Data and Learning Where it Matters for Contact-Rich Manipulation

Anonymous Author(s)
Affiliation
Address
email

Abstract: Learned policies trained end-to-end on large datasets often remain brittle in high-precision tasks and struggle with generalization. We find that these limitations largely stem from a lack of structure and focus in data collection. Our key insight is to leverage dense data collection only for the critical segment of contact-rich tasks and to rely on traditional planning during simple free-space motion. We propose an automated data-collection scheme in combination with offline deep reinforcement learning for the critical segment of the task, eliminating reliance on a teleoperator's skill and on online policy updates. Across four challenging real-world tasks, using only 2–2.5 h of autonomous data collection, we achieve an average success rate of 96 %, compared to the strongest baseline at 55 %. Notably, performance remains high in out-of-distribution scenarios where end-to-end approaches struggle. Our results pave the way for targeted data collection for contact-rich tasks and for high success rates in precision applications. *We will upload all our videos, training datasets, and evaluation datasets. Website.*

Keywords: Data for Robotics, Learning, Manipulation, Foundation Models, DRL

1 Introduction

End-to-end robot learning has recently become very popular [1, 2, 3], especially in contact-rich manipulation tasks that are hard to model analytically. This progress has been further accelerated by large-scale data collection [4, 5]. Yet certain limitations of end-to-end policies become apparent: they struggle with high-precision tasks, and generalization to out-of-distribution (OOD) scenarios remains limited. It is still unclear whether simple data scaling will close this gap, especially given that data collection pipelines remain expensive [6, 7].

We argue that at least part of this gap is not merely a matter of scale, but of *what* data is collected and *how* tasks are structured. Specifically, we show that high-precision tasks usually fail at the contact-rich, critical segment — such as an insertion with tight clearance. The other parts of the task typically involve unconstrained free-space motion and are thus easier to handle. End-to-end approaches treat these fundamentally different phases uniformly during data collection and policy learning [1, 2, 3, 6, 7].

We propose to exploit this structure explicitly. Our key insight is to *densely collect data where it matters* — at the critical segment. For the remainder of the task, we go to the other extreme and solve it without using robot-specific data, using off-the-shelf pose estimation and motion planning. Recent advances in language-guided image segmentation [8] and pose estimation [9] make such sequential pipelines increasingly practical for real-world manipulation while reducing the reliance on robot data.

First, we introduce a scheme for autonomous data acquisition that performs dense data collection at the critical segment, in combination with offline deep reinforcement learning (DRL) for policy learning. The autonomous data-acquisition scheme eliminates the constant dependency on a human teleoperator. For deployment, we propose combining the learned policy with motion planning for

Submitted to the 10th Conference on Robot Learning (CoRL 2026). Do not distribute.

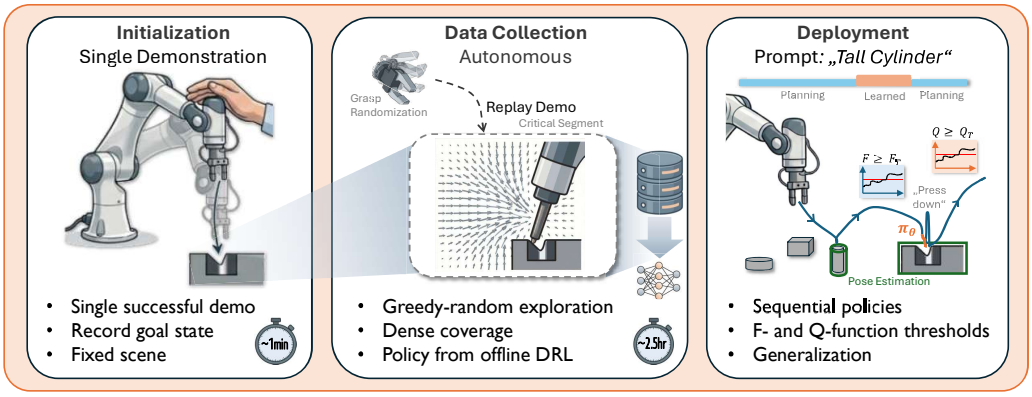


Figure 1: **Method**. (**Left**) First, we record a single demonstration in the scene using kinesthetic teaching. The scene layout remains unchanged for the subsequent data collection. (**Middle**) The demonstration is replayed until the critical segment is reached, and then we execute a mixed greedy-random policy π_{explore} for dense data collection, yielding successful and failed rollouts. We combine data collection with offline DRL to obtain a learned policy π_θ. (**Right**) In deployment, π_θ for the contact-rich segment is sequentially chained with off-the-shelf pose estimation and motion planning.

free-space motion and switching policies based on contact events and Q-function values. We evaluate our approach on four challenging real-world tasks and difficult out-of-distribution scenarios. Using only $2\,\text{h}$ to $2.5\,\text{h}$ of wall-clock time for autonomous data collection, our method achieves success rates of 94 to $98\,\%$ across four real-world tasks and $96\,\%$ on average, outperforming baselines that achieve no more than $55\,\%$. While end-to-end methods struggle under out-of-distribution conditions, our approach maintains high success rates.

In summary, we present three core contributions. (**1**) We propose a compositional framework that leverages dense data collection and a learned policy in an offline manner for the critical task segment, while relying on motion planning for free-space motion. (**2**) We introduce an autonomous data-collection scheme that leverages mixed greedy-random exploration for dense data collection at the critical segment. (**3**) During deployment, we integrate planning and learning using contact events and Q-functions for policy switching, achieving success rates above 94% across all tasks.

2 Related Work

Data collection strategies. Imitation Learning typically relies on teleoperation data [2, 10, 11]. DRL on hardware similarly requires human intervention [12]. Automated data acquisition schemes are mostly limited to simulation. Assembly-by-disassembly generates demonstrations by automatic disassembly and trains a DRL policy using imitation rewards [13]. Other approaches use search for data generation [14, 15] or to augment initial human demonstrations [16]. In contrast, we propose an automated data-acquisition scheme for the real world for the critical task segment.

Contact-rich manipulation. Methods relying on pure motion planning use fine-tuned pose estimation to achieve high success rates [17, 18, 19]. Learned policies achieve close to 100% when ground-truth 6D object poses are available [13, 20, 21]. HIL-SERL demonstrated 100% success rate with DRL and vision input, but only for short-horizon tasks [12, 22]. Imitation policies typically achieve around 80% success rate on complex tasks [2, 23, 24], and we show that their failure cases are concentrated at the critical segment. For traditional control methods, direct force or hybrid force-position control has been proven essential for contact-rich tasks [25, 26], with first learning-based models adopting the approach [3, 27, 28, 29].

Modular Policies. A large body of work separates different stages of manipulation tasks into dedicated sub-policies [30, 31], which can also be trained independently on heterogeneous data and combined through diffusion guidance [32] or a learned router [33]. For robotic foundation models, mixture-of-experts architectures have become popular, enabling scalable multitask learning by spe-

cializing lightweight experts across behaviors [34, 35]. In contrast to prior approaches that compose multiple learned policies, we exploit the structure of contact-rich manipulation itself, learning only the critical segment to improve robustness and OOD generalization.

3 Method

3.1 Problem Setup

We model contact-rich interaction as an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, \rho, \mathcal{P}, r, \gamma)$, where $\mathbf{s}_t \in \mathcal{S}$ and $\mathbf{a}_t \in \mathcal{A}$ are the state and action at time step t, $\rho(\mathbf{s}_0)$ is a distribution over initial states, $\mathcal{P}(\mathbf{s}_{t+1} \mid \mathbf{s}_t, \mathbf{a}_t)$ are the unknown transition dynamics, $r_t = r(\mathbf{s}_t, \mathbf{a}_t) \in \mathbb{R}$ is a reward function encoding the task, and $\gamma \in [0, 1)$ is a discount factor. We observe the scene with a single wrist camera and obtain state estimates of task-relevant objects using pose estimation. Our goal is to collect a dataset for training a policy $\pi_\theta : \mathcal{S} \to \mathcal{A}$ for the critical segment parameterized by a neural network with parameters θ that maximizes the expected discounted return $J(\pi_\theta) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$. An overview of our methodology is provided in Figure 1.

3.2 Autonomous Offline Data Collection

We first set up the scene with the objects at fixed locations in the workspace. If necessary, fixtures can be 3D printed to keep the objects in place (Appendix A.1). Then, we record a single demonstration that performs the task using, for example, kinesthetic teaching, and put all objects back to their initial locations. Our further scheme relies on keeping the scene layout unchanged during data collection.

We start by replaying the demonstration trajectory until the critical segment of the task is reached. In practice, we define the critical task segment based on the uncertainty of the pose estimation (Appendix A.5). Then, we execute an explorative data-collection policy

$$\pi_{\text{explore}}(\mathbf{s}_t) = \mathbf{a}_t = \begin{cases} \mathbf{a} \sim \mathcal{U}(\mathcal{A}) & \text{with probability } p \\ \mathbf{a}_t^* & \text{with probability } 1 - p, \end{cases} \tag{1}$$

where $\mathcal{U}$ is the uniform distribution, $p \in [0, 1]$ is a probability threshold, and $\mathbf{a}_t^*$ is a greedy action. There are multiple ways to instantiate greedy policies [36, 37, 38, 13]. We use the action that minimizes the distance to the goal state $\mathbf{s}_g$ with a velocity v:

$$\mathbf{a}_t^* = v \cdot \frac{(\mathbf{s}_g + \Delta\mathbf{s}_{\text{grasp}}) - \mathbf{s}_t}{\|(\mathbf{s}_g + \Delta\mathbf{s}_{\text{grasp}}) - \mathbf{s}_t\|_2}, \tag{2}$$

where $\Delta\mathbf{s}_{\text{grasp}}$ is the randomized grasp offset to the reference demonstration, and the state is the Cartesian end-effector pose. Additionally, we apply a safety filter to the action (1) to prevent exploration from drifting outside the critical segments, constraining the end-effector position to a sphere with center $\mathbf{s}_{\text{safe}}$ and radius r_{safe}.

This combination of random and optimal actions provides dense coverage of the critical segment while maintaining high success rates. The data collection episodes end either by truncation (timeout-based) or by task completion. The task reward r_t is a sparse reward determined by the distance of state $\mathbf{s}_t$ to the final state of the hardcoded trajectory $\mathbf{s}_g$: $r_t = R_{\text{success}} \cdot \mathbb{1}(\|\mathbf{s}_t - (\mathbf{s}_g + \Delta\mathbf{s}_{\text{grasp}})\|_2 \leq \epsilon)$. Computing the reward state-by-state eliminates the need for a pre-trained vision-based reward classifier [39, 40], which would require additional demonstrations. Instead, we obtain a vision-based classifier as the result of the DRL training in the form of the Q-function.

Contact-rich manipulation requires tight integration of perception and force feedback, particularly during the insertion phase [29]. For this reason, we employ hybrid force-position control [25, 29] during the critical, force-sensitive interactions. Specifically, we apply a desired force along the insertion axis and position-control the remaining degrees of freedom. This enables compliant and force-controlled behavior with precise positioning, and the same is applied during deployment.

3.3 Policy Learning

To minimize data collection effort, we require a sample-efficient learning algorithm. Similar to RLPD [41], we build on top of SAC [42] and employ improvements for sample efficiency. To mitigate Q-function overestimation, we add layer normalization [43] to the critic and use Randomized Ensembled Double Q-Learning [44], where we train N_Q Q-functions and select a random subset S of size $N_{Q,S}$ to compute the TD-targets as

$$y_t = r_t + \gamma \mathbb{E}_{\mathbf{a}' \sim \pi_\theta} \left[\min_{i \in S} Q_{\bar{\phi}_i}(\mathbf{s}_{t+1}, \mathbf{a}') - \alpha \log \pi_\theta(\mathbf{a}' \mid \mathbf{s}_{t+1}) \right]. \tag{3}$$

The Q-functions and policy are trained via the standard SAC objective [42] using y_t. Since our setting does not rely on online learning, we fix the entropy coefficient α rather than adapting it during learning. This setup is a significant algorithmic and architectural simplification compared to online policy optimization.

3.4 Full Task Execution

Interaction with the scene is done using a language-guided segmentation model (SAM3 [8]), which we prompt with commands such as "purple Lego brick", "gray plastic cover with circular grille", and "yellow cardboard salt box" for our tasks. We feed the segmentation masks into an off-the-shelf pose estimation model [9, 45] to obtain 6D pose estimates for the two objects in the scene that we manipulate, unless otherwise specified. Pose estimation is run multiple times to refine the estimates throughout the trajectory (Appendix A.5). The pose estimates serve as the basis for motion planning, which uses simple waypoint following. Our formulation can easily be extended to include trajectory optimization or obstacle avoidance.

All waypoints for motion planning are defined relative to the object's 6D pose and based on the single demonstration. The robot first moves towards the pick-up object until it reaches a specified waypoint. Then, we grasp the object and transport it to the critical segment (also using waypoints). The learned policy is triggered once contact is made, i.e., if $\|\mathbf{F}\|_2 \geq \mathcal{F}_{\text{threshold}}$ [46, 47] (Appendix A.2). Then, the learned policy is executed and terminated based on the learned Q-functions and a threshold λ_{success} to detect successful insertion:

$$Q(\mathbf{s}_t, \pi_\theta(\mathbf{s}_t)) = \frac{1}{N_Q} \sum_{i=1}^{N_Q} Q_{\phi_i}(\mathbf{s}_t, \pi_\theta(\mathbf{s}_t)) > \lambda_{\text{success}}, \tag{4}$$

We additionally use a hysteresis-style detection to avoid false negative cases. Further details and examples for the Q-function-based success classification are in Appendix A.4. After the learned policy terminates, any remaining motions, such as a push-down to complete the insertion, are executed.

4 Experimental Results

4.1 Task descriptions

We evaluate our approach on multiple challenging real-world tasks that require handling deformable components, precise alignment, and contact-rich interaction, as shown in Figure 2. Experimental details are provided in Appendix A.7.

Shelf stocking: This task requires stocking a tightly packed shelf with a yellow salt box. Most objects on the shelf are made of cardboard and may deform during interaction. The policies need to learn to react to those changes, precisely align the salt box, and perform an additional push to fully complete the insertion. *Partial success* is if the object is placed on the shelf, but not inserted.

Lego stacking tasks have recently enjoyed popularity in the robot learning community. The task is to pick up a brick and stack it on another 4×2 brick from an unknown position on the base plate.

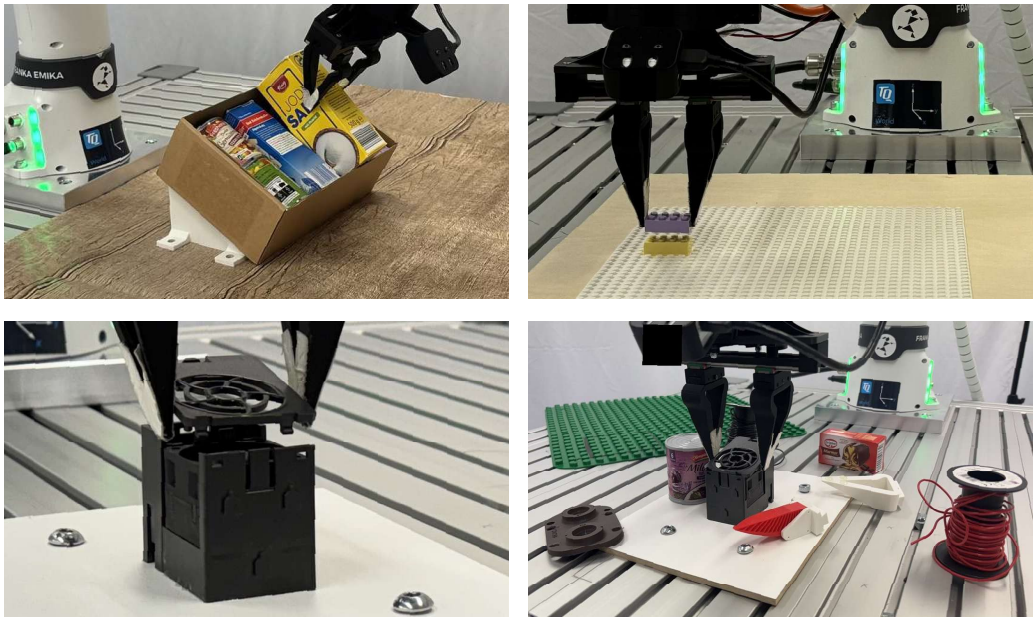


Figure 2: **Tasks.** We evaluate our method on four challenging real world tasks: shelf stocking, Lego stacking, mounting two PBT/PC-molded parts (fan cover), and a difficult version of the fan cover task that we train on scene distractors, similar objects, and changing ground surfaces. For all tasks, we evaluate the policies on the additional out-of-distribution scenarios shown in Appendix A.6.

The task is solved once the two bricks are fully stacked together. *Partial success* is defined as the blocks being placed on top of each other, but not fully inserted.

Fan cover originates from a real-world production line that currently necessitates manual assembly. The assembly requires aligning the studs through intricate, contact-rich motions against the base. The injection-molded PBT/PC are deformable in the process. *Partial success* is defined as the cover being placed on the base, but not fully inserted. **Fan cover (difficult)**: A difficult version of this task introduces additional diversity during data collection and deployment (Figure 2). For both versions of this task, we assume the fan base is fixed and do not estimate it.

4.2 Implementation Details

For our data collection scheme, we set the probability of random actions to $p = 0.8$ and truncate episodes after $10\,\mathrm{s}$ (150 steps). We train our policies for 5 epochs, which takes $1\,\mathrm{h}$ on a GPU workstation. The policy receives a single RGB viewpoint from a wrist-mounted camera, and pose estimation uses additional depth measurements [9, 45]. The images are cropped to the critical task segment (Appendix A.9), embedded using DINOv2 [48], and projected down to 16 dimensions using an MLP before concatenation with the state. The state includes F/T measurements, controller error, Cartesian velocity, and the previous action. During deployment, we use $\lambda_{\text{success}} = 0.93 R_{\text{success}}$ for the Q-function based success classifiers. Further hardware and training details are in Appendix A.8.

Baselines. We benchmark against end-to-end Imitation Learning and DRL. For Imitation, we choose DiTFlow [49] and DP [1]) as task-specific policies and train them from scratch for 50k gradient steps ($5\,\mathrm{h}$). For DiTFlow, we include a version trained on data collected by a novice teleoperator inexperienced with data collection. $\pi_{0.5}$ [2] is our baseline for a foundation model that we finetune for 40k steps on $8 \times$ H100 GPUs ($18\,\mathrm{h}$). For DRL, we use HIL-SERL [12], with 20 initial demonstrations and a human guiding policy learning as per the original implementation. We include a baseline with pure pose estimation and motion planning (PE & MP) without any learning.

Table 1: Data collection

Task	# Episodes (Failures[1]) / # Datapoints / Data collection wall-clock time			
	Teleop	**Teleop (Novice)[2]**	**HIL-SERL**	**Ours**
Shelf stocking	100 (22) / 24k / 2.3 h	100 (18) / 34k / 2.3 h	346 (133) / 27k / 3.2 h	**300 / 27k / 2** h
Lego stacking	100 (30) / 33k / 2.5 h	100 (27) / 42k / 2.6 h	495 (221) / 32k / 3.2 h	**500 / 30k / 2.5** h
Fan Cover	100 (50) / 39k / 2.75 h	—	—	**300 / 33k / 2** h
Fan Cover (hard)	100 (55) / 40k / 2.5 h	—	—	—[3]

[1]Episodes that failed during teleoperation — i.e., where the teleoperator did not successfully complete the task — are indicated in brackets and were excluded from policy training. [2]"Novice" refers to an inexperienced teleoperator who has been given a couple of trials before starting data collection. All other data have been collected by an experienced operator. [3]No additional data collected.

Table 2: Success Rates (Partial Success Rates[1]) [%]

Task	DiTFlow	DiTFlow (Novice)[2]	DP	Finetuned $\pi_{0.5}$	MP & PE[3]	HIL-SERL	Ours
Shelf stocking	38 (96)	34 (80)	44 (100)	42 (92)	98 (98)	10 (24)	**98 (100)**
Lego stacking	60 (88)	22 (48)	44 (92)	6 (80)	78 (100)	6 (22)	**94 (100)**
Fan Cover	30 (88)	—	30 (96)	20 (96)	22 (94)	—	**96 (98)**
Fan Cover (hard)	18 (82)	—	26 (92)	12 (76)	22 (94)	—	**94 (96)**[4]
Average	37 (89)	—	36 (95)	20 (86)	55 (97)	—	**96 (99)**

We roll out each policy for 50 trials. [1]Partial success is when the objects are placed correctly but not fully inserted or assembled. [2]'Novice' refers to a policy trained on data collected by a novice teleoperator. All other data have been collected by an experienced operator. [3]Motion Planning & Pose Estimation. [4]We deployed the same policy as for the simple version of the task without additional data collection.

4.3 Policy Evaluations

Our central claim is that traditional planning combined with data collection and learning for the critical segment leads to high success rates and robust policies. We aim for equal wall-clock time for data collection across all methods (Table 1). During the same wall-clock time, our data collection scheme collects significantly more data at the critical segment. We note that despite being largely autonomous, our data collection scheme required a few interventions when the scene was not reset correctly. Those interventions put significantly less strain on the operator than teleoperation.

While end-to-end methods capture the task semantics, they typically struggle at the critical segment, as shown by high *partial* success rates — indicating correct grasping and coarse placement — but low overall success rates (Table 2). HIL-SERL struggled primarily with undirected gripper-actions during exploration, leading to frequent failures given the long task horizon. We also observe high variance in training outcomes during runs, which is characteristic of online policy optimization [50, 51]. Our sequential pipeline, consisting of MP, PE, learned policy, and policy switching, achieves consistently $\geq 90\%$ SR. Remaining failure cases are due to OOD scenarios and hardware inaccuracies. See Appendix A.10 for an overview of failure cases.

A modular approach enables tailoring policy parameters for each task segment. For instance, during the planned push-down motion, we control the force to a magnitude similar to that in the single kinesthetic demonstration. This leads to overall fewer forces being applied to the objects, which is

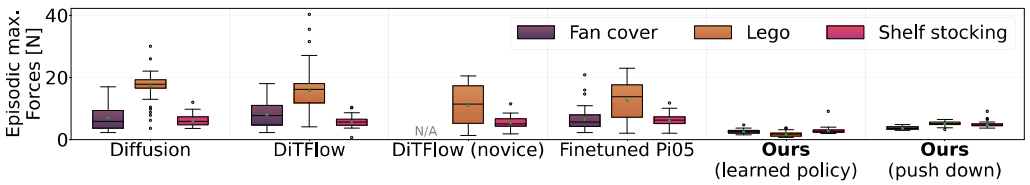


Figure 3: Maximum forces applied during policy rollouts. **Ours** applies significantly less forces and torques to the objects than baselines. In the figure, we divide ours between the learned insertion policy and the planned push-down motion. Torques are shown in Appendix A.12.

Table 3: Success Rates (Partial Success Rates[1]) [%] for out-of-distribution scenarios.

Task	DiTFlow	DP	Finetuned $\pi_{0.5}$	HIL-SERL	Ours
Shelf stocking	20 (40)	25 (45)	10 (65)	0 (5)	**90 (100)**
Lego stacking	10 (10)	10 (25)	0 (0)	0 (0)	**85 (95)**
Fan Cover	0 (30)	0 (20)	0 (50)	—	**90 (95)**
Fan Cover (hard)	0 (50)	5 (50)	5 (50)	—	**95 (95)**[2]

We roll out each policy for 20 trials. [1]Partial success is when the objects are placed correctly but not fully inserted or assembled. [2]We deployed the same policy as for the simple version of the task — without additional data collection for this task.

critical for the sensitive cardboard and PBT/PC molded parts (Figure 3). Applied maximum forces are reduced on average by 49% ($-4.7\,\mathrm{N}$) compared to baselines.

A policy trained on data from a novice operator (DiTFlow (Novice)) achieves lower success rates (Table 2), demonstrating the dependency on demonstration data and motivating automated data collection. We find that novice demonstrations are characterized by multi-modal demonstrations, i.e., the task is demonstrated in different ways, higher speed variance ($\mathrm{CV}_{\text{novice}} = 0.285$ vs. $\mathrm{CV}_{\text{expert}} = 0.185$), and overall longer trajectories ($27\,\mathrm{s}$ vs. $24\,\mathrm{s}$).

4.4 Generalization

We expose the policies to OOD scenarios, including distractor objects, different object configurations, placements, and backgrounds (Appendix A.6). End-to-end methods struggle with OOD scenarios (Table 3). The fan cover (hard) task includes the standard fan cover OOD scenarios during training, and during evaluation additional distractors are introduced. The augmentations during training increased partial success rates to 50%, but fully complete insertions remain few (5%). Substantially more diverse data would likely be required to achieve generalization.

As our policies operate on local camera images, the success rates stay similar to in-distribution evaluations. Remaining failure cases include distractor objects being placed close to the insertion point, taking local visual observations out of distribution (Appendix A.10). We present additional qualitative results for complex scenarios such as dynamically grasping from a human hand, arbitrary object poses, and further randomizations — all without additional data collection — in Appendix A.6. The motion planning can easily be extended to include trajectory optimization and avoidance [52].

4.5 Data Collection Ablations

We analyze key parameters of our autonomous data acquisition scheme. We use a MuJoCo [53] simulated setup of the Lego tasks, train policies on three different seeds, and report the results on 1000 policy rollouts. The results focus on three insights.

Successful policies require dense data coverage of the critical segment (Figure 4 a+b). Policies trained on data with little exploration ($p <= 0.3$) achieve low success rates. Little exploration results in short trajectories that yield little training data (see Figure 4 (a)). Note that $p = 0$ is close to imitation learning data. On the other hand, pure exploration ($p = 1$) leads to low SR and thus unsuccessful policies. In practice, one requires sufficiently long rollouts while maintaining a high success rate, which we observe for $0.6 \leq p \leq 0.85$. The time spent at the critical segment per rollout is critical, necessitating DRL to learn from non-optimal demos.

Scaling of success rate and dataset size with pose estimation uncertainty (Figure 4 c). Increasingly uncertain pose estimation expands the critical segment of the task. In simulation, we require 400 rollouts for $6\,\mathrm{mm}$ and 700 rollouts for $12\,\mathrm{mm}$ uncertainty. This corresponds to $2\,\mathrm{h}$ to $3.5\,\mathrm{h}$ of data collection in the real world. We tend to require more data in simulation than on hardware for the same success rates, possibly because of inconsistent simulation physics during contacts.

Multi-modal sensing for high success rates (Appendix A.11). Combining a wrist-mounted F/T sensor with a single wrist camera achieves $100\,\%$ success in simulation. This is the setup we also

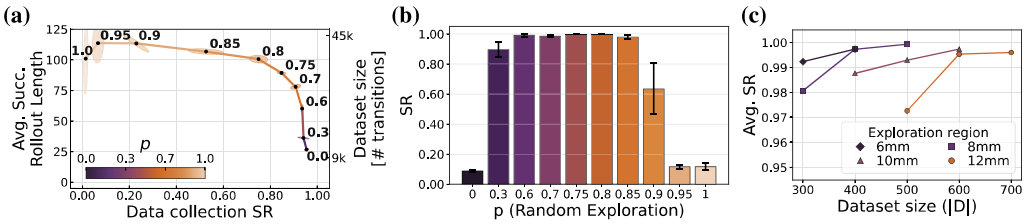


Figure 4: Ablation of data collection parameters. **(a)** Effect of random action sampling rate p on average successful rollout length, dataset size, and success rate during data collection. **(b)** Trained policy success rate for different random exploration rates p during data collection. **(c)** Scaling of SR and dataset size with pose estimation uncertainty.

chose for our hardware experiments. F/T measurements provide useful feedback even when the task state is not fully observable from images alone, e.g., when objects are already in contact.

5 Discussion

We use motion planning for the free-space motion of the task, but this doesn't necessarily have to be the case. Policies for free-space motion could come from other inexpensive, non-robot data sources that can be collected at a large scale, such as human videos [54]. Future work can investigate how to balance, collect, and integrate different data sources for modular policies.

Sequential policies naturally give rise to task-segment specific priors, which increase performance significantly for difficult tasks, as we demonstrate with reduced contact forces compared to end-to-end methods. A key question is the design of switching mechanisms, and Q-function-based switching worked well for our tasks. VLM-based task segment switching for semantically-rich tasks is an interesting direction for future research.

All vision-based policies are sensitive to visual OOD shifts. However, locally operating policies narrow down the set of OOD scenarios and remain robust under scene-level distractors. In contrast to end-to-end methods, visual randomization can remain tractable for local policies by leveraging simulation and sim-and-real cotraining [55] — potentially without additional real robot data.

6 Conclusion

Robot data is expensive, and we should consider how to collect and use it efficiently. While clearly more robot data is needed overall, we argue for focusing collection on tasks and task-segments that are otherwise difficult to solve. By adopting a modular approach with targeted data collection, we maintain high success rates for OOD scenarios, demonstrating that targeted reliance on robot data and structured task decomposition are key enablers of robust generalizable manipulation while keeping data acquisition efforts at bay.

7 Limitations

Task diversity. We demonstrate high success rates on tasks with sequential separation of free-space and contact-rich motion. For other tasks, such as t-shirt folding, sequential boundaries are harder to define — motivating future work in policy switching mechanisms determined by intelligent LLMs.
Hardware Precision. Our method benefits from hardware that precisely reaches target poses for waypoint tracking and policy switching, leading to high success rates (Appendix A.10). End-to-end approaches can absorb hardware inaccuracies through large-scale data collection.
Pose Estimation. Exotic objects, reflective surfaces, or partial occlusions challenge off-the-shelf pose estimation models. However, these models can be fine-tuned relatively easily using only images and CAD data, without requiring expensive robot demonstrations.

References

[1] 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*, 44(10-11):1684–1704, 2025.

[2] K. Black, N. Brown, J. Darpinian, K. Dhabalia, D. Driess, A. Esmail, M. R. Equi, C. Finn, N. Fusai, M. Y. Galliker, D. Ghosh, L. Groom, K. Hausman, b. ichter, S. Jakubczak, T. Jones, L. Ke, D. LeBlanc, S. Levine, A. Li-Bell, M. Mothukuri, S. Nair, K. Pertsch, A. Z. Ren, L. X. Shi, L. Smith, J. T. Springenberg, K. Stachowicz, J. Tanner, Q. Vuong, H. Walke, A. Walling, H. Wang, L. Yu, and U. Zhilinsky. $\pi_{0.5}$: a vision-language-action model with open-world generalization. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 17–40. PMLR, 2025.

[3] Y. Li, H. Jiang, J. Xia, H. Zhang, J. Du, Y. Zhou, J. Zeng, C. Hao, J. Ren, Q. Yu, et al. Forcevla2: Unleashing hybrid force-position control with force awareness for contact-rich manipulation. *arXiv preprint arXiv:2603.15169*, 2026.

[4] A. O'Neill, A. Rehman, A. Maddukuri, A. Gupta, A. Padalkar, A. Lee, A. Pooley, A. Gupta, A. Mandlekar, A. Jain, et al. Open x-embodiment: Robotic learning datasets and rt-x models: Open x-embodiment collaboration 0. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 6892–6903, 2024.

[5] A. Khazatsky, K. Pertsch, S. Nair, A. Balakrishna, S. Dasari, S. Karamcheti, S. Nasiriany, M. K. Srirama, L. Y. Chen, K. Ellis, et al. DROID: A large-scale in-the-wild robot manipulation dataset. In *Robotics: Science and Systems*, 2024.

[6] T. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *Robotics: Science and Systems (RSS)*, 2023.

[7] NVIDIA et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[8] N. Carion, L. Gustafson, Y.-T. Hu, S. Debnath, R. Hu, D. Suris, C. Ryali, K. V. Alwala, H. Khedr, A. Huang, et al. Sam 3: Segment anything with concepts. *arXiv preprint arXiv:2511.16719*, 2025.

[9] B. Wen, W. Yang, J. Kautz, and S. Birchfield. Foundationpose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17868–17879, 2024.

[10] H. Li, Y. Cui, and D. Sadigh. How to train your robots? the impact of demonstration modality on imitation learning. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1113–1120, 2025.

[11] S. Belkhale, Y. Cui, and D. Sadigh. Data quality in imitation learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 80375–80395. Curran Associates, Inc., 2023.

[12] J. Luo, C. Xu, J. Wu, and S. Levine. Precise and dexterous robotic manipulation via human-in-the-loop reinforcement learning. *Science Robotics*, 10(105), 2025.

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

[14] Y. Tian, J. Xu, Y. Li, J. Luo, S. Sueda, H. Li, K. D. D. Willis, and W. Matusik. Assemble them all: Physics-based planning for generalizable assembly by disassembly. *ACM Transactions on Graphics*, 41(6):1–11, 2022.

[15] Y. Tian, K. D. Willis, B. Al Omari, J. Luo, P. Ma, Y. Li, F. Javid, E. Gu, J. Jacob, S. Sueda, et al. Asap: Automated sequence planning for complex robotic assembly with physical feasibility. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4380–4386. IEEE, 2024.

[16] L. Ankile, A. Simeonov, I. Shenfeld, and P. Agrawal. Juicer: Data-efficient imitation learning for robotic assembly. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5096–5103. IEEE, 2024.

[17] B. Fu, S. K. Leong, X. Lian, and X. Ji. 6d robotic assembly based on rgb-only object pose estimation. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4736–4742. IEEE, 2022.

[18] A. Morgan, B. Wen, J. Liang, A. Boularias, A. Dollar, and K. Bekris. Vision-driven compliant manipulation for reliable; high-precision assembly tasks. In *Robotics: Science and Systems XVII*, RSS2021. Robotics: Science and Systems Foundation, 2021.

[19] B. Fu, S. K. Leong, Y. Di, G. Wang, J. Tang, F. Tombari, and X. Ji. Lanpose: Language-instructed 6d object pose estimation for robotic assembly. In *European Conference on Computer Vision*, pages 43–59. Springer, 2024.

[20] B. Tang, M. A. Lin, I. A. Akinola, A. Handa, G. S. Sukhatme, F. Ramos, D. Fox, and Y. S. Narang. IndustReal: Transferring Contact-Rich Assembly Tasks from Simulation to Reality. In *Proceedings of Robotics: Science and Systems*, 2023.

[21] G. Schoettler, A. Nair, J. A. Ojea, S. Levine, and E. Solowjow. Meta-reinforcement learning for robotic industrial insertion tasks. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 9728–9735. IEEE, 2020.

[22] T. Bi and R. D'Andrea. Sample-efficient learning to solve a real-world labyrinth game using data-augmented model-based reinforcement learning. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7455–7460. IEEE, 2024.

[23] L. Ankile, A. Simeonov, I. Shenfeld, M. Torne, and P. Agrawal. From imitation to refinement-residual rl for precise assembly. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 01–08. IEEE, 2025.

[24] A. Goyal, V. Blukis, J. Xu, Y. Guo, Y.-W. Chao, and D. Fox. RVT-2: Learning Precise Manipulation from Few Demonstrations. In *Proceedings of Robotics: Science and Systems*, 2024.

[25] H. L. Brown, G. Hollinger, and S. Lee. Learning hybrid-control policies for high-precision in-contact manipulation under uncertainty. *arXiv preprint arXiv:2604.19677*, 2026.

[26] F. Shao, S. Endo, S. Hirche, and F. Ficuciello. Interactive force-impedance control. *IEEE Robotics and Automation Letters*, 11(5):6488–6495, 2026.

[27] J. Liang, X. Cheng, and O. Kroemer. Learning preconditions of hybrid force-velocity controllers for contact-rich manipulation. In *Proceedings of The 6th Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pages 679–689. PMLR, 2023.

[28] W. Liu, J. Wang, Y. Wang, W. Wang, and C. Lu. Forcemimic: Force-centric imitation learning with force-motion capture system for contact-rich manipulation. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1105–1112. IEEE, 2025.

[29] H. Fang, S. Tang, M. Mei, H. Qin, Z. He, J. Chen, Y. Feng, C. Wang, W. Liu, Z. He, et al. Force policy: Learning hybrid force-position control policy under interaction frame for contact-rich manipulation. *arXiv preprint arXiv:2602.22088*, 2026.

[30] M. A. Lee, C. Florensa, J. Tremblay, N. Ratliff, A. Garg, F. Ramos, and D. Fox. Guided uncertainty-aware policy optimization: Combining learning and model-based strategies for sample-efficient policy learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7505–7512, 2020.

[31] H. Fang, S. Tang, M. Mei, H. Qin, Z. He, J. Chen, Y. Feng, C. Wang, W. Liu, Z. He, et al. Force policy: Learning hybrid force-position control policy under interaction frame for contact-rich manipulation. *arXiv preprint arXiv:2602.22088*, 2026.

[32] L. Wang, J. Zhao, Y. Du, E. H. Adelson, and R. Tedrake. PoCo: Policy composition from and for heterogeneous robot learning. In *Robotics: Science and Systems*, 2024.

[33] H. Chen, J. Xu, H. Chen, K. Hong, B. Huang, C. Liu, J. Mao, Y. Li, Y. Du, and K. Driggs-Campbell. Multi-modal manipulation via multi-modal policy consensus. *arXiv preprint arXiv:2509.23468*, 2025.

[34] Y. Wang, Y. Zhang, M. Huo, R. Tian, X. Zhang, Y. Xie, C. Xu, P. Ji, W. Zhan, M. Ding, et al. Sparse diffusion policy: A sparse, reusable, and flexible policy for robot learning. 2024.

[35] R. Römer, Y. Zhang, Y. Li, and A. P. Schoellig. Clare: Continual learning for vision-language-action models via autonomous adapter routing and expansion. *IEEE Robotics and Automation Letters*, 2026.

[36] A. Mandlekar, S. Nasiriany, B. Wen, I. Akinola, Y. Narang, L. Fan, Y. Zhu, and D. Fox. Mimicgen: A data generation system for scalable robot learning using human demonstrations. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 1820–1864. PMLR, 2023.

[37] A. Mandlekar, C. R. Garrett, D. Xu, and D. Fox. Human-in-the-loop task and motion planning for imitation learning. In *Proceedings of The 7th Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pages 3030–3060. PMLR, 2023.

[38] C.-H. Yeh, T.-S. Nan, R. Vuorio, W. Hung, H. Y. Wu, S.-H. Sun, and P.-C. Hsieh. Action-constrained imitation learning. In *Forty-second International Conference on Machine Learning*, 2025.

[39] T. Lee, A. Wagenmaker, K. Pertsch, P. Liang, S. Levine, and C. Finn. Roboreward: General-purpose vision-language reward models for robotics. *arXiv preprint arXiv:2601.00675*, 2026.

[40] A. Liang, Y. Korkmaz, J. Zhang, M. Hwang, A. Anwar, S. Kaushik, A. Shah, A. S. Huang, L. Zettlemoyer, D. Fox, et al. Robometer: Scaling general-purpose robotic reward models via trajectory comparisons. *arXiv preprint arXiv:2603.02115*, 2026.

[41] P. J. Ball, L. Smith, I. Kostrikov, and S. Levine. Efficient online reinforcement learning with offline data. In *International Conference on Machine Learning*, pages 1577–1594. PMLR, 2023.

[42] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

[43] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[44] X. Chen, C. Wang, Z. Zhou, and K. W. Ross. Randomized ensembled double q-learning: Learning fast without a model. In *International Conference on Learning Representations*, 2021.

[45] T. Collins and A. Bartoli. Infinitesimal plane-based pose estimation. *International Journal of Computer Vision*, 109(3):252–286, 2014.

[46] J. Stranghöner, P. Hartmann, M. Braun, S. Wrede, and K. Neumann. Share-rl: Structured, interactive reinforcement learning for contact-rich industrial assembly tasks. *arXiv preprint arXiv:2509.13949*, 2025.

[47] X. Zhang, S. Jin, C. Wang, X. Zhu, and M. Tomizuka. Learning insertion primitives with discrete-continuous hybrid action space for robotic assembly tasks. In *2022 International conference on robotics and automation (ICRA)*, pages 9881–9887. IEEE, 2022.

[48] M. Caron, H. Touvron, I. Misra, H. Jegou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9630–9640, 2021.

[49] S. Dasari, O. Mees, S. Zhao, M. K. Srirama, and S. Levine. The ingredients for robotic diffusion transformers. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15617–15625. IEEE, 2025.

[50] Y. Chen, S. Tian, S. Liu, Y. Zhou, H. Li, and D. Zhao. ConRFT: A Reinforced Fine-tuning Method for VLA Models via Consistency Policy. In *Proceedings of Robotics: Science and Systems*, 2025.

[51] E. Nikishin, M. Schwarzer, P. D'Oro, P.-L. Bacon, and A. Courville. The primacy bias in deep reinforcement learning. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 16828–16847. PMLR, 2022.

[52] D. Coleman, I. Sucan, S. Chitta, and N. Correll. Reducing the barrier to entry of complex robotic software: a moveit! case study. *arXiv preprint arXiv:1404.3785*, 2014.

[53] E. Todorov, T. Erez, and Y. Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033. IEEE, 2012.

[54] M. Lepert, J. Fang, and J. Bohg. Phantom: Training robots without robots using only human videos. In *Proceedings of The 9th Conference on Robot Learning*, volume 305 of *Proceedings of Machine Learning Research*, pages 4545–4565. PMLR, 2025.

[55] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev, S. Reed, K. Goldberg, A. Mandlekar, L. Fan, and Y. Zhu. Sim-and-real co-training: A simple recipe for vision-based robotic manipulation. In *Proceedings of Robotics: Science and Systems (RSS)*, 2025.

[56] R. Cadene, S. Alibert, A. Soare, Q. Gallouedec, A. Zouitine, S. Palma, P. Kooijmans, M. Aractingi, M. Shukor, D. Aubakirova, M. Russi, F. Capuano, C. Pascal, J. Choghari, J. Moss, and T. Wolf. Lerobot: State-of-the-art machine learning for real-world robotics in pytorch. https://github.com/huggingface/lerobot, 2024.

[57] D. S. J. Pro, O. Hausdörfer, R. Römer, M. Dösch, M. Schuck, and A. P. Schoellig. Crisp-compliant ros2 controllers for learning-based manipulation policies and teleoperation. *IEEE Robotics and Automation Practice*, 2026.

A Appendix

A.1 Data collection setups

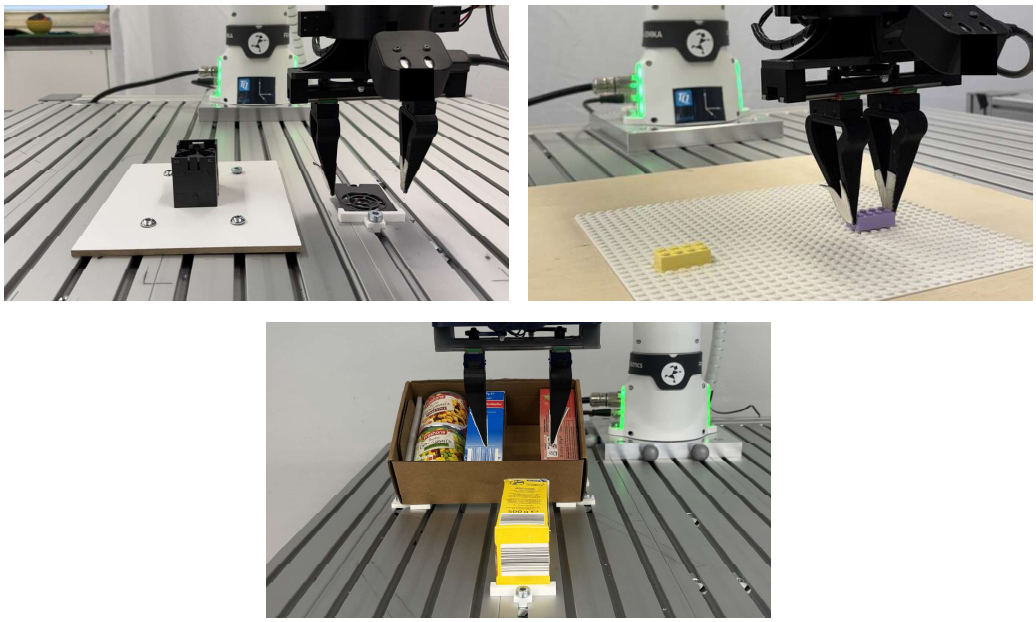


Figure 5: Data collection setups for our different scenes. Our setup requires a fixed scene for data collection. For Lego, we use the base plate to define the layout. For the other tasks, we 3D print small fixtures to keep the part in place.

A.2 Policy Switching

We can choose different mechanisms to trigger the learned policy. Distance-based thresholds depend on the measurements and noise of the depth sensor and can be unreliable for reflective objects. Thresholds based on the uncertainty of the pose estimation [30] suffer the same issue, and additionally require calibration of the uncertainty for different objects, as the uncertainty depends on the object and deployment conditions. For our chosen tasks, we find that a F/T-based threshold as used in [29, 46, 47] is most reliable: $\|\mathbf{F}\|_2 \geq \mathcal{F}_{\text{threshold}}$. Thus, we start the critical segment once contact is made.

For determining the success of the learned policy for the critical segment, we use switching based on the learned Q-function as described in the main text.

A.3 Safety Filter

We constrain the end-effector position to a sphere with center $\mathbf{s}_{\text{safe}}$ and radius r_{safe} during exploration, yielding the finally applied action:

$$\mathbf{a}'_t = \begin{cases} v \cdot \dfrac{\mathbf{s}_{\text{safe}} - \mathbf{s}_t}{\|\mathbf{s}_{\text{safe}} - \mathbf{s}_t\|_2} & \text{if } \|\mathbf{s}_{\text{safe}} - \mathbf{s}_t\|_2 > r_{safe} \\ \mathbf{a}_t & \text{otherwise.} \end{cases} \tag{5}$$

This formulation redirects the applied action toward the safe region whenever exploration drives the end-effector outside the defined sphere. This promotes dense exploration within the critical segment.

A.4 Success Prediction

The task-completion signal stems from the success classifier, which is the Q-function. The Q-function predicts what state-action pair results in the sparse success reward. Therefore, we use a threshold λ_{success} to detect successful insertion:

$$Q(\mathbf{s}_t, \pi_\theta(\mathbf{s}_t)) > \lambda_{\text{success}} \tag{6}$$

where we average the Q-values across the ensemble of N_Q Q functions:

$$Q(\mathbf{s}, \mathbf{a}) = \frac{1}{N_Q} \sum_{i=1}^{N_Q} Q_{\theta_i}(\mathbf{s}, \mathbf{a}). \tag{7}$$

In practice λ_{success} needs to be tuned for deployment. If λ_{success} is set too low the policy stops the insertion prematurely, and set too high it fails to identify successes. To avoid the first case (false positive), we choose λ_{success} high. To avoid the latter case (false negative), we additionally use a Hysteresis-style maximum detector with two thresholds λ_{high} and λ_{low}:

$$\mathcal{H}(t) := \exists t' < t : Q(\mathbf{s}_{t'}, \pi_\theta(\mathbf{s}_{t'})) > \lambda_{\text{high}} \ \wedge \ Q(\mathbf{s}_t, \pi_\theta(\mathbf{s}_t)) < \lambda_{\text{low}}, \tag{8}$$

where we use $\lambda_{\text{high}} = 0.8 R_{\text{Success}}$ and $\lambda_{\text{low}} = 0.6 R_{\text{Success}}$, with R_{Success} the sparse success reward. This term works, since we use a sparse reward setting and the Q-function decreases sharply after reaching its maximum value at the insertion point. Overall, we get the following classifier:

$$\text{success} = \begin{cases} 1 & \text{if } Q(\mathbf{s}_t, \pi_\theta(\mathbf{s}_t)) > \lambda_{\text{success}} \\ 1 & \text{if } \mathcal{H}(t) \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

Note that this approach has a significant advantage over other methods that require a separately trained reward classifier prior to policy training. Instead of manually recording successful and failed episodes before training starts, our success classifier is a natural byproduct of policy training.

Results. We show the resulting Q-value predictions $Q_\phi(\mathbf{s}_t, \pi_\theta(\mathbf{s}_t))$ as a function of policy timesteps for truncated and successful trajectories in the training data in Figure 6. The separation between successful and truncated trajectories is clearly visible, motivating our method of success classification.

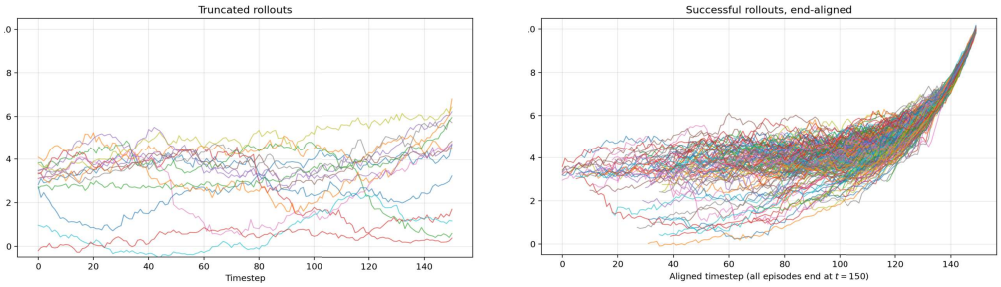


Figure 6: Q-function predictions for truncated and successful rollouts. Successful episodes are aligned such that successful termination occurs at timestep t = 150. The maximum Q-Value is determined by the sparse reward $R_{\text{success}} = 10$.

A.5 Evaluation of Pose Estimation

We evaluate the pose estimation (FoundationPose [9]) quantitatively on the fan cover task. For the evaluation protocol, we placed the fan cover at a fixed position in the scene. We then run pose estimation from a distance of $\approx 30\,\text{cm}$ (coarse) and $\approx 10\,\text{cm}$ (fine) each 50 times from different end-effector positions. Quantitative results are reported in Figure 7 and qualitative results in Figure 8. For the fine estimates, all estimates fall within a range of $\pm 1\,\text{mm}$ and $\pm 3.3°$.

Critical segment: The critical segment is determined by the uncertainty in the pose estimation — effectively, we correct for this uncertainty using robot data. In practice, we choose the critical segment to be larger than the expected uncertainty.

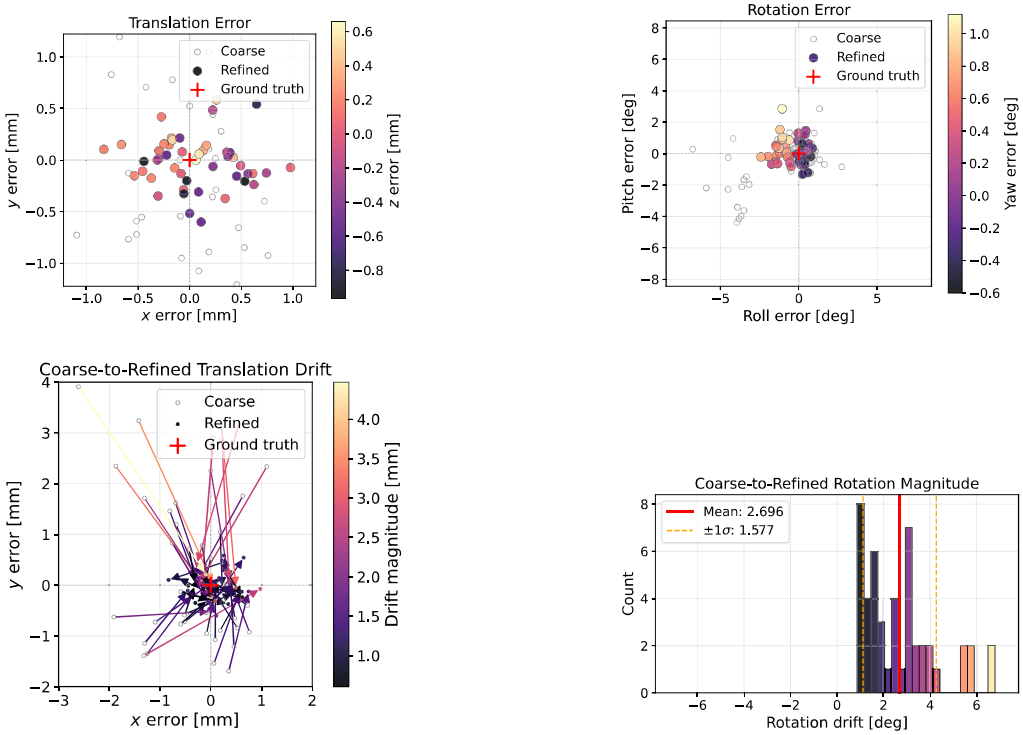


Figure 7: Quantitative pose estimation evaluation. **Top row**: Pose estimation distribution for coarse and fine estimates. **Bottom row**: Correction of pose errors between coarse and fine estimates for positions and orientations.

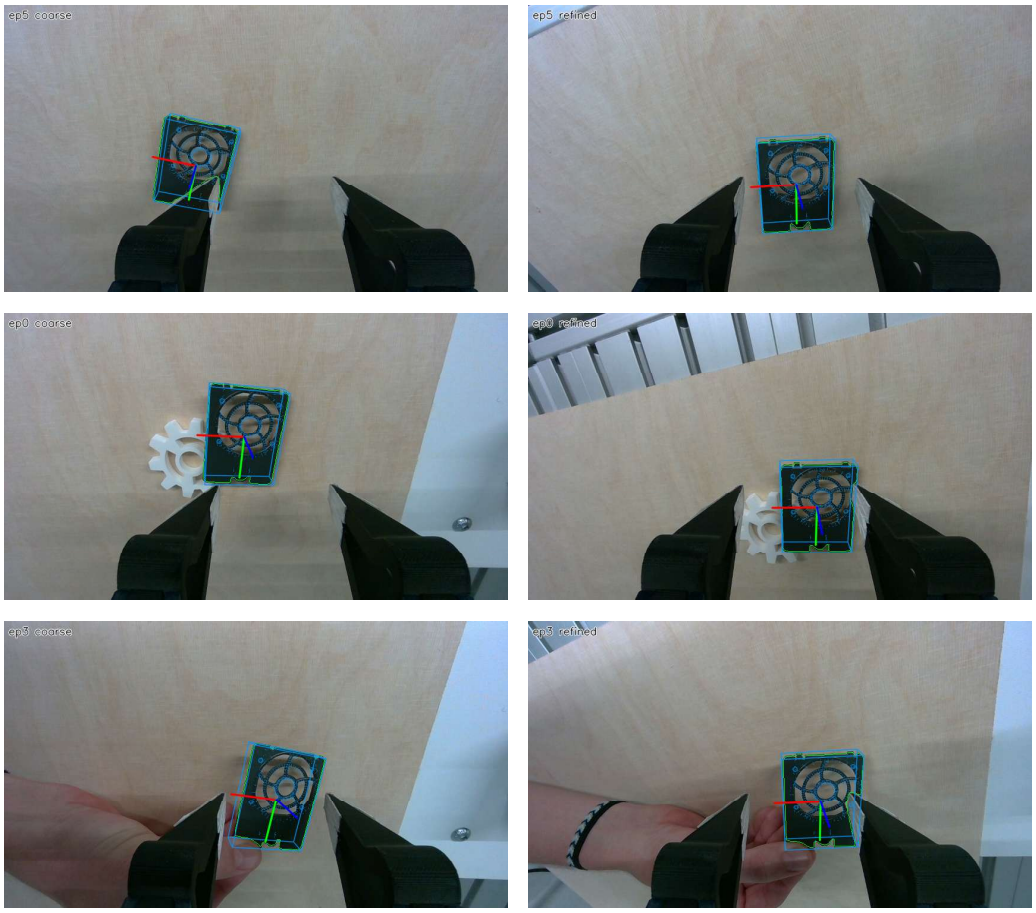


Figure 8: Qualitative pose estimation results including partial occlusions and arbitrary poses. **Left**: Coarse estimates from larger distance. **Right:** Fine estimates from closer distance.

A.6 Out-of-distribution (OOD) evaluations

We show the out-of-distribution settings for quantitative evaluations in Figure 9, and additional qualitative scenarios only for our method in Figure 10.

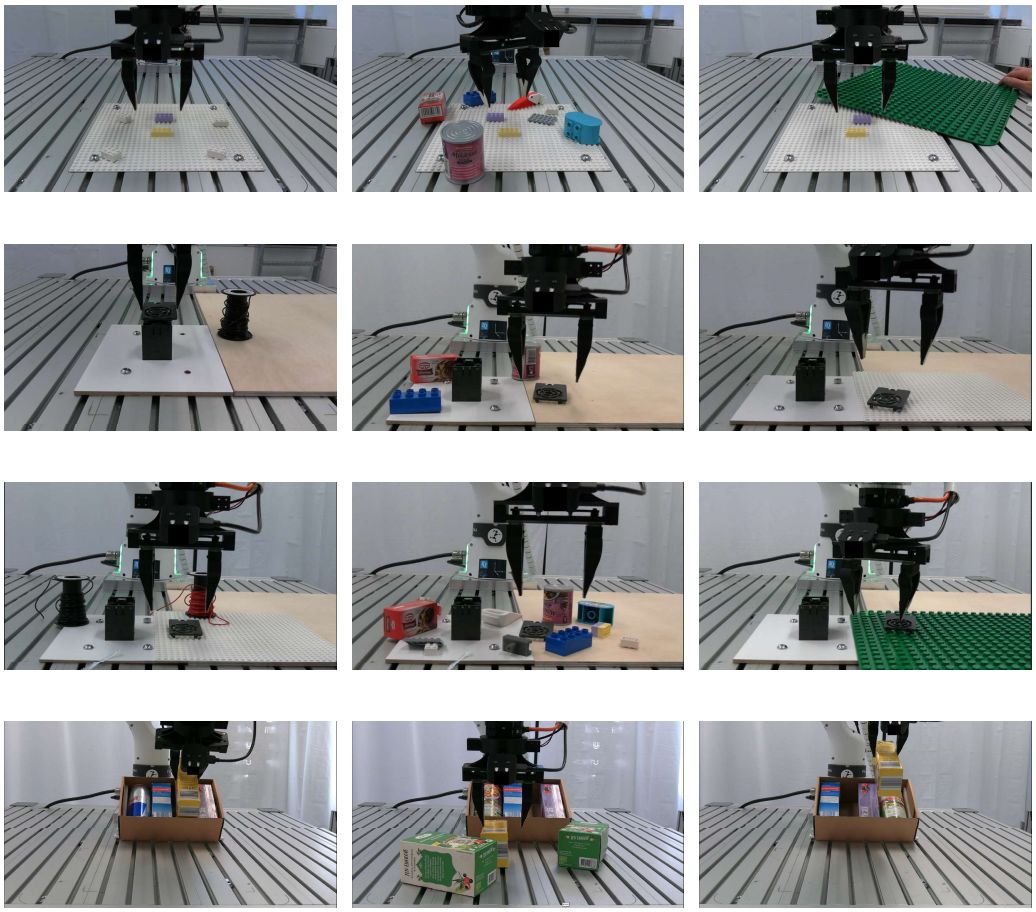


Figure 9: Overview of our out-of-distribution settings used for the quantitative evaluations. For each task, we include evaluations with slight scene distractors, severe distractors, and workspace color changes. **Tasks from top row to bottom**: Lego Stacking, fan cover, fan cover (difficult), shelf stocking. The out-of-distribution settings for the *fan cover* tasks were included in the data collection (in-distribution) for the *fan cover (difficult)* task to test how well training on distractors generalizes.

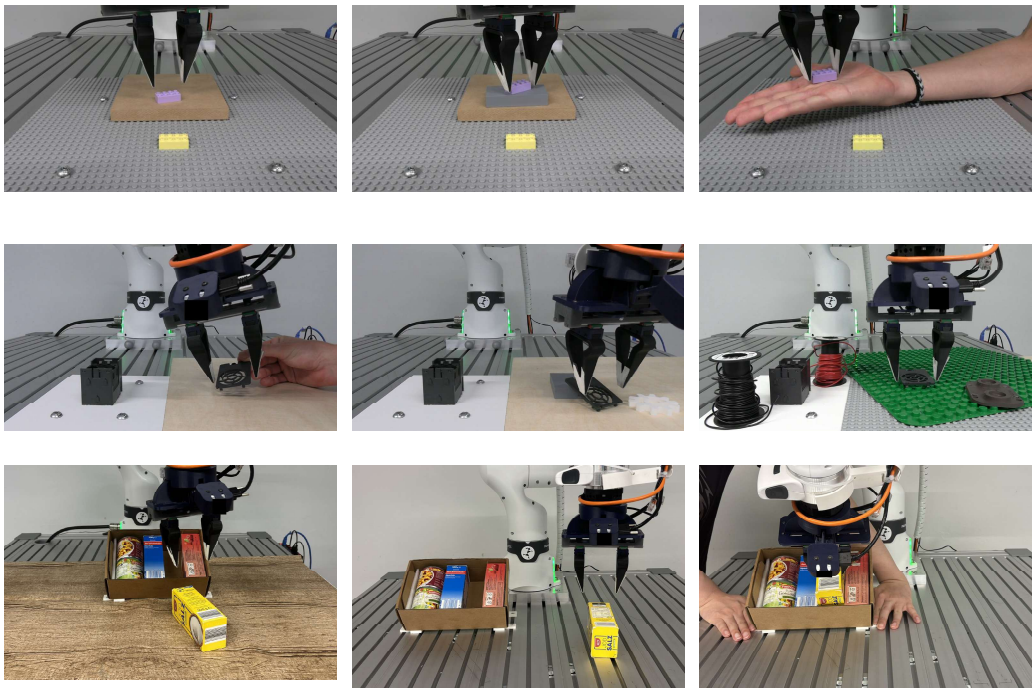


Figure 10: Further qualitative evaluations only performed with method. We place the parts at different heights, perform dynamic grasping from a human hand, and fully randomize the object poses in 6D – all without additional data collection compared to the in-distribution experiments.

A.7 Experimental Details

For all in-distribution evaluations, we place the base object at a fixed position in the workspace. For our method, we still use pose estimation to estimate the base object to demonstrate the full pipeline, except for the fan cover task, where we assume the base is fixed. The inserted object is randomized for every rollout in a given region and is always estimated for our method. For the quantitative out-of-distribution evaluations, the same applies. Here, the positions of the distractor objects are randomized for each rollout. All learned policies are actuated in four dimensions (three translations and one rotation). This made teleoperation for precise tasks significantly easier, and for fair comparison, we applied the same procedure to all baselines and our method. However, for our method, we also show full 6D manipulation trajectories in the additional out-of-distribution evaluations.

For all baselines, we use their LeRobot implementations [56]. The data used for imitation learning is filtered for zero-action frames. We tried to replicate HIL-SERL's success rates for our tasks, which was difficult. We simplified the shelf stacking task slightly by assuming the yellow salt package was pregrasped.

A.8 Training parameters

Training and architectural details for our method are provided in Table 4.

Training. We train our policies, DP, and DiTFlow on a local workstation with an NVIDIA RTX5090 GPU. Our policy trains roughly $1\,\mathrm{h}$, while the others train $5\,\mathrm{h}$. $\pi_{0.5}$ is full-finetuned on two cluster nodes with four H100 GPUs each for $18\,\mathrm{h}$. The equivalent cost for a commercial cloud compute provider would roughly be $200 as at the time of writing.

Hardware. Our hardware is a FRANKA RESEARCH 3. The force-torque sensor we use is the BOTA-DENS-IND2-B4. The wrist-mounted camera is an Intel Realsense D405.

Parameter	Value	Explanation
p	0.8	Probability of choosing a random action (during data collection).
L_{trunc}	150	Episode truncation length (steps).
R_{success}	10	Task completion reward.
Batch size	512	Samples per optimization step.
Q learning rate	5×10^{-4}	Adam step size for critic ensemble.
Policy learning rate	3×10^{-4}	Adam step size for actor.
α	1×10^{-3}	Fixed entropy regularization weight.
γ	0.97	Reward discount factor.
τ	0.999	Polyak averaging coefficient.
UTD	5	Update-to-data ratio, equals number of epochs.
N_Q	10	Critic ensemble size.
$N_{Q,S}$	2	Critic ensemble subset size.
Actor	MLP	Two-layers, hidden size 128, ReLU.
Q-functions	MLP	Ensemble of $N_Q = 10$, two-layers hidden size 128, LayerNorm + ReLU.
f_{lowlevel}	1000Hz	CRISP controllers [57] for direct torque-control.
f_{policy}	15Hz	Control frequency of the DRL policy.

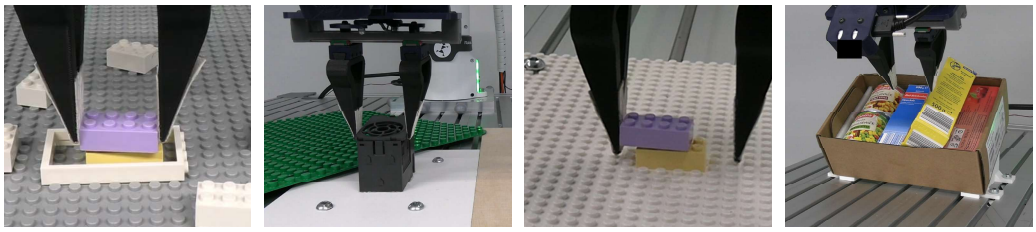
Table 4: Data collection and training hyperparameters.

A.9 Camera Cropping

We show the visual inputs received by our policies in Figure 11. The used image resolution is 224×224.

Figure 11: Visual inputs used for our policies. Policies for our method receive a single viewpoint from a wrist-mounted camera as input that is cropped to the critical insertion region of the task.

A.10 Failure cases



(a) (b) (c) (d)

Figure 12: Policy failure cases for our method that partially concern baselines as well. **(a) + (b)** Our policy fails if distractors are placed very close to the object within the cropped region of the wrist image taking the policy OOD. **(b)** Hardware limitations, such as objects sticking to the gripper, play a role when success rates reach 100%. **(c)** In the one failure case of our method for the shelf insertion, the policy went OOD during insertion, possibly because objects in the shelf moved.

A.11 Data Collection Ablations (Sensor Modalities)

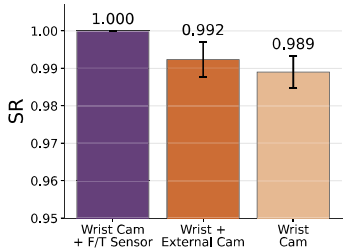


Figure 13: Policy success rate for different combinations of sensing modalities.

A.12 Episodic maximum torques

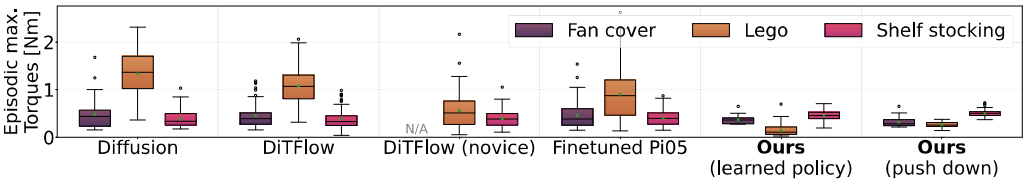


Figure 14: Maximum torques applied during policy rollouts. **Ours** applies significantly less torques to the objects than baselines. In the figure, we divide ours between the learned insertion policy and the planned push-down motion.