

Long-Horizon Mobile Manipulation in Isaac Sim

A Three-Layer Hierarchical Architecture for Household Item Fetching Using
TIAGo, Nav2, ACT, and Residual PPO

Samson Oseiwe Ajadalu (1012365691)
Rajesh Yadav (1012893326)

MIE1076 — AI Applications in Robotics II

April 2026

Repository: <https://github.com/SamsonAjadalu/tiago-nav2-act-fetch>



Contents

	Abstract	3
1	Introduction	3
	1.1 Motivation and Problem Statement	3
	1.2 Contributions and Scope	3
2	Background and Related Work	4
	2.1 Long-Horizon Mobile Manipulation	4
	2.2 Navigation with Nav2	4
2.2.1	Localization in Indoor Mobile Robotics	4
2.2.2	Path and Motion Planning for Mobile Manipulation	5
	2.3 Visuomotor Manipulation with ACT and LeRobot	5
2.3.1	Image Processing and Visual Perception for Manipulation	5
2.3.2	Pose and Grasp Estimation / Visuomotor Grasping	5
	2.4 Residual Reinforcement Learning	5
3	System Overview	6
	3.1 Task Definition	6
	3.2 Three-Layer Architecture	6
	3.3 Key Architectural Decisions	6
4	Navigation Stack	6
	4.1 Design	6
	4.2 Final Working Configuration	8
5	Manipulation Baseline with ACT	8
	5.1 Why ACT	8
	5.2 Observation, Action, and Inference Pipeline	9
	5.3 ACT Architecture and Objective	9
	5.4 Final Deployment Configuration	9
6	Data Collection, Training, and Fine-Tuning	10
	6.1 VR Teleoperation and Recorded Data	10
	6.2 Dataset Curation and Conversion	11
	6.3 Two-Stage ACT Training	11
7	Apartment-Scene Adaptation and Domain Shift	11
	7.1 Transfer Failure	12
	7.2 Diagnosis	12
	7.3 Adaptation Strategy	12
8	Residual PPO on Frozen ACT	13
	8.1 Motivation	13
	8.2 Residual RL Formulation	13
	8.3 Environment and Observation/Action Design	13
	8.4 Two-Phase Reward Logic and Residual Gating	14
	8.5 Residual PPO Training Configuration	14
9	Experimental Setup and Evaluation Protocol	15
	9.1 Experimental Environment	15
	9.2 Integrated Demo Scenario	15
	9.3 Evaluation Protocol	15
10	Results and Discussion	16
	10.1 Navigation Results	17
	10.2 ACT Training and Simple-Scene Performance	17

	10.3	Apartment-Scene Adaptation Results	17
	10.4	Residual PPO Results	18
	10.5	Discussion	19
11		Limitations	19
	11.1	Simulation-Only Evaluation	19
	11.2	Incomplete Quantitative Coverage	19
	11.3	Operational Constraints	19
12		Conclusion and Future Work	20
	12.1	Conclusion	20
	12.2	Future Work	20
		References	21
Appendix A		Additional Figures and Tables	23
		Appendix A.1 Additional Tables	23
Appendix B		Residual PPO Reward Definition	23

Abstract

This report presents a long-horizon mobile manipulation system for household item fetching with the PAL Robotics TIAGo mobile manipulator in NVIDIA Isaac Sim. The task couples room-scale navigation, local visuomotor approach, grasp execution, and return-to-home behaviour in a single episode, making it difficult to solve reliably with either pure navigation or pure manipulation methods alone. The proposed solution uses a three-layer hierarchy: ROS 2 Nav2 provides global transit to a manipulation pose, an Action Chunking Transformer (ACT) policy trained through imitation learning performs the local approach and grasp, and residual Proximal Policy Optimization (PPO) is investigated as a corrective layer over the frozen ACT policy.

Demonstrations were collected through a VR teleoperation pipeline using a Meta Quest 3S, SteamVR/ALVR, OpenXR, and Lula inverse kinematics in Isaac Sim. After cleaning and conversion to the LeRobot format, Stage 1 ACT training used 79 simple-scene episodes and Stage 2 resumed training on a combined 226-episode corpus with apartment-scene upweighting, reaching 160,000 training steps. Under the same 30-trial success-rate protocol, the simple-scene Stage 1 checkpoint achieved 72%, the simple-scene Stage 2 checkpoint achieved 85%, direct apartment-scene transfer from the simple-scene Stage 1 checkpoint achieved 27%, and Stage 2 mixed-dataset training recovered apartment-scene success to 83%. Validated integrated runs indicate that the complete navigate-grasp-return cycle can be executed in simulation.

The residual PPO pipeline was fully implemented in Isaac Lab. Under the same 30-trial success-rate protocol, residual PPO improved success from 85% to 93% in the simple scene and from 83% to 91% in the apartment scene over the corresponding frozen Stage 2 ACT baselines. The historical reference PPO run further improved mean episode return from approximately -531 to a peak near -209 with episode length remaining fixed at the horizon, providing supporting evidence rather than the primary quantitative result. Together, these matched success-rate improvements support PPO as a promising refinement layer.

1 Introduction

1.1 Motivation and Problem Statement

Household fetch tasks are a representative example of long-horizon mobile manipulation. A service robot asked to retrieve an object from a table must first move through the environment, then reposition itself accurately relative to the object, then execute a grasp, and finally return with the object. Each stage has different spatial and temporal requirements: navigation depends on map-scale planning and localisation, while grasping depends on local perception, fine motion, and contact-sensitive execution.

This project addresses the following task: TIAGo starts from a designated home pose in a static indoor scene, navigates to a workspace pose near the target, executes a grasp, and returns to the home pose with the object. The working hypothesis of the report is that the task is best solved by hierarchical decomposition rather than a single monolithic controller. In particular, global transit is treated as a classical planning problem, while local object approach and grasping are treated as a learned visuomotor control problem. Residual reinforcement learning is then used as a refinement mechanism during the final stage of grasp execution.

1.2 Contributions and Scope

The report makes four main contributions:

- It presents an integrated Nav2 + ACT mobile manipulation pipeline on TIAGo in Isaac Sim, and validated integrated runs indicate that the full navigate-grasp-return cycle can be executed in

simulation.

- It develops a VR teleoperation, dataset conversion, and two-stage ACT training workflow that produces a reproducible 160K-step manipulation baseline.
- It quantifies the domain shift encountered in the apartment scene, showing that direct transfer from the Stage 1 simple-scene checkpoint achieves 27% success, while mixed-dataset fine-tuning with apartment-scene upweighting recovers apartment-scene success to 83%.
- It implements and evaluates a residual PPO refinement pipeline over the frozen Stage 2 ACT policy, including matched ACT-only versus residual PPO comparisons that improve success from 85% to 93% in the simple scene and from 83% to 91% in the apartment scene, with a historical reference run providing supporting return-based evidence.

2 Background and Related Work

2.1 Long-Horizon Mobile Manipulation

Long-horizon mobile manipulation combines room-scale movement with contact-rich object interaction. Recent benchmark and system papers show that these tasks remain difficult because robots must integrate navigation, scene understanding, and grasp execution over extended horizons while remaining data efficient and robust to changing layouts and object placements. [1, 2, 3] A common response in this literature is hierarchical or modular decomposition, in which room-scale transit and local manipulation are handled by different components matched to their distinct spatial and temporal scales. That perspective motivates the present report’s separation between navigation and local learned manipulation.

2.2 Navigation with Nav2

For indoor navigation in static maps, mobile manipulation systems commonly rely on layered navigation stacks that combine localisation, planning, feedback control, and supervisory task logic rather than end-to-end learned navigation. This remains attractive when the room-scale transit problem is well structured and reliability matters more than learning the entire navigation policy from scratch. [4, 5, 6] Nav2 is a representative ROS 2 implementation of this classical stack, with the project-specific details documented in its supporting concepts and behaviour-tree references. [7, 8]

2.2.1 Localization in Indoor Mobile Robotics

In static indoor environments with a known occupancy map, probabilistic localisation methods in the adaptive Monte Carlo localisation (AMCL) tradition remain standard because they combine odometry and range sensing in a form that is both robust and computationally practical. Fox’s KLD-sampling formulation is a classic reference point for adaptive particle-filter localisation, while later work has extended AMCL-style pipelines with richer mapping information to improve robustness in indoor settings. [9, 10, 4] These methods remain particularly appropriate when the map is known in advance and the task does not depend on learned semantic localisation. For that reason, this report uses AMCL within Nav2 as the localisation layer.

2.2.2 Path and Motion Planning for Mobile Manipulation

Once localisation is available, room-scale mobile manipulation depends on more than path finding alone: it requires global planning, local control, and task-level coordination between transit and manipulation. Recent surveys of task and motion planning emphasise that feasible mobile manipulation depends on this interaction between high-level task structure and motion-level execution, and planning formulations for mobile manipulators likewise treat base motion and manipulation context as tightly coupled rather than independent problems. [5, 6] Nav2 packages the navigation portion of this stack through planner and controller servers together with behaviour-tree-based orchestration. [4, 8] In the present system, that makes Nav2 a natural choice for room-scale transit, leaving the learned components to focus on the short-horizon manipulation problem near the object.

2.3 Visuomotor Manipulation with ACT and LeRobot

Visuomotor manipulation differs from room-scale navigation in that control depends directly on local visual evidence, robot state, and contact-sensitive timing. Recent work has therefore shifted toward demonstration-driven visuomotor policies that learn the perception-to-action mapping directly from images and proprioception, rather than depending on hand-engineered perception modules alone. [11, 12, 3] ACT is one such policy, and LeRobot supplies a standard dataset format, training pipeline, checkpoint management, and inference utilities that make it easier to train and deploy in practice. [13, 14]

2.3.1 Image Processing and Visual Perception for Manipulation

In contemporary manipulation systems, image processing is often embedded inside an end-to-end visual representation rather than treated as a separate handcrafted front-end. RGB-based visuomotor policies are especially effective when the important information is local, such as object appearance, hand-object alignment, and short-range scene context near the grasp site. This design appears in fine-grained manipulation work based on ACT, in diffusion-based visuomotor control, and in whole-body teleoperated mobile manipulation systems such as Mobile ALOHA. [11, 12, 3] Compared with heavier generative control policies, ACT offers a simpler chunked-action formulation that remains well suited to the image-conditioned short-horizon control problem addressed in this report.

2.3.2 Pose and Grasp Estimation / Visuomotor Grasping

The literature on grasping spans two broad directions. One direction uses explicit pose or grasp-estimation pipelines, as surveyed in recent reviews of deep grasp synthesis and exemplified by modern object-pose systems such as FoundationPose for 6D pose estimation and tracking. [15, 16] A second direction uses learned visuomotor policies that fold pose-sensitive approach and grasp execution directly into the control policy itself, as in fine-grained ACT-style manipulation from demonstrations. [11] This report follows the second route: it does not use a standalone analytic grasp estimator, and instead uses ACT for pose-sensitive approach and grasp execution near the object.

2.4 Residual Reinforcement Learning

Residual reinforcement learning is best understood as learning a corrective policy on top of a fixed controller or base learned policy. Johannink et al. introduced this perspective for robot control, Alakuijala et al. showed how residual learning can be layered on top of demonstration-based policies,

and more recent work has applied the same idea to precise assembly and multi-fingered grasping, where local contact errors dominate performance near task completion. [17, 18, 19, 20] This literature makes residual refinement especially relevant when a base imitation policy already captures the coarse task structure but still makes local pose and timing errors near contact. That is precisely the role assigned to residual PPO on top of frozen ACT in this report.

3 System Overview

3.1 Task Definition

The system is organised around a four-phase task: (1) start from the home pose, (2) navigate to a manipulation pose near the target, (3) execute a grasp with the right arm and gripper, and (4) return to the home pose while holding the object. The architecture explicitly treats the navigation-to-manipulation handoff as a design boundary rather than forcing a single controller to span the entire episode.

3.2 Three-Layer Architecture

The final system has three layers:

1. **Navigation layer.** Nav2 performs room-scale transit from the home pose to a pose from which the object is reachable.
2. **Manipulation baseline.** ACT receives RGB and proprioceptive observations and outputs local joint-action chunks for approach and grasp.
3. **Residual refinement layer.** PPO applies a constrained corrective action over the frozen ACT output in the final alignment zone.

Figure 1 summarises the interfaces between these layers and highlights the handoff from room-scale navigation to local visuomotor control.

3.3 Key Architectural Decisions

The main design decisions are:

- Decompose the task by scale: navigation handles metres, ACT handles centimetres.
- Train manipulation by imitation first, then adapt it with targeted fine-tuning rather than retraining from scratch.
- Restrict residual PPO to a narrow action subspace and a narrow spatial zone near the grasp target.
- Evaluate the system as an integrated fetch task, not as isolated navigation and manipulation modules.

4 Navigation Stack

4.1 Design

The navigation subsystem uses the ROS 2 Nav2 stack under the `/tiago` namespace. AMCL provides localisation against a prebuilt occupancy map, the planner server produces a global path to the

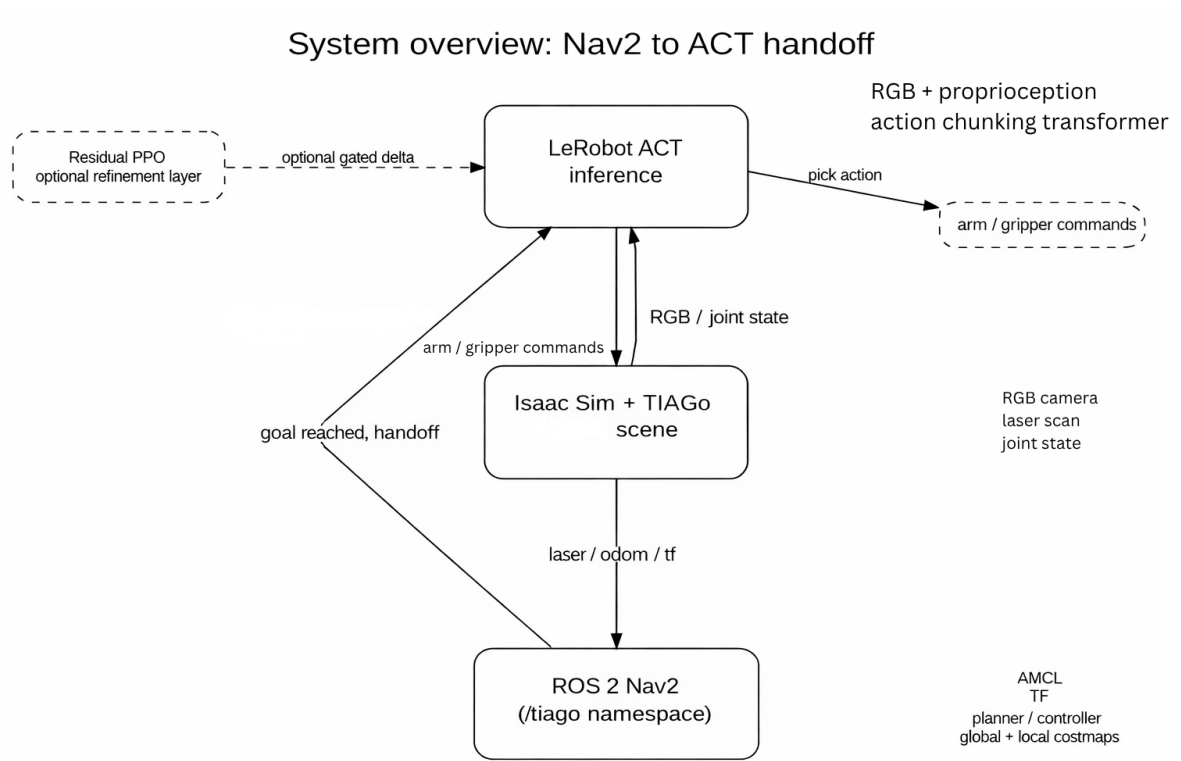


Figure 1: Three-layer system architecture showing Nav2 global transit, ACT local manipulation, and optional residual PPO correction applied over the ACT output near the grasp target.

workspace pose, the controller server tracks that path, and the behaviour-tree navigator manages task execution. [4, 7, 8] In this project, Nav2 is responsible for getting TIAGo to a manipulation-ready pose; local arm motion and grasping are delegated to the ACT-based manipulation layer.

Figure 2 shows the high-level Nav2 execution flow used in the final stack, from localisation and global planning through local control to base velocity commands.

4.2 Final Working Configuration

Three implementation details were essential to stable performance. First, launch files, parameter files, frame names, and topics all had to be kept consistent under the `/tiago` namespace; otherwise the stack would start but not behave coherently. Second, AMCL had to be explicitly initialised and verified at session start so that the robot began from a known pose. Third, both global and local costmaps had to publish correctly before issuing goals, because the manipulation handoff only matters if navigation first reaches the intended workspace pose. Together, these points define the final validated navigation configuration used in the report.

Once those points were resolved, navigation was reliable in validated runs. The BT navigator returned success for the integrated demo, and the navigation-to-manipulation handoff became a stable boundary in the full pipeline.

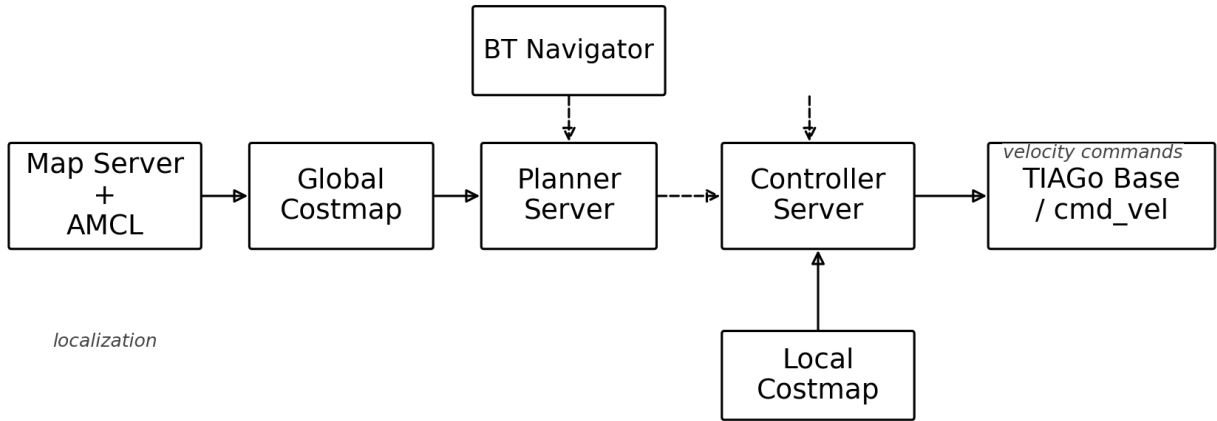


Figure 2: Nav2 execution flow used in the navigation stack, showing the interaction between map server and AMCL, the planner and controller servers, the BT navigator, local and global costmaps, and the resulting TIAGo base velocity commands.

5 Manipulation Baseline with ACT

5.1 Why ACT

ACT was chosen as the manipulation baseline because the final approach-and-grasp phase is a visuomotor control problem with strong temporal structure. The policy must interpret the object from camera observations while coordinating arm, torso, and gripper motion over a short horizon. ACT is well suited to that setting because it predicts multi-step action chunks rather than isolated single-step commands, which improves smoothness and reduces the jitter that often accumulates in purely one-step policies. [11]

5.2 Observation, Action, and Inference Pipeline

At control step t , the deployed policy receives an RGB observation together with a low-dimensional proprioceptive state and predicts an action chunk over a finite horizon. In the configuration used here, the proprioceptive state vector had dimension 85, the action vector also had dimension 85, and the chunk horizon was fixed to $H = 100$. Denoting the RGB observation by I_t , the proprioceptive state by $x_t \in \mathbb{R}^{85}$, and the predicted action sequence by $\hat{\mathbf{u}}_{t:t+H-1}$, the deployed ACT policy can be written as

$$\hat{\mathbf{u}}_{t:t+H-1} = \pi_{\theta}(I_t, x_t), \quad H = 100, \quad (1)$$

with each per-step action $\hat{\mathbf{u}}_{t+k} \in \mathbb{R}^{85}$. In practice, the RGB stream provides the dominant scene information, while the 85-dimensional state carries the proprioceptive channels needed to condition the action chunk on the current arm, gripper, and torso configuration.

Inference uses the LeRobot deployment stack so that the same dataset statistics used in training are also used at runtime. Incoming low-dimensional channels are normalised before the forward pass, and predicted actions are unnormalised before they are dispatched to the articulation controller. That training-consistent normalisation path was important for keeping deployment numerically aligned with the distribution seen during optimisation. [13, 14]

5.3 ACT Architecture and Objective

The manipulation policy used an ACT configuration with a ResNet-18 vision backbone feeding a transformer of width $d_{\text{model}} = 512$. Multi-head attention used $n_{\text{heads}} = 8$, the feedforward blocks had dimension 3200, the encoder depth was 4 layers, and the decoder depth was 7 layers. A latent-variable model was enabled (`use_vae=true`) with latent dimension 32 and dropout 0.1 throughout the transformer stack. This architecture was chosen to balance two competing demands of the task: the visual front end had to remain lightweight enough for stable offline imitation learning, while the sequence model still had to be expressive enough to represent multimodal short-horizon behaviours during final wrist alignment, approach correction, and gripper closure.

Training optimised the standard ACT objective combining action reconstruction with latent regularisation,

$$\mathcal{L}_{\text{ACT}} = \mathcal{L}_1 + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}}, \quad \lambda_{\text{KL}} = 10. \quad (2)$$

The $\mathcal{L}_1$ term drives accurate reproduction of demonstrated action chunks, while the KL term regularises the latent representation induced by the VAE component. In this project, that combination was preferable to a purely deterministic regression objective because the demonstrations contain small but meaningful variation in final grasp approach trajectories.

5.4 Final Deployment Configuration

Four deployment details were decisive for stable ACT rollouts. First, input normalisation and output unnormalisation had to match the LeRobot training statistics; otherwise the arm produced physically implausible command magnitudes. Second, temporal ensembling was used to smooth overlapping chunk predictions and suppress high-frequency jitter. Third, the continuous gripper channel was mapped to a hysteresis-based open/close attachment rule so that inference-time grasp events matched the semantics of the demonstrations. Fourth, the set of pickable objects was restricted to intended targets so that the policy could not spuriously attach to nearby geometry such as the tabletop. These are part of the evaluated ACT baseline rather than incidental implementation details. [14]

Figure 3 shows the full ACT workflow from VR teleoperation and LeRobot dataset conversion to checkpoint training and runtime inference.

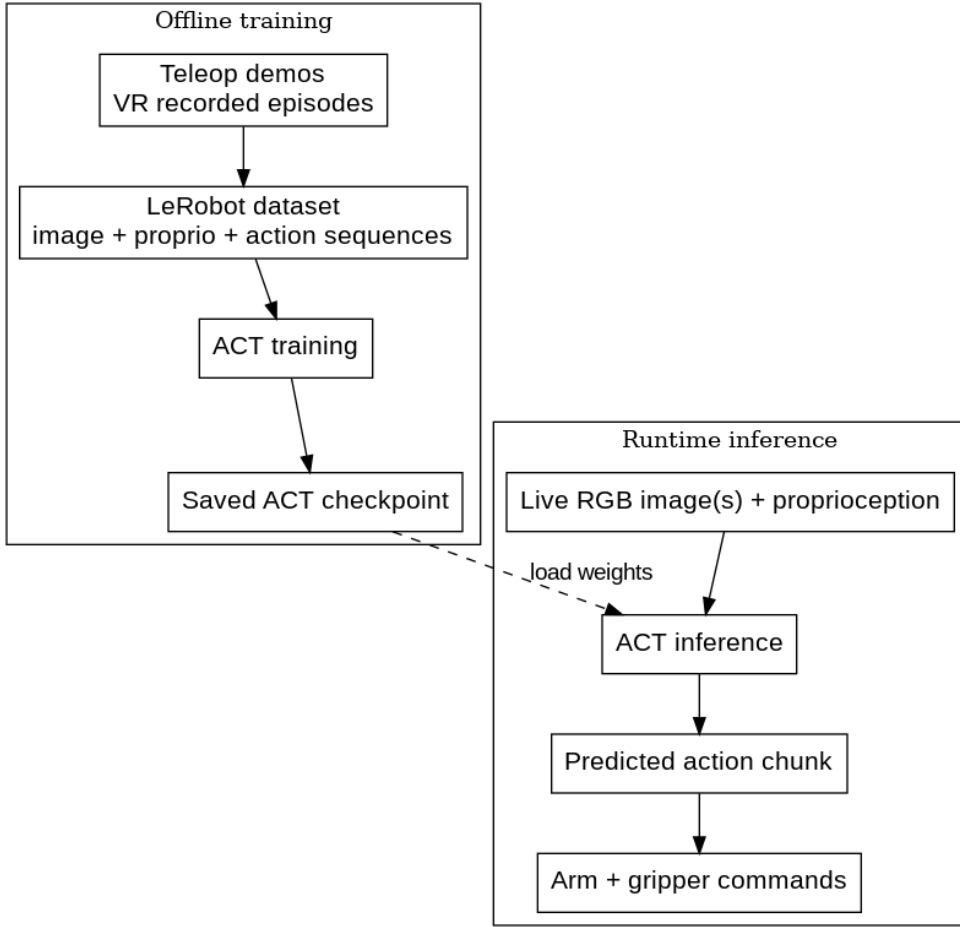


Figure 3: ACT training and inference pipeline showing the full workflow from VR teleoperation and LeRobot dataset conversion to checkpoint training and runtime inference.

6 Data Collection, Training, and Fine-Tuning

6.1 VR Teleoperation and Recorded Data

Demonstrations were collected with a VR teleoperation setup built from a Meta Quest 3S, SteamVR/ALVR streaming, OpenXR in Isaac Sim, and the Lula inverse-kinematics solver for TIAGo’s right arm. The right-hand controller pose was mapped to a target end-effector pose, solved through IK, and applied to the robot online, while grasp and release were controlled through a trigger-driven fixed-joint attachment mechanism consistent with the logic later used during ACT inference. This overall teleoperation-based data-collection design is consistent with recent mobile-manipulation demonstration pipelines such as Mobile ALOHA. [3]

At each step, the recorder stored synchronised RGB observations, proprioceptive joint states, and action targets. This produced data already aligned with the ACT training interface and reduced train–deployment mismatch by preserving the same interaction semantics between demonstration

collection and policy execution.

6.2 Dataset Curation and Conversion

The simple-scene dataset was cleaned to 79 usable episodes after failed or corrupted demonstrations were removed; this corpus became the Stage 1 training set. Additional apartment-scene demonstrations were then recorded, cleaned, and converted into the same LeRobot format. After integration and apartment-scene upweighting, the Stage 2 combined dataset contained 226 episodes. The purpose of the second corpus was not merely to enlarge the dataset, but to expose the policy more consistently to the visual and geometric statistics of the target apartment scene.

6.3 Two-Stage ACT Training

ACT training used AdamW with learning rate 10^{-5} and weight decay 10^{-4} . The batch size was 8, the dataloader used 4 workers, gradient clipping used a norm of 10, training metrics were logged every 100 steps, and checkpoints were saved every 10,000 steps.

Stage 1 trained from scratch from step 0 to 80,000 on the 79-episode simple-scene dataset. Stage 2 then resumed from the Stage 1 checkpoint and continued training to 160,000 global steps on the 226-episode combined corpus. Appendix A summarises the two-stage ACT training schedule in Table 8. The quantitative comparison between the resulting Stage 1 and Stage 2 checkpoints is deferred to Section 10, where both policies are evaluated under the same protocol.

Table 1 summarises the ACT model and training configuration used in the reported experiments.

Item	Setting
Input modalities	RGB observation with proprioceptive state
State and action dimensions	85-dimensional state, 85-dimensional action
Chunk horizon	$H = 100$
Vision backbone	ResNet-18
Transformer size	$d_{\text{model}} = 512$, 8 heads, 4 encoder layers, 7 decoder layers
Optimizer	AdamW
Learning rate	10^{-5}
Batch size	8
Training stages	Stage 1: 0–80,000 steps; Stage 2: resume to 160,000 global steps

Table 1: ACT model and training configuration summary.

7 Apartment-Scene Adaptation and Domain Shift

This section analyses transfer from the simple scene, which serves as the minimal baseline environment for ACT training and evaluation, to the apartment scene, which serves as the more realistic target environment for transfer, adaptation, and integrated system testing. Figure 4 summarises the two evaluation environments and helps frame the domain-shift discussion around differences in visual context, scene layout, and grasp geometry.



Figure 4: Evaluation environments used in the report. Left: simple scene used for baseline ACT training and evaluation. Right: photorealistic indoor apartment scene used for transfer, adaptation, and integrated system testing.

7.1 Transfer Failure

The apartment scene is a more realistic evaluation environment than the simple training scene: it has richer visual context, a more complex layout, and a taller table that changes the grasp geometry. When the Stage 1 ACT checkpoint trained only on the simple scene was transferred directly to this apartment scene, the resulting behaviour was systematically poor. The most recognisable failure mode was an oscillating vertical approach trajectory in which the arm alternated between ascent and descent without converging to the object.

7.2 Diagnosis

That failure is best explained as domain shift rather than random instability. Three factors were especially important: the richer visual context changed the observation distribution, the different scene layout altered the approach path and occlusion pattern, and the taller table changed the required grasp geometry. In short, both the observation distribution and the required action distribution moved at the same time.

7.3 Adaptation Strategy

The adaptation strategy combined targeted runtime corrections with new data. Only the fixes that affected the method or the reported results are retained here: an apartment-scene-safe launcher with deterministic spawn behaviour and fallback, torso-height initialisation appropriate for the taller table, robust object-path resolution, and a consistent frame convention for recording and playback. These changes made the apartment-scene evaluation stable enough to support data collection and deployment.

The policy adaptation itself came from new apartment-scene demonstrations and mixed-dataset fine-tuning. Apartment-scene examples were added to the training corpus and upweighted during Stage 2 so that the policy would learn the visual and geometric conditions of the target scene rather than simply overfitting to the simple baseline environment. [21]

8 Residual PPO on Frozen ACT

8.1 Motivation

Even a strong imitation-learned manipulation policy can still make contact-critical errors in the final centimetres of a grasp: the wrist can approach at a slightly wrong angle, the fingers can close before the object is centred, or the end-effector can drift just enough to turn a near miss into a failed pickup. Those errors are local enough that they are better treated as a correction problem than as a reason to relearn the entire manipulation policy. [17, 18, 19, 20]

8.2 Residual RL Formulation

In this project, PPO is used as a residual controller layered over a frozen ACT baseline rather than as a replacement for imitation learning. Let a_t^{ACT} denote the action proposed by ACT at control step t , and let Δa_t^{RL} denote the correction predicted by PPO. The executed action is

$$a_t = a_t^{\text{ACT}} + \Delta a_t^{\text{RL}}. \quad (3)$$

During PPO training and inference, the ACT weights remain fixed and only the residual policy is updated. This separation preserves the coarse approach behaviour learned from demonstrations while allowing reinforcement learning to optimise the final contact-sensitive adjustments that imitation alone did not fully resolve. [17, 18]

The residual is also spatially gated. In the narrow-residual configuration, the correction is applied only when the end-effector is inside a local zone around the object and is zeroed elsewhere, so the ACT action passes through unchanged outside the last-mile grasp region. That masking keeps PPO focused on the part of the episode where small geometric corrections matter and avoids using RL to relearn motions that ACT already handles adequately.

8.3 Environment and Observation/Action Design

Residual PPO was trained in a direct-control Isaac Lab environment centred on the final approach to the object. The observation design was intentionally compact: instead of exposing the full scene state, the policy received a 42-dimensional vector describing the local grasp geometry. Expressing the tool centre point (TCP) in an object-centred frame removes irrelevant world-frame variation and makes the residual policy reason directly about the relative pose that determines grasp success.

The 42-dimensional observation emphasises quantities that are meaningful near contact. These include the TCP position relative to the object, a minimum rotation error computed against a symmetric set of candidate grasp orientations around the object, low-dimensional geometric terms, finger-span information, and the finite-state phase indicator used by the reward logic. The symmetric rotation term is important because a cylindrical object admits equivalent grasp poses under discrete rotations about its axis; measuring error to the nearest valid target avoids penalising geometrically identical wrist orientations. Likewise, the geometric and finger-span channels expose whether the hand is approaching with a plausible opening and alignment for antipodal closure.

The residual action is defined over selected control channels rather than over the full robot. In narrow mode, the non-zero correction is applied primarily to wrist and gripper joints, while the larger arm trajectory and other motion components continue to follow ACT. This choice makes the policy’s action space match the intended role of the residual controller: not to replace the demonstration policy, but to refine the final local alignment and closure behaviour.

8.4 Two-Phase Reward Logic and Residual Gating

The reward logic is split into ALIGN and GRASP phases because the manipulation objective changes qualitatively at contact. In ALIGN, the policy is encouraged to reduce translational and rotational error to the canonical grasp pose while staying open and approaching cleanly. Early-close penalties discourage finger closure before the hand is properly positioned, and side-contact penalties suppress lateral brushing that often precedes pushes or unstable grasps. Once the finite-state machine enters GRASP, the reward shifts toward gap alignment, antipodal shaping, lift, and terminal success. This phase structure matters because it prevents the policy from conflating pre-contact approach behaviour with post-contact lifting behaviour under a single undifferentiated signal.

Residual gating further constrains where PPO can intervene. Outside the configured narrow zone around the object, the residual is zeroed and the frozen ACT action is executed unchanged. Inside that zone, PPO can apply its selected joint corrections, and the diagnostic `narrow_residual_in_zone_frac` records the fraction of rollout time for which the end-effector was inside the zone and therefore eligible for residual control. For the historical PPO reference run, the shaped reward was scaled by $\alpha = 0.1$. The full reward definition and term-by-term specification are given in Appendix B.

8.5 Residual PPO Training Configuration

Training used PPO through RL-Games with ACT held fixed and the controller operating in residual mode. The historical reference run used reward scale $\alpha = 0.1$, which kept dense shaping terms numerically modest while preserving a large terminal incentive for a successful grasp. The most important named terms in this configuration were the success bonus, early-close penalty, side-contact proxy penalty, and quadratic residual-action regularisation. In effect, PPO was asked to learn a small, high-value correction policy on top of a strong imitation baseline. [19, 20] The corresponding historical reference result is reported in Section 10.4.

Table 2 summarises the residual PPO configuration used in the reported experiments.

Item	Setting
Algorithm	PPO (RL-Games)
Control mode	Frozen ACT baseline with additive residual PPO correction
Observation design	42-dimensional local grasp state with TCP in an object-centred frame, minimum symmetric rotation error, geometric terms, finger span, and FSM phase indicator
Residual action scope	Selected joints only; mainly wrist and gripper channels in narrow-residual mode
Reward shaping scale	$\alpha = 0.1$ in the historical reference run
Success bonus	$+50.0 \text{ success}_t m_t$
Early-close penalty	$-0.8 \text{ early_close}_t (1 - m_t)$
Side-contact proxy penalty	$-1.2 \text{ side_contact_proxy}_t (1 - m_t)$
Action regularization	$-0.0005 \ a_t\ _2^2$

Table 2: Residual PPO setup summary.

9 Experimental Setup and Evaluation Protocol

9.1 Experimental Environment

All experiments were conducted in NVIDIA Isaac Sim 5.1 using the TIAGo dual-functional robot model. The evaluation machine used an NVIDIA RTX 4090 GPU. Two scenes were used: the original simple scene for baseline ACT evaluation and the apartment scene for transfer and adaptation experiments. The navigation map was fixed, and the target object was placed upright on a table.

Accordingly, the reported system assumes a static indoor scene, a fixed navigation map, a single target object placed upright on a table, and simulation-only evaluation in Isaac Sim.

9.2 Integrated Demo Scenario

The integrated demonstration follows the same sequence used to motivate the architecture: TIAGo starts at home, Nav2 drives it to a manipulation pose, the manipulation subsystem takes over, the object is grasped, and the robot returns to the home pose carrying it. This sequence is important because it shows that the handoff between navigation and manipulation works in the complete system rather than only in isolated module tests.

Figure 5 shows the integrated execution context used for this end-to-end fetch loop.

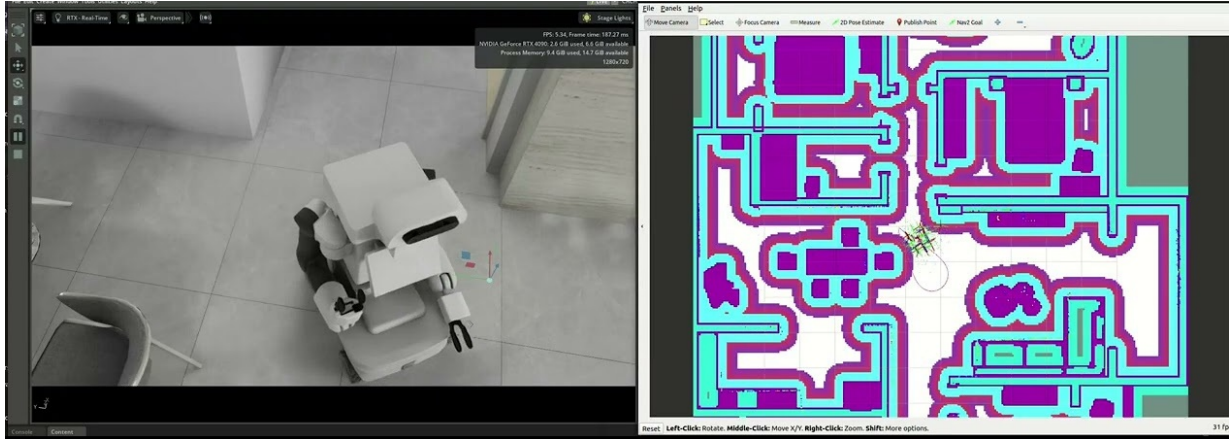


Figure 5: Integrated system demo view during the fetch loop. The Isaac Sim scene (left) and the concurrent Nav2 view (right) show the combined navigation-to-manipulation execution context.

9.3 Evaluation Protocol

Four evaluation views were used in the report:

- **Navigation validation.** A navigation run is successful when Nav2 reaches the workspace pose and the behaviour-tree navigator returns success.
- **Simple-scene ACT comparison.** Stage 1 and Stage 2 ACT checkpoints are run under the same evaluation code. A trial is successful when the gripper attaches the object and retains it after lift.
- **Apartment-scene adaptation.** Direct transfer from Stage 1 is compared quantitatively against the fine-tuned Stage 2 behaviour in the apartment scene under the same success criterion.

- **Residual PPO evaluation.** Residual PPO is assessed in both scenes under the same task conditions, success criterion, and 299-step horizon and termination logic as the corresponding ACT-only grasp evaluations. The primary quantitative metric is success rate; secondary diagnostics include failure rate, geometric alignment terms, and episode-length trends. A final large-sample locked ACT-versus-PPO comparison remains future work.

The Stage 1 versus Stage 2 ACT comparison in the simple scene used the same evaluation procedure and 30 trials per checkpoint. The reported success-rate difference therefore reflects the effect of the additional training and combined dataset, not a changed evaluation script.

For residual PPO, holding the scene, success logic, and rollout horizon fixed is equally important: otherwise an apparent gain in reward could be explained by easier initial conditions or by shorter episodes rather than by better grasp refinement. The PPO result discussed in Section 10 is therefore interpreted together with episode-length traces and episode-level geometric diagnostics rather than reward alone.

Table 3 summarises the common simulator, robot, scene, and evaluation settings used across the reported experiments.

Item	Setting
Simulator	NVIDIA Isaac Sim 5.1
Robot	TIAGo dual-functional mobile manipulator
Scenes	Simple scene and apartment scene
Navigation success	Nav2 reaches the workspace pose and returns success
Manipulation success	Object is attached and retained after lift
ACT comparison	Stage 1 vs Stage 2 in the simple scene, plus Stage 1 direct transfer vs Stage 2 in the apartment scene; 30 trials per reported condition
PPO evaluation	Stage 2 ACT baseline vs residual PPO in both scenes, same success criterion and 299-step horizon; primary metric success rate, secondary metrics failure rate, geometric terms, and episode-length trends

Table 3: Experimental setup and evaluation summary.

10 Results and Discussion

Table 4 provides the quantitative summary of the final results across both evaluation environments, and Table 5 summarises the same outcome at the component level before the subsection-level discussion.

Environment	Stage 1 ACT	Stage 2 ACT, combined	Stage 2 ACT + residual PPO
Simple scene	72%	85%	93%
Apartment scene	27%	83%	91%

Table 4: Cross-environment success-rate summary for the final reported policies.

Component	Main result
Nav2 navigation	Stable transit to the manipulation pose in validated integrated runs, supporting a reliable navigation-to-manipulation handoff.
ACT baseline	In the simple scene, Stage 1 achieved 72% success and Stage 2 improved this to 85% over the same 30-trial protocol.
Apartment-scene adaptation	In the apartment scene, direct transfer from the Stage 1 simple-scene checkpoint achieved 27%, while Stage 2 mixed-dataset training recovered success to 83%.
Residual PPO	Matched evaluation improved success from 85% to 93% in the simple scene and from 83% to 91% in the apartment scene over the corresponding frozen Stage 2 ACT baselines; the historical reference run also improved mean episode return from approximately -531 to a peak near -209 while episode length remained fixed.

Table 5: Main results summary across the navigation, ACT, apartment-scene adaptation, and residual PPO components.

10.1 Navigation Results

The navigation subsystem achieved the intended role in the hierarchy: Nav2 provided stable transit to the manipulation pose in validated runs.

10.2 ACT Training and Simple-Scene Performance

The ACT baseline learned the simple-scene grasping behaviour from the teleoperation corpus and produced stable rollouts once the final inference configuration was in place. The Stage 2 checkpoint was visibly smoother and more reliable than earlier checkpoints, which is consistent with the additional training time and the larger combined corpus. Appendix A provides the two-stage ACT training summary in Table 8.

The primary quantitative ACT result in the simple scene is the Stage 1 versus Stage 2 comparison shown in Table 6. Continuing training from 80K to 160K on the combined dataset improved success rate from 72% to 85% under the same evaluation procedure. In the broader cross-environment summary of Table 4, the same Stage 2 training also provides the baseline from which apartment-scene recovery and residual PPO refinement are evaluated.

Variant	Training Data	Trials	Success Rate
Stage 1	Simple-scene dataset only	30	72%
Stage 2	Combined dataset with apartment-scene upweighting	30	85%

Table 6: Simple-scene ACT comparison under the same evaluation code.

10.3 Apartment-Scene Adaptation Results

The apartment-scene experiments convert the domain-shift story into a quantitative one. As summarised in Table 4, direct transfer from the simple-scene Stage 1 checkpoint achieved only 27% success in the apartment scene, confirming that the shift in visual context, layout, and grasp geometry was substantial. After Stage 2 mixed-dataset training with apartment-scene data, apartment-scene

success improved to 83%, showing that the added data and resumed ACT training recovered most of the lost performance.

Residual PPO then improved apartment-scene success further from the frozen 83% Stage 2 ACT baseline to 91%. Taken together, these results show that mixed-dataset ACT training was the main recovery mechanism for domain shift, while residual PPO provided a further, consistent refinement on top of the recovered baseline.

10.4 Residual PPO Results

The residual PPO pipeline produced a meaningful matched improvement over the frozen Stage 2 ACT baseline in both environments. Because PPO was applied as a residual over the frozen Stage 2 ACT policy, the correct ACT baseline for comparison is the final Stage 2 checkpoint rather than the earlier Stage 1 result. Table 7 isolates this comparison: in the simple scene, success improved from 85% for the ACT-only baseline to 93% for the residual PPO variant, while in the apartment scene it improved from 83% to 91%.

In the historical reference run, the mean training return (**rewards/iter**) improved from approximately -531 to a peak near -209 . Figure 6 shows this trend directly. Just as importantly, the companion episode-length curve remained flat at the fixed 299-step horizon throughout training.

Environment	Stage 2 ACT baseline	Residual PPO	Gain
Simple scene	85%	93%	+8 points
Apartment scene	83%	91%	+8 points

Table 7: Matched Stage 2 ACT-only versus residual PPO comparison across both evaluation environments.

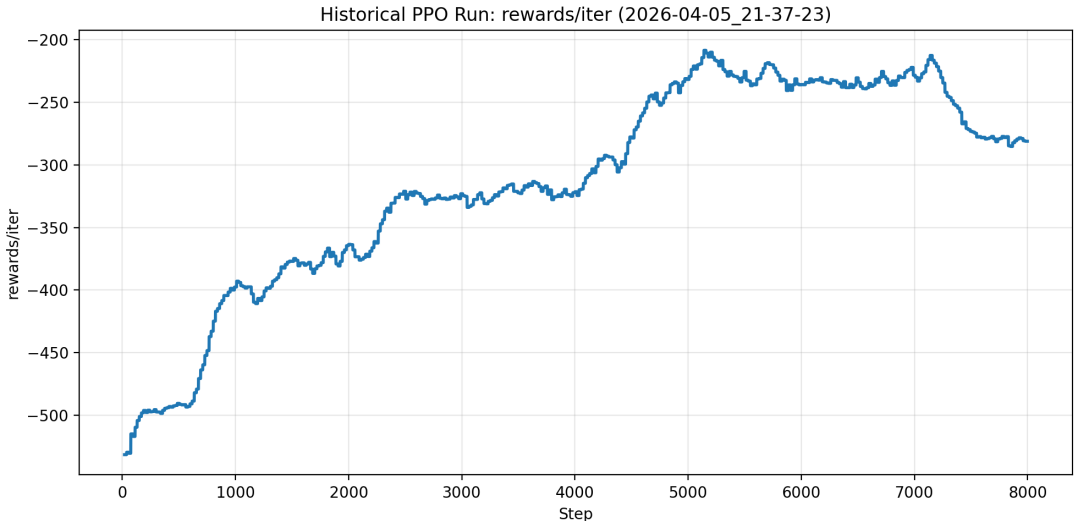


Figure 6: Historical PPO training return showing mean episode return improving from approximately -531 to a peak near -209 , supporting the claim that residual PPO improved behaviour over the frozen Stage 2 ACT baseline.

This interpretation is consistent with direct side-by-side qualitative evaluation. Under the same scene and task conditions, the best PPO checkpoint aligned faster near the object and produced cleaner grasp closure than the ACT-only Stage 2 baseline, especially in the final approach where small wrist-angle and closure-timing errors matter most. Taken together, the matched 85% to 93% improvement in the simple scene, the matched 83% to 91% improvement in the apartment scene, the fixed-horizon return improvement, and the cleaner qualitative last-mile refinement provide strong evidence for the PPO layer over the frozen Stage 2 ACT policy.

The supporting episode-length control figure is provided in Appendix A as Figure 7.

10.5 Discussion

Taken together, the results support the central claim of the report: the fetch task is better handled as a hierarchy than as a single monolithic problem. Nav2 solved the room-scale transit problem reliably once configured correctly. In the simple scene, resumed ACT training improved success from 72% in Stage 1 to 85% in the final Stage 2 ACT-only baseline. In the apartment scene, direct transfer from the Stage 1 simple-scene checkpoint achieved only 27%, confirming strong domain shift, while Stage 2 mixed-dataset training recovered success to 83%. Residual PPO then provided a further, consistent refinement in both environments, improving the frozen Stage 2 ACT baseline from 85% to 93% in the simple scene and from 83% to 91% in the apartment scene.

The report’s quantitative progression is therefore $72\% \rightarrow 85\% \rightarrow 93\%$ in the simple scene and $27\% \rightarrow 83\% \rightarrow 91\%$ in the apartment scene. Together with the historical PPO return improvement from approximately -531 to a peak near -209 , treated here as supporting evidence rather than the primary result, these matched success-rate gains show that both the baseline and the refinement layer moved in the intended direction.

11 Limitations

11.1 Simulation-Only Evaluation

All experiments were performed in simulation. That means the results do not address sim-to-real issues such as sensor noise, real contact dynamics, hardware latency, or calibration drift. [22] The navigation layer is structurally closer to deployable real-robot software than the learned manipulation components, but the report still makes no real-world performance claim.

11.2 Incomplete Quantitative Coverage

The report now includes matched 30-trial success-rate results for both the simple scene and the apartment scene, but two important quantitative limitations remain. First, the reported evaluation budget is still modest, so the success-rate estimates would benefit from larger trial counts and explicit confidence intervals. Second, although matched ACT-versus-residual-PPO comparisons are reported in both environments, a larger-sample comparison under a fully locked protocol remains future work.

11.3 Operational Constraints

The manipulation pipeline remains sensitive to scene configuration, especially robot spawn pose, torso height, and object placement in the apartment scene. The VR teleoperation pipeline was

good enough to collect the reported dataset, but it was not a frictionless data-collection system; it required controlled session startup and stable XR runtime conditions.

12 Conclusion and Future Work

12.1 Conclusion

This report presented a three-layer architecture for long-horizon household item fetching in Isaac Sim. The final system combines Nav2 for global transit, ACT for local visuomotor manipulation, and residual PPO for narrow corrective refinement. Validated integrated runs indicate that the full navigate-grasp-return cycle can be executed in simulation. Quantitatively, the simple-scene progression was 72% for Stage 1 ACT, 85% for Stage 2 ACT on the combined dataset, and 93% after residual PPO refinement. In the apartment scene, direct transfer from the simple-scene Stage 1 checkpoint achieved 27%, Stage 2 mixed-dataset ACT training recovered success to 83%, and residual PPO improved it further to 91%. The historical PPO reference run further improved mean episode return from approximately -531 to a peak near -209 while episode length stayed fixed. Together, these results support the claim that hierarchical decomposition was the right design choice for the task.

12.2 Future Work

The most important next steps are clear:

1. extend the reported simple-scene and apartment-scene evaluations to larger trial counts with confidence intervals and broader seed coverage;
2. execute a larger-sample, fully locked ACT-versus-residual-PPO comparison under matched seeds and episode budgets;
3. broaden the apartment-scene dataset to cover more spawn poses, object placements, and approach geometries; and
4. investigate sim-to-real transfer only after the simulation-side evaluation is quantitatively stronger, in line with the broader transfer challenges surveyed in [22].

References

- [1] T. Yenamandra, A. Ramachandran, K. Yadav, A. Wang, M. Khanna, T. Gervet, T. Yang, V. Jain, A. Clegg, J. Turner, D. Batra, D. Chaplot, D. Hillis, J. Johnson, A. Kumar, S. Rusinkiewicz, D. Schwartz, C. Sherp, A. Srouji, and Y. Artzi, “LAMBDA: A benchmark for data-efficiency in long-horizon indoor mobile manipulation,” *arXiv preprint arXiv:2412.05313*, 2024. [Online]. Available: <https://arxiv.org/abs/2412.05313>
- [2] H. Saha, F. Fotouhi, Q. Liu, and S. Sarkar, “A modular vision language navigation and manipulation framework for long horizon compositional tasks in indoor environment,” *Frontiers in Robotics and AI*, vol. 9, p. 930486, 2022. [Online]. Available: <https://doi.org/10.3389/frobt.2022.930486>
- [3] Z. Fu, T. Z. Zhao, and C. Finn, “Mobile ALOHA: Learning bimanual mobile manipulation using low-cost whole-body teleoperation,” in *Proceedings of The 8th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 270. PMLR, 2025, pp. 4066–4083. [Online]. Available: <https://proceedings.mlr.press/v270/fu25b.html>
- [4] S. Macenski, F. Martín, R. White, and J. G. Clavero, “The Marathon 2: A navigation system,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 2718–2725. [Online]. Available: <https://doi.org/10.1109/IROS45743.2020.9341207>
- [5] H. Guo, F. Wu, Y. Qin, R. Li, K. qin Li, and K. Li, “Recent trends in task and motion planning for robotics: A survey,” *ACM Computing Surveys*, vol. 55, no. 13s, pp. 1–36, 2023. [Online]. Available: <https://doi.org/10.1145/3583136>
- [6] Z. Jiao, Z. Zhang, W. Wang, D. Han, S.-C. Zhu, Y. Zhu, and H. Liu, “Efficient task planning for mobile manipulation: A virtual kinematic chain perspective,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 8288–8294. [Online]. Available: <https://doi.org/10.1109/IROS51168.2021.9636554>
- [7] Nav2 Documentation, “Nav2 concepts,” Online documentation. [Online]. Available: <https://docs.nav2.org/concepts/index.html>
- [8] —, “Nav2 behavior trees,” Online documentation. [Online]. Available: https://docs.nav2.org/behavior_trees/
- [9] D. Fox, “Adapting the sample size in particle filters through KLD-sampling,” *The International Journal of Robotics Research*, vol. 22, no. 12, pp. 985–1003, 2003. [Online]. Available: <https://doi.org/10.1177/0278364903022012001>
- [10] M. Peavy, P. Kim, H. Oyediran, and K. Kim, “Integration of real-time semantic building map updating with adaptive monte carlo localization (AMCL) for robust indoor mobile robot localization,” *Applied Sciences*, vol. 13, no. 2, p. 909, 2023. [Online]. Available: <https://doi.org/10.3390/app13020909>
- [11] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn, “Learning fine-grained bimanual manipulation with low-cost hardware,” in *Proceedings of Robotics: Science and Systems (RSS)*, Daegu, Republic of Korea, Jul. 2023. [Online]. Available: <https://roboticsproceedings.org/rss19/p016.html>

- [12] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. C. M. Burchfiel, and S. Song, “Diffusion policy: Visuomotor policy learning via action diffusion,” in *Proceedings of Robotics: Science and Systems (RSS)*, Daegu, Republic of Korea, Jul. 2023. [Online]. Available: <https://roboticsproceedings.org/rss19/p026.html>
- [13] Hugging Face, “LeRobot: State-of-the-art AI for real-world robotics,” GitHub repository. [Online]. Available: <https://github.com/huggingface/lerobot>
- [14] —, “ACT (action chunking with transformers),” LeRobot documentation. [Online]. Available: <https://huggingface.co/docs/lerobot/en/act>
- [15] R. Newbury, M. Gu, L. Chumbley, A. Mousavian, C. Eppner, J. Leitner, J. Bohg, A. Morales, T. Asfour, D. Kragic, D. Fox, and A. Cosgun, “Deep learning approaches to grasp synthesis: A review,” *IEEE Transactions on Robotics*, vol. 39, no. 5, pp. 3994–4015, 2023. [Online]. Available: <https://doi.org/10.1109/TRO.2023.3280597>
- [16] B. Wen, W. Yang, J. Kautz, and S. Birchfield, “FoundationPose: Unified 6D object pose estimation and tracking of novel objects,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024, pp. 17 868–17 879. [Online]. Available: <https://doi.org/10.1109/CVPR52733.2024.01692>
- [17] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine, “Residual reinforcement learning for robot control,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 6023–6029. [Online]. Available: <https://doi.org/10.1109/ICRA.2019.8794127>
- [18] M. Alakuijala, G. Dulac-Arnold, J. Mairal, J. Ponce, and C. Schmid, “Residual reinforcement learning from demonstrations,” *CoRR*, vol. abs/2106.08050, 2021. [Online]. Available: <https://arxiv.org/abs/2106.08050>
- [19] L. Ankile, A. Simeonov, I. Shenfeld, M. Torne, and P. Agrawal, “From imitation to refinement – residual RL for precise visual assembly,” *CoRR*, vol. abs/2407.16677, 2024. [Online]. Available: <https://arxiv.org/abs/2407.16677>
- [20] F. Ceola, L. Rosasco, and L. Natale, “RESPRECT: Speeding-up multi-fingered grasping with residual reinforcement learning,” *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3045–3052, 2024. [Online]. Available: <https://doi.org/10.1109/LRA.2024.3363532>
- [21] W. Zhang, Y. Li, Y. Qiao, S. Huang, J. Liu, F. Dayoub, X. Ma, and L. Liu, “Effective tuning strategies for generalist robot manipulation policies,” *CoRR*, vol. abs/2410.01220, 2024. [Online]. Available: <https://arxiv.org/abs/2410.01220>
- [22] W. Zhao, J. Peña Queraltá, and T. Westerlund, “Sim-to-real transfer in deep reinforcement learning for robotics: A survey,” in *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2020, pp. 737–744. [Online]. Available: <https://arxiv.org/abs/2009.13303>

Appendix A Additional Figures and Tables

Table 8 consolidates the two-stage ACT training schedule, and Figure 7 provides the PPO episode-length control curve cited in Section 10.4.

Appendix A.1 Additional Tables

Stage	Step Range	Dataset	Key Output
Stage 1	0 $\rightarrow$ 80,000	Simple-scene dataset (79 clean episodes)	ACT checkpoint 080000
Stage 2	80,000 $\rightarrow$ 160,000	Combined corpus (226 episodes with apartment-scene upweighting)	ACT checkpoint 160000

Table 8: Two-stage ACT training summary.

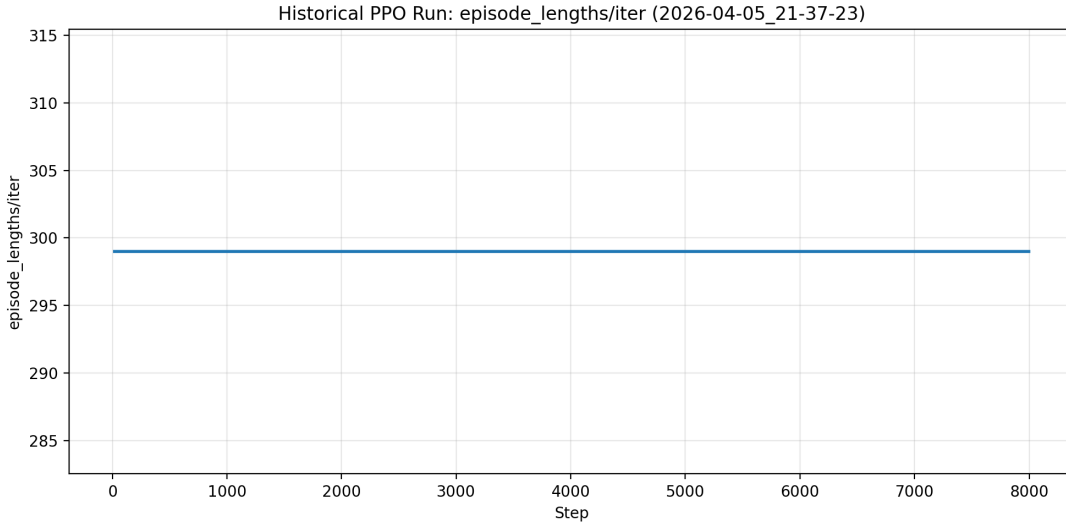


Figure 7: Historical PPO episode-length control showing that episode length remained fixed at the rollout horizon, so the return improvement is not explained by shorter rollouts.

Appendix B Residual PPO Reward Definition

This appendix is the reproducibility specification for the reward used in the historical residual PPO reference run. The main body summarises the two-phase logic and residual gating, while the full equation, gate definition, and term-by-term descriptions are collected here.

$$R_t = 0.1 r_{\text{env},t}, \quad (4)$$

$$r_{\text{env},t} = 0.01 \text{ alive}_t - 0.10 \text{ dist}_t + 50.0 \text{ success}_t m_t + 4.0 \text{ centered}_t \text{ near}_t m_t \quad (5)$$

$$- 1.0 \text{ bad_close}_t - 0.8 \text{ early_close}_t (1 - m_t) - 1.2 \text{ side_contact_proxy}_t (1 - m_t) \quad (6)$$

$$- 0.05 \text{ push_speed}_t \text{ far}_t - 0.0005 \|a_t\|_2^2 \quad (7)$$

$$+ (1 - m_t)(0.5 \text{ approach_pos}_t + 1.0 \text{ approach_rot}_t) \quad (8)$$

$$+ m_t(0.5 \text{ gap_align}_t + 0.8 \text{ antipodal}_t + 1.5 \text{ lift}_t). \quad (9)$$

FSM gate definition. The phase indicator m_t equals 1 when the tool centre point is sufficiently close to the canonical grasp pose (position error below 0.045 m and geodesic rotation error below 0.25 rad) for at least 6 consecutive steps. Otherwise $m_t = 0$.

Reward term definitions.

alive_t	Constant per-step survival term.
dist_t	Euclidean distance from the tool centre point to the canonical grasp target position.
success_t	Binary success signal for a lifted grasp in the GRASP phase.
centered_t	Antipodal centring score along the finger span.
near_t	Binary indicator that the tool centre point is within the near-distance threshold.
bad_close_t	Penalty for closing when the object is not near and not centred.
early_close_t	Penalty for finger closure during ALIGN.
$\text{side_contact_proxy}_t$	Penalty for lateral contact while misaligned.
push_speed_t	Penalty for rapid motion away from the target when far.
$\ a_t\ _2^2$	Residual action penalty.
approach_pos_t	Position-shaping term active during ALIGN.
approach_rot_t	Rotation-shaping term active during ALIGN.
gap_align_t	Gap-axis alignment term active during GRASP.
antipodal_t	Antipodal placement term active during GRASP.
lift_t	Lift reward active during GRASP.

Narrow residual zone. The PPO delta action is applied only when the end-effector is inside the configured narrow residual zone around the canonical grasp pose. Outside that zone, the residual is zeroed and the ACT action is applied unchanged. In the reported experiments, the non-zero residual was restricted to wrist and gripper control indices.