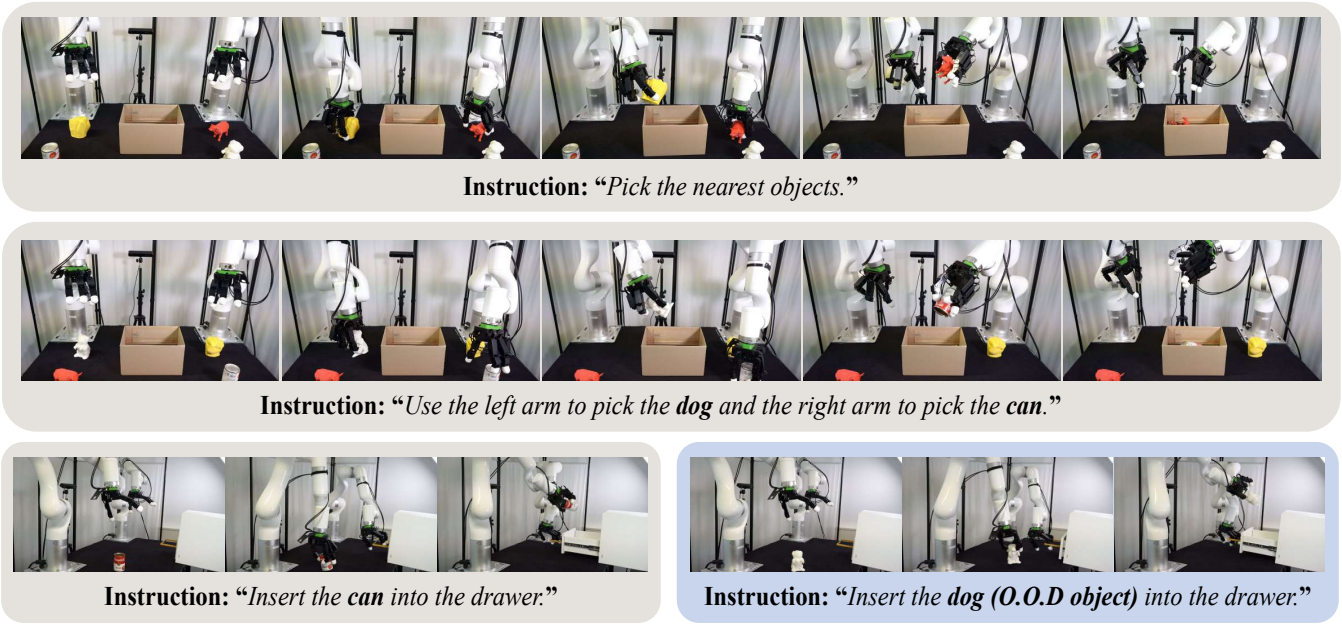


Fig. 5. Visualization of real-world results. The first two rows show the *Multi-Object* task under different language instructions, while the last row illustrates the *Insert* task with both in-domain and out-of-domain objects.

aligning hand poses with multiple objects introduces constraints that slow down learning. This suggests that our four reward terms can be selectively combined depending on the task to relax the constraints. By contrast, the object reference trajectory reward performs poorly across all tasks. The key limitation is its reliance on predefined trajectories: when objects are heavily randomized, the robot often encounters joint limits that while following the trajectory. These results strengthen our key idea: instead of forcing adherence to a fixed trajectory, it is more effective to provide flexible goal conditions that allow the RL agent to discover.

Implementation efficiency. In addition to performance, scalability requires efficient reward design across tasks. Empirically, the most time-consuming component of our approach is defining the initial hand alignment pose. Since only one pose per hand is required, reward setup takes roughly one hour per task. In contrast, setting up object reference trajectory rewards can take up to a full day, as it involves video recording, object tracking, and coordinate alignment between the video and the simulator.

C. Evaluation of Policy Distillation

We evaluate the distilled vision-based policy in both simulation and real-world settings, considering both single-task and multi-task variants. Each task has equal percentage in the constructed multi-task dataset. Simulation results are obtained with 256 parallel environments, while real-world performance is measured over 20 trials per experiment.

Comparison with SOTA baselines. We compare our approach against two recent point-cloud-based policies, DP3 [42] and iDP3 [43]. The key architectural differences lie in the point cloud encoder: DP3 employs a linear encoder, iDP3 uses a pyramid convolutional encoder, and our method adopts a PointNet++ encoder (without max pooling) to better preserve spatial features. For fairness, we standardize the input size to $2,048$ points, which lies between the settings used in the baselines ($1,024$ and $4,096$). As shown in Tab. I, our policy consistently outperforms both DP3 and iDP3 in single-task and multi-task settings. Interestingly, the performance gap between DP3 and iDP3 is smaller than reported in their original papers. We attribute this to differences in data: our dataset, with on the order of 10^3 trajectories collected from converged RL policies, is orders of magnitude larger and more consistent than human-teleoperated datasets with only 10^1 trajectories, which likely reduces sensitivity to architectural differences.

In Fig. 5, we visualize real-world rollouts, showing that the multi-task policy successfully follows diverse language instructions, enabled by the augmented language labels.

	Multi-Object	*Insert*	*Lift*	*Multi-Task*
DP3 [42]	0.42	0.38	0.703	0.06
iDP3 [43]	0.41	0.47	0.65	0.14
Ours	0.80	0.74	**0.96**	0.67
w/o sep. head	**0.812**	**0.79**	0.93	**0.70**
w/o state pre.	0.72	0.68	0.85	0.57

	Multi-Object	*Insert*	*Lift*	*Multi-Task*
DP3 [42]	10/20	5/20	10/20	4/20
iDP3 [43]	8/20	3/20	11/20	2/20
Ours	**15/20**	**16/20**	14/20	**8/20**
w/o sep. head	11/20	8/20	**16/20**	2/20
w/o state pre.	10/20	12/20	8/20	3/20

TABLE I

(TOP) SIMULATION AND (BOTTOM) REAL-WORLD RESULTS COMPARING BASELINES DP3 [42] AND IDP3 [43] AND ABLATED DESIGN CHOICES.

Furthermore, in the *Insert* task, we evaluate an out-of-distribution object (a dog) that was excluded from the *Insert* training set but included in the *Multi-Object* task. Due to the multi-task training paradigm, the policy effectively transfers knowledge from the multi-object dataset to the novel task, highlighting the potential of learning multi-task policies from specialized RL teachers.

Ablations on design choices. We further investigate two design choices: (i) the use of separate action heads and (ii) the state-prediction auxiliary loss. As shown in Tab. I, removing separate action heads and using a single shared head achieves higher success rates, we hypothesize that it is easier to fit the training distribution. However, in the real world, the separated heads yield significantly higher success rates except *Lift*. In practice, we observe that independent heads increase robustness to out-of-distribution cases arising from sim-to-real gaps. For example, if one hand grasps an object with an unexpected behavior, the other hand remains unaffected, allowing the policy to recover more effectively. However, since *Lift* is a single-object manipulation task, both hands will be affected if the box is tilted and therefore a shared observation and action space is preferred. Incorporating the state-prediction auxiliary loss consistently improves performance across tasks. This leverages simulation's unique advantage of providing ground-truth object states, effectively anchoring the visual encoder to task-relevant features. We view this as a "cherry on the cake", and recommend including this auxiliary signal whenever policies are distilled from simulation data.

Dataset Size	*Multi-Object*		*Insert*		*Multi-Task*	
	Sim	Real	Sim	Real	Sim	Real
500	0.590	9/20	0.351	5/20	0.398	3/20
2,000	0.758	13/20	0.539	7/20	0.519	3/20
5,000	**0.801**	**15/20**	**0.738**	**16/20**	**0.672**	**8/20**

TABLE II

RESULTS OF POLICIES TRAINED WITH VARYING DATASET SIZES.

Impact of dataset size. As shown in Tab. II, our transformer-based architecture scales effectively as the dataset size increases. We attribute this to both the model capacity of the transformer and the underlying data distribution within the dataset. Since only successful trajectories are collected, corner-case initializations are more likely to fail, leading to an imbalanced dataset. With larger datasets, these rare cases are better included, allowing the student policy to imitate more robust strategies. We also observe that the effect of dataset size depends on task complexity. For example, the *Multi-Object* task is relatively straightforward, requiring less inter-arm coordination and no interaction with articulated objects. In contrast, the *Insert* task is more complex, and therefore suffers greater performance degradation as dataset size decreases. This finding highlights both the importance of dataset scale and the critical role of large-scale synthetic data generation.

V. CONCLUSIONS

In this work, we proposed RLDC, a scalable pipeline that leverages reinforcement learning to generate high-quality, diverse training data for bimanual dexterous manipulation. The scale and diversity of training data are increasingly critical for developing multi-task and foundation models in robotics. In response, we show that synthetic data generated by RL with extensive randomization and automatic language labeling can efficiently produce large-scale multimodal datasets. Unlike direct augmentation of human demonstrations, RL-generated data are collected from closed-loop robot policies in simulation, which encourages consistency with robot kinematics and task feasibility under randomized conditions.

To enable scalable RL-based data collection, we introduced several innovations: (1) a generalizable reward function tailored to bimanual dexterous manipulation reduces engineering overhead and accelerates policy prototyping; (2) converged teacher policies are rolled out to collect diverse trajectories with extensive randomization and grounded language instructions; and (3) the dataset is distilled into a diffusion-based student policy with visual anchoring for robust sim-to-real transfer. Experiments show that our reward design achieves success rates comparable to finely tuned task-specific rewards and that multi-task policies can be learned entirely from synthetic data with strong real-world transfer.

Limitations and future work. Our study has two key limitations. First, setting up simulation environments and initial hand poses remain time-consuming. While reward engineering accounts for much of this cost, additional effort is required to tune robot physics parameters, prepare assets, and define success criteria. Recent work [28] suggests that parallel simulation can accelerate physics parameter tuning. Community resources such as [13]–[16] increasingly provide reusable assets and environments to mitigate this burden, and another line of work [26], [44] focuses on automating real-to-sim that rapidly reconstruct real-world environments in simulation. Second, RLDC inherits the behavior biases of its RL teachers. Although the resulting trajectories are robot-feasible, they are not necessarily human-like and can contain unnecessary finger motions or abrupt releases. This limitation is especially relevant for delicate objects and tasks requiring controlled placement, fine in-hand regrasping, finger gaiting, or deformable-object interaction. Future work could combine RL-generated trajectories with human demonstrations, smoothness/contact regularization, or release-quality constraints to obtain both feasible and natural behaviors.

REFERENCES

[1] K. Black, N. Brown, D. Driess, A. Esmail, M. Equi, C. Finn, N. Fusai, L. Groom, K. Hausman, B. Ichter *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[2] O. M. Team, D. Ghosh, H. Walke, K. Pertsch, K. Black, O. Mees, S. Dasari, J. Hejna, T. Kreiman, C. Xu *et al.*, "Octo: An open-source generalist robot policy," *arXiv preprint arXiv:2405.12213*, 2024.

[3] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi *et al.*, "Openvla: An open-source vision-language-action model," *arXiv preprint arXiv:2406.09246*, 2024.

[4] J. Wen, Y. Zhu, J. Li, M. Zhu, Z. Tang, K. Wu, Z. Xu, N. Liu, R. Cheng, C. Shen *et al.*, "Tinyvla: Towards fast, data-efficient vision-language-action models for robotic manipulation," *IEEE Robotics and Automation Letters*, 2025.

[5] M. Reuss, H. Zhou, M. Rühle, Ö. E. Yağmurlu, F. Otto, and R. Lioutikov, "Flower: Democratizing generalist robot policies with efficient vision-language-action flow policies," *arXiv preprint arXiv:2509.04996*, 2025.

[6] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," *The International Journal of Robotics Research*, 2024.

[7] Z. Fu, T. Z. Zhao, and C. Finn, "Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation," in *Conference on Robot Learning (CoRL)*, 2024.

[8] N. Funk, J. Urain, J. Carvalho, V. Prasad, G. Chalvatzaki, and J. Peters, "Actionflow: Equivariant, accurate, and efficient policies with spatially symmetric flow matching," *arXiv preprint arXiv:2409.04576*, 2024.

[9] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system," *arXiv preprint arXiv:2307.04577*, 2023.

[10] R.-Z. Qiu, S. Yang, X. Cheng, C. Chawla, J. Li, T. He, G. Yan, D. J. Yoon, R. Hoque, L. Paulsen *et al.*, "Humanoid policy human policy," *arXiv preprint arXiv:2503.13441*, 2025.

[11] T. G. W. Lum, O. Y. Lee, C. K. Liu, and J. Bohg, "Crossing the human-robot embodiment gap with sim-to-real rl using one human demonstration," *arXiv preprint arXiv:2504.12609*, 2025.

[12] Z. Yuan, T. Wei, L. Gu, P. Hua, T. Liang, Y. Chen, and H. Xu, "Hermes: Human-to-robot embodied learning from multi-source motion data for mobile dexterous manipulation," *arXiv preprint arXiv:2508.20085*, 2025.

[13] T. Chen, Z. Chen, B. Chen, Z. Cai, Y. Liu, Q. Liang, Z. Li, X. Lin, Y. Ge, Z. Gu *et al.*, "Robotwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation," *arXiv preprint arXiv:2506.18088*, 2025.

[14] S. Nasiriany, A. Maddukuri, L. Zhang, A. Parikh, A. Lo, A. Joshi, A. Mandlekar, and Y. Zhu, "Robocasa: Large-scale simulation of everyday tasks for generalist robots," *arXiv preprint arXiv:2406.02523*, 2024.

[15] H. Geng, F. Wang, S. Wei, Y. Li, B. Wang, B. An, C. T. Cheng, H. Lou, P. Li, Y.-J. Wang *et al.*, "Roboverse: Towards a unified platform, dataset and benchmark for scalable and generalizable robot learning," *arXiv preprint arXiv:2504.18904*, 2025.

[16] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu *et al.*, "Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments," in *Conference on robot learning*. PMLR, 2022, pp. 477–490.

[17] Z. Jiang, Y. Xie, K. Lin, Z. Xu, W. Wan, A. Mandlekar, L. J. Fan, and Y. Zhu, "Dexmimicgen: Automated data generation for bimanual dexterous manipulation via imitation learning," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 16 923–16 930.

[18] J. Wang, Y. Qin, K. Kuang, Y. Korkmaz, A. Gurumoorthy, H. Su, and X. Wang, "Cyberdemo: Augmenting simulated human demonstration for real-world dexterous manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 17 952–17 963.

[19] Z. Li, Y. Jin, D. O. Apraez, C. Semini, P. Liu, and G. Chalvatzaki, "Morphologically symmetric reinforcement learning for ambidextrous bimanual manipulation," *arXiv preprint arXiv:2505.05287*, 2025.

[20] T. Lin, Z.-H. Yin, H. Qi, P. Abbeel, and J. Malik, "Twisting lids off with two hands," in *8th Annual Conference on Robot Learning (CoRL)*, 2024. [Online]. Available: https://openreview.net/forum?id=3wBqoPfoeJ

[21] T. Chen, M. Tippur, S. Wu, V. Kumar, E. Adelson, and P. Agrawal, "Visual dexterity: In-hand dexterous manipulation from depth," in *Icml workshop on new frontiers in learning, control, and dynamical systems*, 2023.

[22] H. Qi, A. Kumar, R. Calandra, Y. Ma, and J. Malik, "In-hand object rotation via rapid motor adaptation," in *Conference on Robot Learning*. PMLR, 2023, pp. 1722–1732.

[23] T. Chen, E. Cousineau, N. Kuppuswamy, and P. Agrawal, "Vegetable peeling: A case study in constrained dexterous manipulation," *arXiv preprint arXiv:2407.07884*, 2024.

[24] C. Xu, Q. Li, J. Luo, and S. Levine, "Rldg: Robotic generalist policy distillation via reinforcement learning," *arXiv preprint arXiv:2412.09858*, 2024.

[25] J. Eschmann, "Reward function design in reinforcement learning," *Reinforcement learning algorithms: Analysis and Applications*, pp. 25–33, 2021.

[26] P. Dan, K. Kedia, A. Chao, E. W. Duan, M. A. Pace, W.-C. Ma, and S. Choudhury, "X-sim: Cross-embodiment learning via real-to-sim-to-real," *arXiv preprint arXiv:2505.07096*, 2025.

[27] Z. Chen, S. Chen, E. Arlaud, I. Laptev, and C. Schmid, "Vividex: Learning vision-based dexterous manipulation from human videos," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 3336–3343.

[28] T. Lin, K. Sachdev, L. Fan, J. Malik, and Y. Zhu, "Sim-to-real reinforcement learning for vision-based dexterous manipulation on humanoids," *arXiv preprint arXiv:2502.20396*, 2025.

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

[30] R. Singh, A. Allshire, A. Handa, N. Ratliff, and K. Van Wyk, "Dextrah-rgb: Visuomotor policies to grasp anything with dexterous hands," *arXiv preprint arXiv:2412.01791*, 2024.

[31] Z. Yuan, T. Wei, S. Cheng, G. Zhang, Y. Chen, and H. Xu, "Learning to manipulate anywhere: A visual generalizable framework for reinforcement learning," *arXiv preprint arXiv:2407.15815*, 2024.

[32] J. Urain, N. Funk, J. Peters, and G. Chalvatzaki, "Se (3)-diffusionfields: Learning smooth cost functions for joint grasp and motion optimization through diffusion," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 5923–5930.

[33] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies Between Affordance and Geometry: 6-DoF Grasp Detection via Implicit Representations," in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.

[34] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[35] A. Hurst, A. Lerer, A. P. Goucher, A. Perelman, A. Ramesh, A. Clark, A. Ostrow, A. Welihinda, A. Hayes, A. Radford *et al.*, "Gpt-4o system card," *arXiv preprint arXiv:2410.21276*, 2024.

[36] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[37] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[38] J.-J. Jiang, X.-M. Wu, Y.-X. He, L.-A. Zeng, Y.-L. Wei, D. Zhang, and W.-S. Zheng, "Rethinking bimanual robotic manipulation: Learning with decoupled interaction framework," *arXiv preprint arXiv:2503.09186*, 2025.

[39] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 11 975–11 986.

[40] Y. Zhou, C. Barnes, J. Lu, J. Yang, and H. Li, "On the continuity of rotation representations in neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 5745–5753.

[41] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," *IEEE Robotics and Automation Letters*, vol. 8, no. 6, pp. 3740–3747, 2023.

[42] Y. Ze, G. Zhang, K. Zhang, C. Hu, M. Wang, and H. Xu, "3d diffusion policy: Generalizable visuomotor policy learning via simple 3d representations," *arXiv preprint arXiv:2403.03954*, 2024.

[43] Y. Ze, Z. Chen, W. Wang, T. Chen, X. He, Y. Yuan, X. B. Peng, and J. Wu, "Generalizable humanoid manipulation with 3d diffusion policies," *arXiv preprint arXiv:2410.10803*, 2024.

[44] M. Torne, A. Simeonov, Z. Li, A. Chan, T. Chen, A. Gupta, and P. Agrawal, "Reconciling reality through simulation: A real-to-sim-to-real approach for robust manipulation," *arXiv preprint arXiv:2403.03949*, 2024.