RADAR: Closed-Loop Robotic Data Generation via Semantic Planning and Autonomous Causal Environment Reset

Yongzhong Wang[1,*], Keyu Zhu[1,*], Yong Zhong[1,*], Liqiong Wang[1], Jinyu Yang[2,†], Feng Zheng[1,3]

***Abstract*— The acquisition of large-scale physical interaction data—a critical prerequisite for modern robot learning—is severely bottlenecked by the prohibitive cost and scalability limits of human-in-the-loop collection paradigms. To break this barrier, we introduce Robust Autonomous Data Acquisition for Robotics (RADAR), a fully autonomous, closed-loop data generation engine that completely removes human intervention from the collection cycle. RADAR elegantly divides the cognitive load into a four-module pipeline. Anchored by merely 2-5 3D human demonstrations as geometric priors, a Vision-Language Model (VLM) first orchestrates scene-relevant task generation via precise semantic object grounding and skill retrieval. Next, a Graph Neural Network (GNN) policy translates these subtasks into robust physical actions via in-context imitation learning. Following execution, the VLM performs automated success evaluation using a structured Visual Question Answering (VQA) pipeline. Finally, to shatter the bottleneck of manual resets, a Finite State Machine (FSM) orchestrates an autonomous environment reset and asymmetric data routing mechanism. Driven by simultaneous forward-reverse planning with a strict Last-In, First-Out (LIFO) causal sequence, the system seamlessly restores unstructured workspaces and robustly recovers from execution failures. This continuous brain-cerebellum synergy transforms data collection into a self-sustaining process. Extensive evaluations highlight RADAR's exceptional versatility. In simulation, our framework achieves up to 90% success rates on complex, long-horizon tasks, effortlessly solving challenges where traditional baselines plummet to near-zero performance. During real-world deployments, the system reliably executes diverse, contact-rich atomic skills—such as deformable object manipulation—via few-shot adaptation without domain-specific fine-tuning, providing a highly scalable paradigm to democratize robotic data acquisition.**

I. Introduction

Recent end-to-end embodied intelligence models have demonstrated remarkable generalization capabilities, driving significant progress in robotic manipulation [1], [2], [3], [4]. However, the scaling of such embodied intelligence is fundamentally bottlenecked by the acquisition of large-scale, high-fidelity physical interaction data. Existing solutions to this data bottleneck face a frustrating dichotomy. Simulation-based methods [5], [6], [7], [8] offer tremendous scalability but struggle with persistent sim-to-real gaps and limited behavioral diversity. Conversely, teleoperation-based methods [9], [10], [11], [12] provide high-quality demonstrations but remain prohibitively expensive and fundamentally unscalable due to the slow and serial nature of human control.

Recent advances have addressed several individual components required for autonomous robot data generation, yet these components remain largely disconnected. At the planning level, VLM-based manipulation methods infer affordances through 2D visual prompts or generated image subgoals [13], [14], [15], [16], but such representations provide limited geometric constraints for precise physical interaction. At the execution level, In-Context Imitation Learning and generative visuomotor policies enable accurate control from demonstrations [17], [18], [19], but typically assume that task specifications, execution contexts, and demonstrations are supplied externally. Meanwhile, autonomous learning frameworks [15], [20] automate parts of task proposal, evaluation, or policy improvement, but do not provide a general mechanism for restoring the physical workspace after interaction. Consequently, existing approaches do not yet form a self-sustaining collection loop that jointly supports task generation, precise execution, outcome verification, and autonomous environment reset.

To address these limitations, we introduce **R**obust **A**utonomous **D**ata **A**cquisition for **R**obotics (**RADAR**), a fully automated, closed-loop pipeline, as illustrated in Fig. 1. Rather than requiring Vision-Language Models (VLMs) to generate intermediate images or directly infer fragile 3D coordinates, or relying on disjointed heuristic scripts for environment reset, RADAR establishes a **brain–cerebellum synergy** by separating high-level cognitive reasoning from low-level physical execution. Specifically, the VLM serves as the cognitive "brain," responsible for semantic reasoning, task generation, and success evaluation, while the GNN-based policy acts as the "cerebellum," executing sub-millimeter, high-frequency control based on 3D geometric priors. Building on recent advances in In-Context Learning (ICL) [17], [21], RADAR uses a small number of human demonstrations as reusable 3D physical priors, enabling the system to generalize these demonstrations into large-scale physical execution. In this way, the framework reduces the sim-to-real gap associated with simulation, the teleoperation burden of manual data collection, and the geometric hallucination risks of direct VLM control. **As a result, RADAR enables the continuous acquisition of high-fidelity robotic data with minimal human intervention.** The complete data-generation process is organized into four modules:

(1) **Scene-Relevant Task Generation**: We employ a Vision-Language Model (VLM) to autonomously construct

*Equal contribution.
†Corresponding author.
[1]Department of Computer Science and Engineering, Southern University of Science and Technology.
[2]Harbin Institute of Technology, Shenzhen.
[3]Spatialtemporal AI.

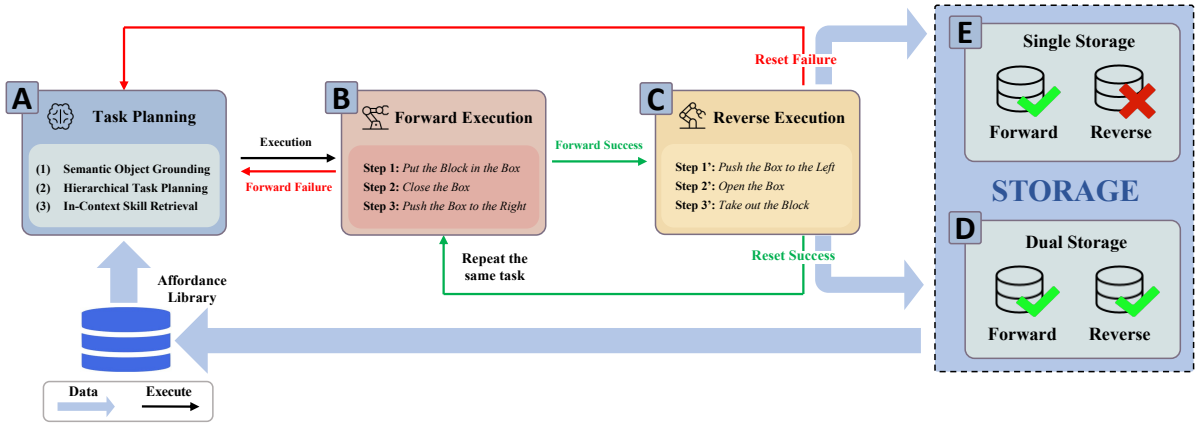

Fig. 1. Overview of the RADAR pipeline and the state transition diagram of its decoupled Finite State Machine (FSM). To ensure logical clarity, the architecture strictly separates physical execution loops (**States A, B, C**) from concurrent data routing actions (**States D, E**). A fully successful execution forms a continuous loop (**B** $\rightarrow$ **C** $\rightarrow$ **B**), concurrently triggering **Dual Storage (D)** to repeatedly harvest trajectory variations without re-planning. In contrast, an asymmetric recovery loop (**B** $\rightarrow$ **C** $\rightarrow$ **A**) bypasses reset failures by selectively saving the valid forward trajectory via **Single Storage (E)** and initiating a novel planning cycle on the altered workspace. This architecture guarantees a truly self-sustaining, human-out-of-the-loop engine.

scene-relevant tasks and extract object segmentation masks based on the current observation. Complex, long-horizon tasks are decomposed into a sequence of atomic subtasks, each matched with a relevant demonstration from a small affordance library as a behavioral prior.

(2) **Task Execution via In-Context Imitation Learning**: The robot performs the assigned subtasks using a Graph Neural Network (GNN)-based in-context imitation learning framework, which maps the selected demonstrations and current observations to executable continuous trajectories.

(3) **Automated Success Evaluation**: The VLM acts as an embodied evaluator to determine the outcome of the execution through a structured Visual Question Answering (VQA) formulation, filtering out failed trajectories.

(4) **Autonomous Environment Reset**: Upon completion of a task, a Finite State Machine (FSM) governs the system to actively compute and execute a causal inverse sequence of the forward actions. This strictly adheres to a Last-In, First-Out (LIFO) logical constraint to restore the environment, facilitating truly continuous and robust data generation **without human assistance**.

While drastically reducing reliance on human labor, RADAR maintains strong performance across both simulated and real-world deployments. In simulation, we evaluate the framework's ability to plan and execute complex, long-horizon manipulation tasks, where it achieves high success rates and robust autonomous resetting. We further validate its practical applicability through deployment on a physical robotic system. Using only one or a few visual demonstrations, RADAR successfully performs a range of challenging atomic skills, including deformable-object manipulation, such as towel folding, and high-precision alignment, such as paper-roll insertion, without requiring domain-specific fine-tuning. Taken together, these results demonstrate the robustness and versatility of our closed-loop pipeline across different domains, establishing RADAR as a scalable and domain-agnostic data-generation engine for physical robot learning.

Our main contributions are highlighted as follows:

- We introduce **RADAR**, a **fully automated, closed-loop** pipeline for real-world robot manipulation data collection. Our system requires only 2-5 manually collected atomic demonstrations and scales them into diverse, task-relevant datasets with strictly minimal human intervention.
- We propose a **scene-relevant task generation** framework, which effectively translates complex, long-horizon tasks in cluttered environments into sequentially executable atomic skills.
- We design a novel **autonomous environment reset mechanism** to achieve autonomous environment resetting. This empowers the robot with causal self-correction and scene-restoration capabilities, unlocking continuous, human-out-of-the-loop data streaming.
- **Extensive Empirical Validation:** We validate RADAR's capabilities across simulated and physical domains. Our pipeline achieves up to 90% success rates on complex, long-horizon simulated tasks, demonstrating highly robust forward planning and execution. Furthermore, real-world deployments establish the system as a powerful proof-of-concept for human-out-of-the-loop data generation, reliably executing diverse, contact-rich physical skills (*e.g.*, deformable object manipulation) via few-shot in-context learning.

II. RELATED WORK

A. Autonomous Data Collection and Evaluation

High-quality robot demonstration data is the cornerstone of robust imitation learning [9], [22]. To overcome manual collection bottlenecks, recent works explore autonomous policy improvement [20]. For instance, the SOAR framework [15] utilizes pre-trained VLMs for task proposal and success detection, employing an image-editing diffusion

model, SuSIE [16], to generate visual subgoals for a goal-conditioned policy [23].

However, SOAR and similar autonomous systems face a fundamental bottleneck: environment resets. Once a robot alters the environment, continuous collection breaks without human intervention. Furthermore, SOAR's single-stage VLM evaluation is highly susceptible to conversational redundancies and visual hallucinations, leading to false-positive success labels [15].

In contrast, our pipeline achieves true closed-loop autonomy via a Simultaneous Forward-Reverse Planning mechanism. The VLM constructs a strict Last-In, First-Out (LIFO) causal sequence to automatically restore the environment. Coupled with an asymmetric failure handling logic, unrestored scenes seamlessly become novel initial states. Moreover, we replace unreliable single-stage evaluation with a robust three-stage Vision-Question-Answering (VQA) pipeline, strictly decoupling VLM visual reasoning from deterministic logic to ensure the absolute fidelity of collected demonstrations.

B. Visual Prompting and Affordance Reasoning

Integrating VLMs into control architectures requires effective affordance representations [24], [25]. Inspired by Set-of-Mark prompting [26], MOKA [13] advanced this via a mark-based visual prompting framework, overlaying 2D candidate keypoints on RGB images to formulate affordance reasoning as a VQA problem [13], [27].

While simplifying VLM reasoning, this 2D paradigm exhibits significant geometric vulnerability. Predicting affordances entirely in 2D pixel space forces MOKA to rely on noisy depth heuristics to back-project points into 3D $SE(3)$ space [13]. Consequently, executions involving complex contact dynamics (*e.g.*, tight insertions) inevitably fail, as 2D pixels cannot encapsulate precise kinematic constraints.

Our approach firmly rejects this fragile 2D guessing. We introduce a 3D prior-based Affordance Library derived from a small set of real human demonstrations. Instead of forcing the VLM to generate coordinates from scratch, we constrain it to perform semantic object grounding and In-Context Skill Retrieval. By retrieving the most geometrically and semantically consistent 3D demonstration as a contextual prior, we shift the physical precision burden to human-demonstrated trajectories. This preserves VLM semantic generalization while achieving strictly feasible, high-fidelity kinematics.

C. In-Context Learning and Graph Diffusion

In-Context Imitation Learning (ICIL) has emerged as a promising paradigm to execute VLM-planned subtasks without exhaustive fine-tuning [19], [21]. Bridging ICIL with multimodal diffusion models [18], Instant Policy [17] formulates ICIL as a conditional graph generation problem using Graph Neural Networks [28]. It employs a graph transformer-based reverse diffusion process to iteratively denoise future action nodes [17].

While the underlying graph diffusion architecture is highly effective for low-level action generation, it fundamentally operates as an isolated execution policy. It heavily relies on human-provided task definitions and demonstrations, lacking the cognitive mechanisms required to autonomously propose novel tasks, chain long-horizon behaviors, or verify execution success. Conversely, autonomous frameworks like SOAR attempt to close this loop using image-editing diffusion models (SuSIE) to generate intermediate visual subgoals [15], [16]. However, generating intermediate pixels introduces high computational latency and severe physical hallucinations (*e.g.*, floating objects), inevitably causing catastrophic execution failures.

Our work resolves this dichotomy by embedding the graph diffusion model as a subordinate execution engine within a fully autonomous, VLM-driven closed loop. We replace error-prone pixel generation with our robust front-end hierarchical task planning, which autonomously grounds objects and translates high-level semantic goals into precise atomic contexts. Following execution, our back-end VQA evaluation strictly verifies task completion and triggers environment resets. By wrapping the sub-millimeter $\mathbb{SE}(3)$ precision of the ICIL graph architecture with these autonomous cognitive layers, we successfully elevate it from a passive execution policy into a continuous, human-out-of-the-loop data generation pipeline.

III. METHOD

As orchestrated by the continuous control flow illustrated in Fig. 1, our fully automated data collection framework operates as a self-sustaining closed loop. To ground the system's physical execution capabilities, we first formalize a foundational set of semantic skills—depicted as the vital data prior in our architecture—as an **Affordance Library**. Building upon this library, our pipeline is elegantly structured into a **four-module framework** that explicitly mirrors the brain-cerebellum division of labor: (1) Scene-Relevant Task Generation, (2) Task Execution via In-Context Imitation Learning, (3) Automated Success Evaluation, and (4) Autonomous Environment Reset.

In the following subsections, we first define the prerequisite Affordance Library, and subsequently detail the technical formulation of each of the four pipeline modules.

In the following subsections, we first define the prerequisite Affordance Library, and subsequently detail the technical formulation of each of the four pipeline phases.

A. Preliminaries: Affordance Library

To facilitate downstream in-context imitation learning policy (detailed in Section III-C), we construct an Affordance Library of manually collected demonstrations. We denote the library as L. Following instant policy [17], each demonstration $d \in L$ is a trajectory of local graph capturing the state of execution over a temporal horizon T. Formally, a demonstration d is defined as a sequence of states:

$$d = \{(P^t, \mathbf{T}^t_{WE}, g^t, \xi)\}_{t=1}^{T} \tag{1}$$

At each timestamp t, our observation at time consists of segmented point cloud P^t, the homogeneous transformation

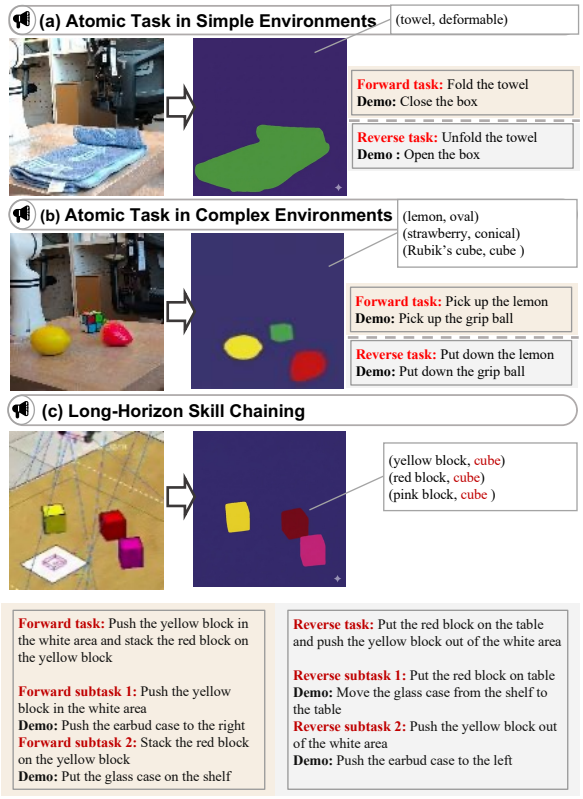

Fig. 2. Overview of our hierarchical Scene-Relevant Task Generation framework across varying complexities. **(a) Atomic Task in Simple Environments:** The VLM performs direct affordance matching, mapping a deformable object task (folding a towel) to a geometrically congruent 3D prior (closing a box). **(b) Atomic Task in Complex Environments:** Through selective attention, the planner actively masks out distractors (*e.g.*, strawberry, Rubik's cube) to precisely ground the target object (lemon) and retrieve a robust prior. **(c) Long-Horizon Skill Chaining:** For multi-step tasks, the VLM orchestrates a forward skill chain (pushing and stacking blocks) while concurrently generating a strict Last-In, First-Out (LIFO) causal sequence to autonomously construct executable environment-resetting plans.

matrix from the world frame to the end-effector frame $\mathbf{T}^t_{WE} \in \mathrm{SE}(3)$ and $g^t \in \{0, 1\}$ indicates the gripper state, where 0 and 1 represent open and closed status, respectively. Specific to simulation, we add a list of object identified by id $\xi_{sim} = \{x_1, x_2, \ldots, x_n\}$ to d for masking. Moreover, in order to dependency on ground-truth trackers, the real-world demonstrations omit these identifiers, relying instead on the VLM-based object grounding described in our task planning module.

B. Module 1: Scene-Relevant Task Generation

As illustrated in Fig. 2, to generate physically feasible and semantically meaningful tasks in unstructured environments, we propose a hierarchical visual task planning framework. To leverage the reasoning capabilities of VLM (denoted as $\mathcal{M}$), we propose a 2-stage process: (1) **semantic object grounding**, where we utilize VLM to ground relevant objects; (2) **hierarchical task planning**, where we translate visual observations into structured task plans.

1) Semantic Object Grounding: To get an accurate understanding of the scene, instead of feeding raw images directly into the planning module (which may lead to hallucinations), we first employ a VLM-based object detector to extract a structured semantic representation of the current scene. The process can be formulated as follows:

$$\xi_{scene} = \mathcal{M}(p_{ground}, s_0) \tag{2}$$

Given an RGB image of the initial scene s_0 with a grounding language prompt p_{ground}, the pipeline queries the VLM to identify all candidate objects and their geometric attributes (*e.g.*, identifying a lemon as "oval" or a strawberry as "conical", as shown in Fig. 2b). We denote this structured output as $\xi_{scene} = \{x_1, x_2, \ldots, x_n\}, x_i = (name, shape)$. This output serves as a hard constraint for subsequent planning, ensuring the robot only attempts to interact with objects physically present in the workspace.

2) Hierarchical Task Planning: We formalize the hierarchical task planning as a conditional generation problem. Given the current scene observation s and a task-level prompt p_{level}, the VLM $\mathcal{M}$ generates a structured response y:

$$y = \mathcal{M}(p_{level}, s) \tag{3}$$

where y encapsulates a set of task-relevant objects ξ_{mask} for precise visual grounding and a sequence of executable subtasks $\mathcal{T} = \{(a_t, d_t, c_t)\}_{t=1}^{T}$. For each step t, the VLM planner identifies an atomic action a_t, retrieves the most semantically and geometrically congruent demonstration d_t from the affordance library L, and provides a textual description c_t to guide the downstream imitation learning policy.

Based on scene complexity and task horizon, our planner dynamically adapts across three scenarios, as comprehensively detailed in Fig. 2: 1) *Atomic Task in Simple Environments*: For uncluttered scenes, the VLM performs **Direct Affordance Matching** (Fig. 2a). For example, it maps the task of "folding a deformable towel" directly to the geometrically similar demonstration of "closing a box", thereby generating its reverse task ("unfolding the towel") based on the "opening a box" demonstration. 2) *Atomic Task in Complex Environments*: When distractors are present, a specific prompt enforces **Selective Attention** (Fig. 2b). The VLM identifies a target subset ξ_{mask} (*e.g.*, focusing purely on the target lemon while actively ignoring the strawberry and Rubik's cube) to prevent hallucinated interactions, retrieving the "pick up grip ball" skill as a robust 3D prior. 3) *Long-Horizon Skill Chaining*: For multi-step tasks, the VLM logically chains atomic skills (Fig. 2c) without decomposing them into smaller primitives. The system prompt enforces physical causality and the simultaneous generation of environment-resetting plans. For example, generating a forward sequence ("push the yellow block into the white area", "stack the red block") concurrently produces a strict Last-In, First-Out (LIFO) constrained reverse sequence ("put the red block on the table", "push the yellow block out"). This explicitly ensures the multi-step plan is both physically executable and inherently self-reversible.

3) In-Context Skill Retrieval via Affordance Library: To bridge high-level planning with low-level control, both atomic and long-horizon planning modes utilize an in-context retrieval mechanism. For every generated subtask, the VLM estimates a similarity rank based on action semantics and object geometry, returning an ordered list of reference demonstrations $\mathcal{D} = \{d_1, d_2, ..., d_r\}$. Specifically, our prompting framework instructs the VLM to evaluate similarities across two distinct dimensions: **Action Similarity** (*e.g.*, ensuring the motion trajectories of folding and closing align) and **Geometric/Functional Similarity** (*e.g.*, recognizing that a "lemon" is geometrically congruent to an oval "grip ball"). This dual-criteria retrieval eliminates the need for hallucinating arbitrary 3D waypoints, allowing the downstream imitation learning policy to leverage robust geometric priors for zero-shot generalization to novel objects.

C. Module 2: Task Execution via In-Context Imitation Learning

Inspired by Instant Policy [17], RADAR employs graph-based In-Context Imitation Learning (ICIL) to execute novel subtasks without task-specific fine-tuning. Given the demonstrations retrieved by the planning module, the policy jointly represents the demonstration context $\mathcal{G}_c$, the current segmented point-cloud observation $\mathcal{G}_l^t$, and the predicted future actions $\mathcal{G}_l^a(a)$ as a heterogeneous graph:

$$\mathcal{G}^k = \mathcal{G}\left(\mathcal{G}_l^a(a^k), \mathcal{G}_c, \mathcal{G}_l^t\right). \tag{4}$$

The demonstration graphs encode object geometry, end-effector poses, gripper states, and temporal motion, providing the policy with a 3D behavioral prior for the current task.

Action generation is formulated as a conditional reverse-diffusion process. Starting from Gaussian noise $a^K \sim \mathcal{N}(0, I)$, a heterogeneous graph transformer predicts the denoising direction for the future action nodes while keeping the demonstration and observation context fixed. One reverse step is written as

$$\mathcal{G}^{k-1} = \mathcal{G}\Big(\mathcal{G}_l^a\big(\alpha_k(a^k - \gamma_k \epsilon_\theta(\mathcal{G}^k, k)) + \sigma_k \mathbf{z}\big), \mathcal{G}_c, \mathcal{G}_l^t\Big), \tag{5}$$

where $\mathbf{z} \sim \mathcal{N}(0, I)$ and α_k, γ_k, and σ_k are determined by the diffusion schedule. The predicted gripper-keypoint updates are converted into valid $\mathrm{SE}(3)$ end-effector transformations through rigid alignment. After K denoising steps, the resulting action a^0 is executed, and the policy is repeatedly queried with updated observations to provide closed-loop control.

D. Module 3: Automated Success Evaluation

To close the loop of our autonomous data collection pipeline, it is imperative to evaluate the outcome of the executed policy without requiring human-in-the-loop verification. To this end, we introduce an **Automated Success Evaluation** module. This module leverages a Vision-Language Model (VLM) to visually inspect the post-execution workspace and determine whether the physical state aligns with the semantic goal of the commanded subtask.

To mitigate the inherent variability and conversational verbosity of raw VLM outputs, we formulate the success detection as a structured, three-stage Visual Question Answering (VQA) pipeline: (1) Semantic Task-to-Query Translation, (2) Vision-Language Assessment, and (3) Robust Boolean Decoding.

1) Semantic Task-to-Query Translation: Standard VLMs often struggle to evaluate imperative task commands (*e.g.*, "put the yellow ball on the blue plate") directly. To optimize the VLM's reasoning, we first translate the task description c_t into a targeted, interrogative VQA query q_{vqa}.

We employ a Large Language Model (LLM), denoted as $\mathcal{M}_{trans}$, guided by a few-shot in-context prompt containing diverse manipulation examples (*e.g.*, mapping "move the red object from the cloth to the table" to "Is the red object on the cloth or the table?"). This process translates the action-oriented command into a state-oriented visual query:

$$q_{vqa} = \mathcal{M}_{trans}(c_t, p_{vqa}) \tag{6}$$

where p_{vqa} represents the few-shot prompt template designed to elicit specific spatial and relational questions.

2) Vision-Language Assessment: Following the execution of the policy over the temporal horizon T, the robot captures the final visual observation of the workspace, denoted as s_T. The generated query q_{vqa} and the image s_T are subsequently fed into a state-of-the-art VLM (*e.g.*, GPT-4V or CogVLM), denoted as $\mathcal{M}_{vlm}$. The VLM acts as an embodied evaluator, analyzing the spatial relationships and object states within s_T to generate a raw textual assessment r_{vlm}:

$$r_{vlm} = \mathcal{M}_{vlm}(s_T, q_{vqa}) \tag{7}$$

3) Robust Boolean Decoding: Since the raw response r_{vlm} from the VLM may contain auxiliary reasoning or conversational fillers (*e.g.*, "Yes, I can see that the object is..." instead of a strict boolean), directly parsing this output is error-prone. To ensure a deterministic control flow for our autonomous pipeline, we introduce a final decoding step.

We utilize a parsing LLM $\mathcal{M}_{parse}$ to distill the verbose evaluation into a strict binary success signal $b_{succ} \in \{True, False\}$. The parser is conditioned on the original task c_t, the generated query q_{vqa}, and the VLM's response r_{vlm}:

$$b_{succ} = \mathcal{M}_{parse}(c_t, q_{vqa}, r_{vlm}) \tag{8}$$

Instead of relying on heuristic error handling, this extracted binary signal b_{succ} directly serves as the deterministic trigger for state transitions in our downstream pipeline. These signals dictate whether the system advances to the reverse resetting module or aborts to initiate a new planning cycle, as detailed in Section III-E. Furthermore, these discrete success metrics are logged to maintain historical task statistics, enabling the system to track the reliability of specific skills over time.

E. Module 4: Autonomous Environment Reset

To achieve truly continuous and autonomous data collection, minimizing human intervention between episodes is

critical. We introduce an **Autonomous Environment Reset** mechanism, which leverages the reasoning capabilities of the VLM to automatically restore the workspace to its initial state s_0 following the completion of a generated task. Instead of relying on hard-coded or heuristic reset scripts, our framework formulates the environment reset as an inverse task planning problem tightly coupled with the forward generation.

1) Simultaneous Forward-Reverse Planning: During the hierarchical task planning phase, the VLM is prompted to act as a causal reasoning engine. It simultaneously generates the primary executable plan (denoted as the *forward task*) and its exact causal inverse (denoted as the *reverse task*).

For atomic tasks, the VLM directly infers the inverse affordance based on the object's geometric properties. For example, if the forward atomic action is "pick up the cup", the VLM proposes "place the cup down" as the reverse task and retrieves the most semantically congruent demonstration from the affordance library L.

For long-horizon multi-step tasks, the environment reset strictly adheres to a Last-In, First-Out (LIFO) causal sequence constraint. Given a forward plan consisting of N subtasks $\mathcal{T}_{fwd} = \{(a_i^f, d_i^f, c_i^f)\}_{i=1}^N$, the VLM constructs a reverse plan $\mathcal{T}_{rev} = \{(a_j^r, d_j^r, c_j^r)\}_{j=1}^N$. Crucially, each reverse subtask at step j must explicitly undo the physical state changes introduced by the forward subtask at step $i = N - j + 1$. This ensures physical feasibility during the restoration phase (*e.g.*, a box must be opened before the cube placed inside it can be extracted).

2) Continuous Collection via Finite State Machine: To seamlessly integrate physical actions and data logging into a continuous, human-out-of-the-loop pipeline, we orchestrate the process using a Finite State Machine (FSM), as visualized in Fig. 1. To ensure logical clarity, our FSM explicitly decouples the physical execution states—Task Planning (**State A**), Forward Execution (**State B**), and Reverse Execution (**State C**)—from the concurrent data routing actions—Dual Storage (**State D**) and Single Storage (**State E**).

Governed by the binary success signal b_{succ} from the VQA evaluator, the system executes the following operational loops, with data storage acting as triggered side-effects:

- **Continuous Success Loop (B $\rightarrow$ C $\rightarrow$ B):** If both the forward task and the reverse reset succeed, the physical environment is perfectly restored. The control flow seamlessly loops from State C directly back to State B to **repeatedly execute the same assigned task**. Concurrently, this successful cycle triggers the **Dual Storage (State D)** mechanism, validating and saving both the forward and reverse trajectories. This cyclic execution allows the system to continuously harvest diverse trajectory variations of a specific skill without the computational overhead of VLM re-planning.
- **Asymmetric Recovery Loop (B $\rightarrow$ C $\rightarrow$ A):** If the forward task succeeds but the reverse reset fails, the physical environment is left un-restored. To recover, the control flow forcibly breaks the loop and routes back to Task Planning (State A), treating the altered workspace as a novel initial scene. Concurrently, this transition triggers the **Single Storage (State E)** mechanism, selectively retaining the perfectly valid forward trajectory while discarding the failed reset attempt.
- **Forward Abort (B $\rightarrow$ A):** If the initial forward execution fails, the invalid trajectory is immediately discarded (no storage triggered). The control flow aborts the current execution queue and transitions back to State A to re-observe and re-plan.

As depicted in the global architecture (Fig. 1), this state-decoupled orchestration guarantees the continuous streaming of high-fidelity data into the Affordance Library. By strictly separating the execution loops from the asymmetric data routing, the pipeline effectively bypasses reset failures and eliminates the need for manual scene restoration, transforming it into a truly self-sustaining engine.

IV. EXPERIMENTS

A. Experimental Setup

We evaluate our automated pipeline in the RLBench simulation [29]. We select 7 atomic tasks and 3 complex long-horizon tasks, comparing our system against two state-of-the-art vision-language manipulation baselines: MOKA [13] and ReKep [14].

Implementation Details & Evaluation Protocol: To rigorously isolate and evaluate the forward task generation and execution capabilities, the environments are reset using simulation ground truth between rollouts. We report the success rate over 10 independent rollouts per task with randomized initial states. Empirically, we provide a single high-quality demonstration (1-shot) as the context, as increasing demonstrations did not yield proportional gains. Furthermore, we utilized a VLM instead of CLIP for skill retrieval, as our preliminary tests showed CLIP embeddings are heavily noun-biased and fail to distinguish fine-grained actionable semantics.

TABLE I

SUCCESS RATES IN RLBENCH OVER 10 ROLLOUTS PER TASK.

Type	Task	ReKep	MOKA	Ours
Atomic	Large Container (Cup)	2/10	2/10	**8/10**
	Large Container (Block)	0/10	3/10	**8/10**
	Large Container (Laptop)	0/10	1/10	**9/10**
	Push Block	4/10	4/10	**10/10**
	Stack Block	4/10	1/10	**8/10**
	Close Box	4/10	3/10	**10/10**
	Open Box	2/10	2/10	**7/10**
Long-horizon	Put Laptop & Cup into Tray	1/10	0/10	**8/10**
	Push & Stack Blocks	0/10	0/10	**4/10**
	Close then Open Box	2/10	1/10	**9/10**

B. Main Results

As shown in Table I, our pipeline significantly outperforms ReKep and MOKA. While baselines exhibit moderate performance on simple atomic tasks, their success rates plummet to near-zero in long-horizon scenarios (*e.g.*, "Push

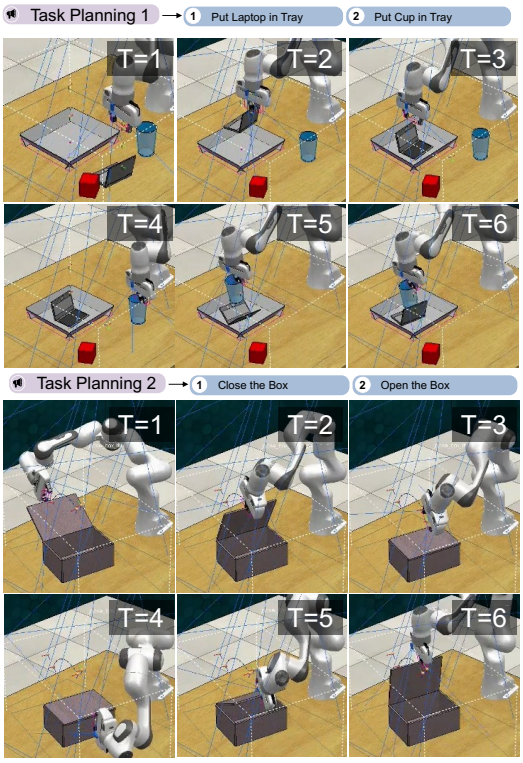

Fig. 3. Visualizations of long-horizon tasks in the RLBench simulation. **(a) Put Laptop & Cup into Tray:** The robot successfully executes a multi-step sequence requiring interactions with multiple distinct objects. **(b) Close then Open Box:** The pipeline reliably performs state-dependent articulated actions, demonstrating its robust skill chaining capability.

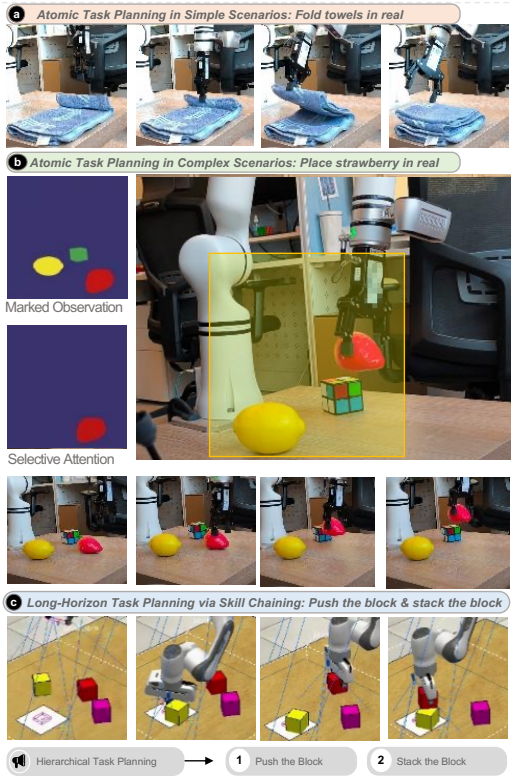

Fig. 4. Qualitative results of our automated data collection pipeline across different scenarios. **(a) Simple Atomic Scenario:** The robot executes deformable object manipulation (folding a towel) in a real-world setting without distractors. **(b) Complex Atomic Scenario:** The pipeline utilizes selective attention to isolate a target object (a strawberry) among visual distractors for precise grasping. **(c) Long-Horizon Scenario:** In simulation, the VLM decomposes a complex instruction into a sequential skill chain (*e.g.*, first pushing, then stacking a block).

& Stack Blocks", as illustrated in Fig. 4c). In contrast, our system maintains robust performance, achieving 80%-90% success rates on demanding multi-step tasks. Specifically, the pipeline effectively orchestrates sequential skill chains and dependent articulated actions (*e.g.*, opening a previously closed box, as visualized in Fig. 3), proving its capability to autonomously gather high-quality data for complex behaviors.

C. Ablation: The Necessity of Point Cloud Masking

TABLE II
ABLATION ON POINT CLOUD MASKING

Task	W/o Masking	Ours (With Masking)
Large Container (Cup)	0.10	**0.80**
Large Container (Block)	0.00	**0.80**
Push Block	0.00	**1.00**

To evaluate our VLM-driven selective attention, we ablate the masking module and feed raw scene point clouds into the execution policy. As shown in Table II, removing semantic masking causes catastrophic failures, with success rates for grasping and pushing dropping to near zero due to susceptibility to distractors. This highlights that explicit semantic masking is crucial for execution robustness in cluttered scenes (a real-world qualitative example is depicted in Fig. 4b).

D. Real-World Deployment and Limitations

To validate practical applicability, we deployed RADAR on a physical Realman RM65-B arm with a RealSense D435i camera, using SAM [30] and XMem++ [31] for real-time 3D object segmentation.

Qualitative Feasibility: Operating under a 1-shot adaptation paradigm without domain-specific fine-tuning, the robot successfully executed intricate tasks. In simple environments, it handled deformable object manipulation (*e.g.*, folding a towel, Fig. 4a). In complex scenarios, it utilized selective attention for precise grasping among distractors (*e.g.*, targeting a strawberry, Fig. 4b). These qualitative results strongly indicate RADAR's immense potential as a scalable, human-out-of-the-loop engine for physical robot learning.

Limitations and Future Work (The Reset Challenge): While our pipeline demonstrates the feasibility of autonomous data generation, fully 100% reliable environment resetting remains an open challenge. Probabilistically, chaining forward execution with a causal reverse reset inherently compounds failure rates ($p_{total} \approx p_{forward} \times p_{reverse}$). Currently, our FSM acts as a robust proof-of-concept for simple-to-moderate scenes. Overcoming this compounding error in highly unstructured environments—perhaps via

multi-modal tactile feedback or high-frequency visual servoing—represents an exciting frontier for future work.

V. Conclusion

We present **R**obust **A**utonomous **D**ata **A**cquisition for **R**obotics (**RADAR**), a self-sustaining, human-out-of-the-loop data generation engine. Orchestrated by a decoupled Finite State Machine (FSM), RADAR achieves a seamless brain-cerebellum synergy by coupling VLM-based cognitive planning and LIFO-constrained autonomous environment resetting with the sub-millimeter precision of GNN-based in-context imitation learning. Anchored by 3D human demonstrations, it circumvents the geometric hallucinations of purely 2D pipelines, while its asymmetric data routing guarantees continuous data streaming despite execution failures.

RADAR achieves up to 90% success on complex simulated tasks and reliably executes contact-rich physical skills via few-shot adaptation without domain-specific fine-tuning. Looking ahead, our future work will proceed along two exciting frontiers: tackling the open challenge of compounding reset errors via multi-modal sensory integration, and leveraging the generated high-fidelity datasets to train downstream foundational visuomotor policies (*e.g.*, Diffusion Policy) for dynamic real-world environments.

References

[1] K. Black *et al.*, "π_0: A vision-language-action flow model for general robot control," *arXiv preprint arXiv:2410.24164*, 2024.

[2] P. Intelligence, K. Black, *et al.*, "$\pi_{0.5}$: a vision-language-action model with open-world generalization," *arXiv preprint arXiv:2504.16054*, 2025.

[3] S. Liu *et al.*, "Rdt-1b: a diffusion foundation model for bimanual manipulation," in *The Thirteenth International Conference on Learning Representations*, 2024.

[4] Y. Yang, S. Zeng, T. Lin, X. Chang, D. Qi, J. Xiao, H. Liu, R. Chen, Y. Chen, D. Huo, F. Xiong, X. Wei, Z. Ma, and M. Xu, "ABot-M0: VLA foundation model for robotic manipulation with action manifold learning," *arXiv preprint arXiv:2602.11236*, 2026.

[5] M. Dalal, A. Mandlekar, C. R. Garrett, A. Handa, R. Salakhutdinov, and D. Fox, "Imitating task and motion planning with visuomotor transformers," in *Conference on Robot Learning*. PMLR, 2023, pp. 2565–2593.

[6] Y. Wang, Z. Xian, F. Chen, T.-H. Wang, Y. Wang, K. Fragkiadaki, Z. Erickson, D. Held, and C. Gan, "Robogen: towards unleashing infinite data for automated robot learning via generative simulation," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 51 936–51 983.

[7] 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 *Conference on Robot Learning*. PMLR, 2023, pp. 1820–1864.

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

[9] A. Brohan *et al.*, "Rt-1: Robotics transformer for real-world control at scale," *Robotics: Science and Systems XIX*, 2023.

[10] F. Ebert, Y. Yang, K. Schmeckpeper, B. Bucher, G. Georgakis, K. Daniilidis, C. Finn, and S. Levine, "Bridge data: Boosting generalization of robotic skills with cross-domain datasets," in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, 6 2022.

[11] A. Brohan *et al.*, "Do as i can, not as i say: Grounding language in robotic affordances," in *Conference on Robot Learning*. PMLR, 2023, pp. 287–318.

[12] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence, "Interactive language: Talking to robots in real time," *IEEE Robotics and Automation Letters*, 2023.

[13] K. Fang, F. Liu, P. Abbeel, and S. Levine, "Moka: Open-world robotic manipulation through mark-based visual prompting," in *Robotics: Science and Systems (RSS)*, 2024.

[14] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "Rekep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2025, pp. 4573–4602.

[15] Z. Zhou, P. Atreya, A. Lee, H. R. Walke, O. Mees, and S. Levine, "Autonomous improvement of instruction following skills via foundation models," in *Conference on Robot Learning*. PMLR, 2025, pp. 4805–4825.

[16] K. Black, M. Nakamoto, P. Atreya, H. R. Walke, C. Finn, A. Kumar, and S. Levine, "Zero-shot robotic manipulation with pre-trained image-editing diffusion models," in *The Twelfth International Conference on Learning Representations*, 2023.

[17] V. Vosylius and E. Johns, "Instant policy: In-context imitation learning via graph diffusion," in *The Thirteenth International Conference on Learning Representations*, 2024.

[18] C. Chi, S. Feng, Y. Du, Z. Xu, E. Cousineau, B. Burchfiel, and S. Song, "Diffusion policy: Visuomotor policy learning via action diffusion," in *Proceedings of Robotics: Science and Systems (RSS)*, 2023.

[19] V. Jain, M. Attarian, N. J. Joshi, A. Wahid, D. Driess, Q. Vuong, P. R. Sanketi, P. Sermanet, S. Welker, C. Chan, I. Gilitschenski, Y. Bisk, and D. Dwibedi, "Vid2robot: End-to-end video-conditioned policy learning with cross-attention transformers," in *Proceedings of (RSS) Robotics Science and Systems*. Proceedings of Robotics: Science and Systems, May 2024.

[20] K. Bousmalis *et al.*, "Robocat: A self-improving generalist agent for robotic manipulation," *Transactions on Machine Learning Research*, 2024. [Online]. Available: https://openreview.net/forum?id=vsCpILiWHu

[21] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn, "Bc-z: Zero-shot task generalization with robotic imitation learning," in *Conference on Robot Learning*. PMLR, 2022, pp. 991–1002.

[22] J. J. Lim, "Open x-embodiment: Robotic learning datasets and rt-x models," in *IEEE International Conference on Robotics and Automation*. IEEE, 2024.

[23] M. Andrychowicz, F. Wolski, A. Ray, J. Schneider, R. Fong, P. Welinder, B. McGrew, J. Tobin, O. Pieter Abbeel, and W. Zaremba, "Hindsight experience replay," *Advances in neural information processing systems*, vol. 30, 2017.

[24] J. J. Gibson, "The theory of affordances," *Hilldale, USA*, vol. 1, no. 2, pp. 67–82, 1977.

[25] L. Manuelli, W. Gao, P. Florence, and R. Tedrake, "kpam: Keypoint affordances for category-level robotic manipulation," in *International Symposium on Robotics Research (ISRR)*, 2019.

[26] J. Yang *et al.*, "Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v," *arXiv preprint arXiv:2310.11441*, 2023.

[27] S. Liu *et al.*, "Grounding dino: Marrying dino with grounded pre-training for open-set object detection," in *European conference on computer vision*. Springer, 2024, pp. 38–55.

[28] T. Wang, R. Liao, J. Ba, and S. Fidler, "Nervenet: Learning structured policy with graph neural networks," in *International Conference on Learning Representations (ICLR)*, 2018.

[29] S. James, Z. Ma, D. R. Arrojo, and A. J. Davison, "Rlbench: The robot learning benchmark & learning environment," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3019–3026, 2020.

[30] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[31] M. Bekuzarov, A. Bermudez, J.-Y. Lee, and H. Li, "Xmem++: Production-level video segmentation from few annotated frames," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 635–644.