2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

From Manual to Operation: A Home Appliance Agent

Bo Mao†, Yuming Huang†, Jiayang Chai†, Huaping Liu, Di Guo*

***Abstract*— Operating household appliances by reading and understanding user manuals remains a fundamental and challenging problem in robotics. Recent works leverage large language models (LLMs) and vision-language models (VLMs) to interpret manuals, improving appliance operation success. However, these approaches fail when manuals are unavailable or incomplete. In this paper, we introduce an autonomous assistant for robotic appliance operation, built upon an LLMs/VLMs-powered multi-agent collaborative framework. Our system can read, comprehend, and summarize manuals, autonomously infer operational logic, and execute actions on appliances with a robotic arm. Importantly, for unseen appliances without manuals, it can acquire operational knowledge from generalized manuals and on-demand web search. Extensive evaluations on over one thousand tasks show that our framework substantially outperforms baselines and achieves robust performance in simulation and real-world experiments.**

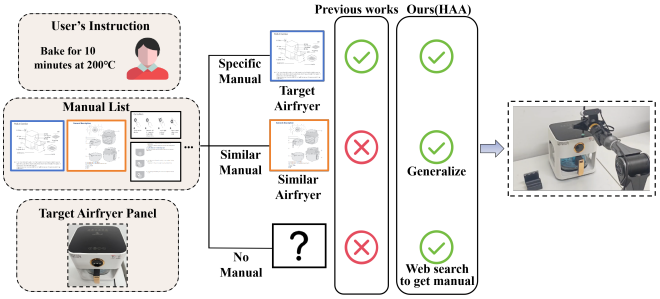


Fig. 1: In addition to operating appliances with matched manuals, the framework can also handle appliances without an available manual by acquiring the requisite knowledge through category-level generalization and on-demand web search. This capability is not present in prior work.

I. Introduction

Recent advances in foundation models, including large language models (LLMs) and vision–language models (VLMs), have enabled robots to follow natural language instructions and perform high-level planning in household environments [1]. Prior work [2], [3] has leveraged spatial constraints to achieve general-purpose manipulation, avoiding the need for task-specific skill training. However, robots still encounter difficulties in appliance-specific operations, where they must map diverse icons and controls to their corresponding functions and reliably execute multi-step procedures [4]. To address this challenge, manuals provide structured textual and visual guidance, offering a promising pathway for robots to acquire the operational logic of new appliances. Nonetheless, three key challenges remain: (i) accurately parsing the textual and visual structure of manuals and grounding it to corresponding physical controls; (ii) transferring knowledge across similar appliances; and (iii) perceiving states accurately to support reflective execution.

Researchers have explored methods like [4]–[6] for robots to learn from appliance manuals much like humans do. A manual-based benchmark dataset CheckManual [4] is introduced to evaluate and standardize progress. Notably, the ApBot system [6] demonstrates that parsing and understanding an appliance's user manual via VLMs can markedly improve success rates in operating that appliance. However, a major limitation of prior work is the dependence on having the exact manual for each appliance. In practice, an appliance's manual may be missing, inaccessible, or simply not provided to the robot. Under such circumstances, most approaches fail to generalize to a new appliance. This reliance on appliance-specific manuals severely limits the autonomy and generality of home-assistant robots, as they cannot handle unseen appliances beyond their training or prior data.

In this paper, we propose Home Appliance Agent (HAA), a novel framework designed to overcome these limitations and enable robust appliance operation across both seen and unseen appliances, as illustrated in Fig. 1. HAA is a multi-agent collaborative system powered by LLMs and VLMs, integrating specialized agents for perception, state estimation, planning, decision-making, and reflection. By leveraging LLMs and VLMs, the agents can read and comprehend available appliance manuals, summarize key operating procedures, and collaboratively plan the robot's actions. Crucially, HAA does not require a pre-existing matched manual for every appliance. When a matched manual is unavailable, the system first generalizes from existing manuals of the most similar category. If similarity is insufficient for reliable transfer, a web-search module retrieves the missing operational information. This hybrid strategy of manual summarization and web-based information gathering enables the robot to generalize to entirely novel appliances, maintaining high success rates even without appliance-specific manuals.

The contributions of the paper are summarized as follows.

- **Multi-Agent Framework**: We develop a collaborative multi-agent architecture integrating perception, state-estimation, planning, execution, and reflection agents for end-to-end robotic appliance operation, all guided

Bo Mao, Yuming Huang, Jiayang Chai and Di Guo are with the School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, China.

Huaping Liu is with the Department of Computer Science and Technology, Tsinghua University, Beijing, China, and also with Beijing National Research Center for Information Science and Technology, China.

†Bo Mao, Yuming Huang and Jiayang Chai contribute equally to this work.

*Corresponding author: guodi.gd@gmail.com.

by foundation models for reasoning and control.

- **Generalized Manual Summarization**: A novel manual-summarization module automatically constructs category-level generalized manuals from multiple appliance manuals, enabling cross-appliance knowledge transfer and zero-shot generalization to new appliances without requiring a matched manual.
- **Comprehensive Evaluation**: We conduct extensive evaluations on public benchmarks (including the CheckManual dataset), additional self-collected appliance manuals, and real-world household appliances. The results demonstrate significantly improved generalization and execution success over baseline methods, validating the effectiveness of our approach in diverse and practical settings.

II. Related Work

A. Foundation Models for Robotic Appliance Operation

Foundation models have recently gained significant traction in robotics, providing a new paradigm for task understanding and high-level planning in household environments [7]. However, these models still struggle with appliance-specific operations. LLMs can plan coarse-grained high-level tasks based on real-world constraints by integrating value functions [8], incorporating perception feedback [9], generating policy code [10], and optimizing sequences under geometric constraints [11]. But, such approaches require separate policy training for each individual control element on every appliance panel. Subsequent research has advanced general-purpose manipulation frameworks that do not rely on task-specific skill training, including building 3D value maps [12], extracting spatial geometry of object parts for constraint formulation [13], clustering keypoints to define spatial constraints [1], leveraging VLMs to select keypoints [2], and dynamically choosing spatial representations according to tasks [3]. Nevertheless, these methods can only operate based on commonsense knowledge when facing specific appliances, making it difficult to map functions to icons correctly, which often results in failures.

To mitigate this discrepancy, researchers have explored manual-based robotic manipulation, inspired by how humans consult appliance manuals to learn novel appliances. Early studies concentrated on assembly tasks [14], [15], while appliance-focused work mainly targeted button detection [16], [17] and physical execution of presses [18], [19]. More recently, manual-guided planning has gained traction. The approach [5] generates operation plans through hand-crafted pipelines, but lacks generalization and proves difficult to adapt to new appliances. The new study [4] introduces the first manual-based appliance operation dataset CheckManual, and the first manual-based appliance task planning model ManualPlan, which leverages LLMs to interpret manuals. Another similar work is ApBot [6], which demonstrates that parsing and interpreting user manuals with VLMs can markedly improve task success rates. Nevertheless, both ManualPlan and ApBot rely on paired manuals and fail to generalize when manuals are missing, which is common in real-world deployments. In contrast, our framework explicitly leverages category-level manuals and on-demand web knowledge to generalize to unseen appliances, enabling robust appliance-specific operation beyond what previous approaches can achieve.

B. Agent Systems

Agents based on LLM and VLM have recently established a new paradigm for tackling complex tasks. Single agent systems typically rely on a single LLM or VLM that performs all tasks and employ various techniques to enhance the agent's capabilities, including Chain-of-Thought (CoT) [20] prompting to elicit intermediate reasoning steps, ReAct [21] to interleave reasoning with actions, and Reflexion [22] to incorporate self-reflection loops for error correction.

Despite these advancements, single-agent approaches often result in information loss and reasoning failures throughout the task pipeline and struggle with complex multi-step tasks. To overcome these limitations, recent studies have increasingly adopted multi-agent collaborative frameworks, in which agents coordinate perception, decision-making, and reflective reasoning. Mobile-Agent series [23]–[25] use a trio of LLM-based multi-agent (a planner, a decision-maker, and a reflection verifier) to navigate mobile user interfaces (UIs), with high-level task planning separated from low-level UI execution via dedicated agents. This role-based design allows one agent to formulate a strategy and others to carry it out, often with a "reflection" agent monitoring and refining the plan [26], [27].

However, in robotics, most LLM/VLM-based frameworks have remained single-agent, with existing robot controllers [1], [12] typically relying on a single centralized LLM or VLM for all decision-making. Only a few studies explore coordination among multiple LLMs and VLMs in embodied robotic settings [28]. To address this gap, we extend multi-agent LLM/VLM-based collaboration to real-world robotic control of home appliances. Our work demonstrates that multiple agents can collectively perceive through vision, plan actions, and execute actions in a zero-shot setting, communicating via shared memory and iterative self-reflection to achieve coordinated manipulation of home appliances.

III. Methodology

A. Problem Formulation

Given a high-level natural language instruction I (e.g., "set 180°C and bake for 30 minutes"), an image of the appliance's control panel $\mathcal{IM}_0$ and an available manual $\mathcal{MA}$, the robot learns to operate the appliance and executes a sequence of low-level physical actions $a_{1:T} = \{a_1, \ldots, a_T\}$, where each $a_t \in \mathcal{A}$, such that the appliance is configured to satisfy the instruction I. The action space $\mathcal{A}$ encompasses all atomic manipulations the robot can perform on the appliance, including pressing buttons, turning knobs, or other discrete interactions required for control.

At time step $t = 0$, the robot attempts to obtain an available manual $\mathcal{MA}$ of the target appliance based on I and $\mathcal{IM}_0$. At each time step t, the robot extracts a structured representation

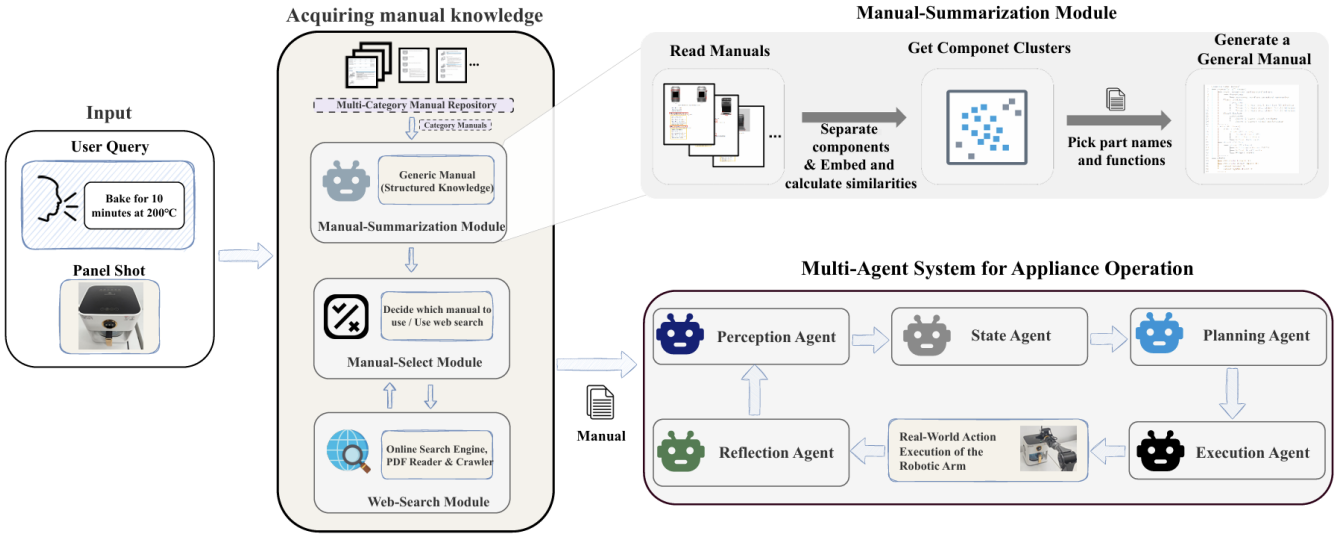


Fig. 2: Illustration of HAA framework, which comprises a manual knowledge acquisition module and a multi-agent appliance-operation module; the inset details the manual-summarization workflow to construct category-level generalized manuals.

of the panel $\mathcal{PA}_t$ and forms the observation $o_t \in \mathcal{O}$ from the raw visual input $\mathcal{IM}_t$ and the available manual $\mathcal{MA}$. Then, the robot infers the appliance state $s_t \in \mathcal{S}$ and selects an atomic action $a_t \in \mathcal{A}$. Executing a_t changes the appliance to a new state s_{t+1}, which yields the next panel image $\mathcal{IM}_{t+1}$. Also, $\mathcal{PA}_{t+1}$ and o_{t+1} are updated. This loop continues until the instruction I is satisfied or the robot determines that the task has been completed. Specifically, the available manual $\mathcal{MA}$ may be an appliance-specific manual (i.e., a manufacturer-provided manual), generalized from similar appliances, or retrieved online.

B. Manual Knowledge Acquisition

The system maintains a manual pool $\mathcal{MR}$ that contains two resources: (i) appliance-specific manuals (original manuals) $\{\mathcal{M}_1, \ldots, \mathcal{M}_l\}$, corresponding to a total of l distinct home appliances across different categories and (ii) generalized manuals $\{\mathcal{MG}^{(1)}, \ldots, \mathcal{MG}^{(n)}\}$ distilled offline by the manual-summarization module from multiple manuals within n categories. At the beginning of a task, given a natural language instruction I and the initial control-panel image $\mathcal{IM}_0$, the system selects a task-adequate manual $\mathcal{MA}$ from $\mathcal{MR}$. The selection follows a set of priorities. If an appliance-matched original manual or a category-matched generalized manual is available, it is used directly. Otherwise, the system retrieves substitute candidates that prioritize manuals whose sets of controls and functions overlap with those detected on the target appliance, and functions are sufficient to execute the current instruction. If neither resource is available, the web-search module is invoked to obtain an appliance-specific manual. The selected manual is then passed to the downstream execution system to guide the multi-agent home appliance operation.

The manual-summarization module produces generalized manuals by aggregating and abstracting common functional knowledge across multiple manuals of the same category, as shown in Fig. 2. Specifically, the module extracts functional component names, usage descriptions, and exemplar tasks, and represents them with semantic embeddings. Clustering is then applied to group semantically related items from different manuals of one category into unified function entries [29], [30]. Each cluster represents a function label or task label. The module consolidates the labels in the clusters and associated descriptions into a concise specification, yielding a category-level generalized manual $\mathcal{MG}^{(c)}$ that is stored back into $\mathcal{MR}$. When the manual pool cannot provide a usable manual due to the low similarity, the web-search module generates targeted queries from $\mathcal{IM}_0$ and I, retrieves relevant web pages or PDF documents, and extracts part labels and stepwise operating procedures. The resulting appliance-specific manual $\mathcal{MA}_{web}$ is normalized and cached for future reuse.

C. Multi-Agent Home Appliance Operation

While the manual knowledge acquisition module provides the agent with a task-adequate manual $\mathcal{MA}$, we design a multi-agent home appliance operation module that integrates five specialized agents in a closed loop to transform the knowledge into actions, as illustrated in Fig. 3. The module takes as input the current instruction I, the panel image $\mathcal{IM}_t$, the selected manual $\mathcal{MA}$, and historical context from the memory unit. Then, the module outputs the low-level action commands a_t for the robotic arm.

1) Memory unit: The memory unit underpins the framework by storing both a long-term repository of manual knowledge $\mathcal{MR}$ and a short-term working memory. Before time step t, the short-term memory maintains the current user instruction I, historical planning information $\mathcal{PL}_{1:t-1}$, historical states $s_{1:t-1}$, and executed actions $a_{1:t-1}$. During task execution, short-term memory entries are continuously updated and reset after each instruction is completed. These memories are repeatedly consumed by the multi-agent home appliance operation module, ensuring robust and stable

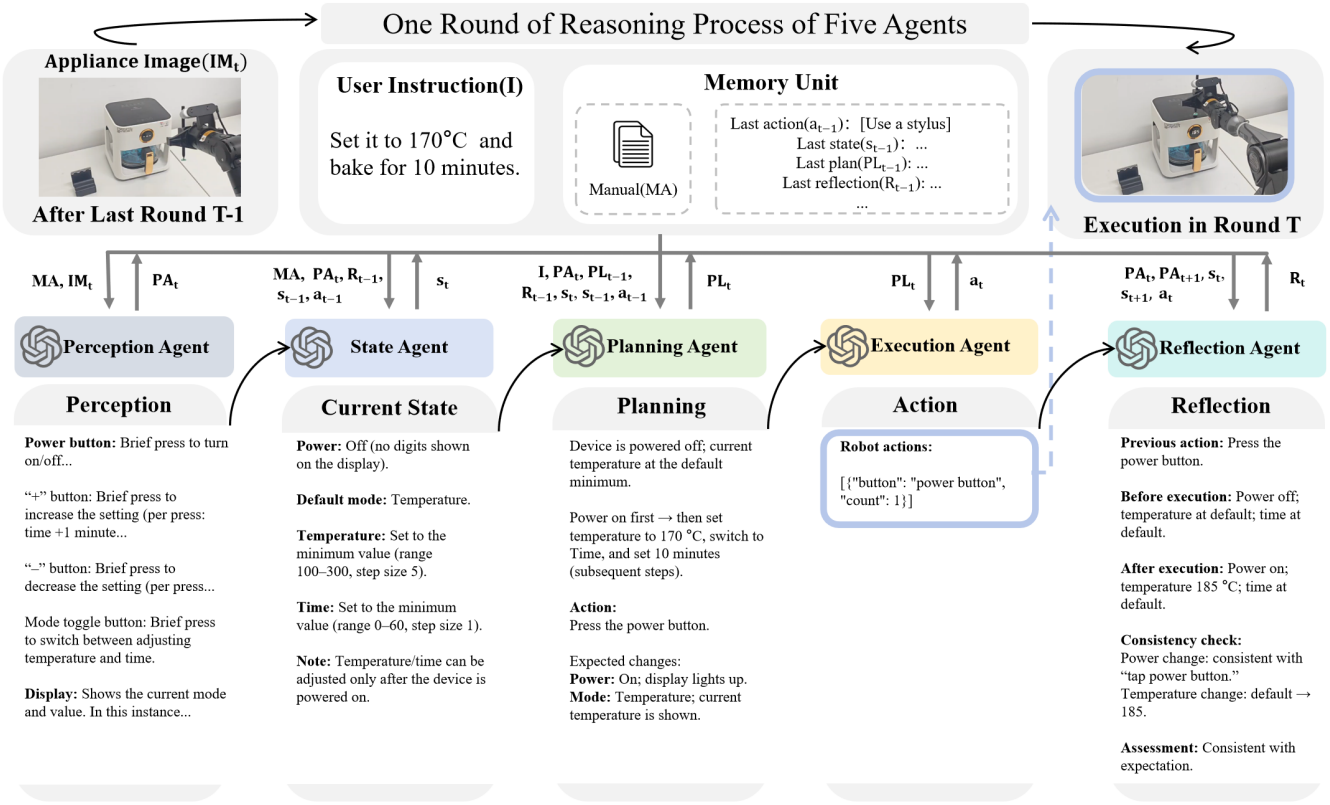


Fig. 3: An illustration of the inputs and outputs of the Multi-agent Home Appliance Operation, showing how they collaborate to determine the next action after the previous task.

execution of long-horizon tasks.

2) Perception agent: The perception agent parses the appliance's control panel and the content of the manual to establish usage procedures and an initial estimate of the appliance state. First, the agent applies optical character recognition (OCR) and object-detection models to identify all interactive components on the control panel (buttons, knobs, displays, etc.) and their locations in the image $\mathcal{IM}_t$. For each detected component, the agent then consults the manual $\mathcal{MA}$ and invokes a VLM to extract the component's name, functional description, and operating method. In parallel, the agent acquires the appliance's current state from visual cues, such as numerical readouts on the display, the status of indicator lights and the current position of knobs. These outputs are collectively referred to as $\mathcal{PA}_t$. This process can be formulated as:

$$\mathcal{PA}_t = f_{\text{perceive}}(\mathcal{IM}_t, \mathcal{MA})$$

where f_{perceive} represents the visual perception module and the VLM of the perception agent.

3) State agent: The perception agent provides a partial but potentially incomplete view of the appliance state. Since many appliances involve multi-level modes and parameters (e.g., power state, operating mode, target temperature, set time) that cannot be fully inferred from a single observation, a single observation is rarely sufficient to capture the full state. To address this limitation, a dedicated state agent is introduced to get a complete appliance state s_t. This agent leverages the manual's state descriptions and historical state information to fill in the gaps left by perception. In implementation, the entire manual $\mathcal{MA}$ is provided to an LLM, which is prompted to enumerate all possible state variables for the appliance (e.g., "Power: on/off," "Mode: bake/broil/…," "Temperature: numeric," etc.). Combining the perception agent's current readings with historical state s_{t-1}, the previous round's reflection information $\mathcal{R}_{t-1}$ and action logs a_{t-1}, the state agent infers the value of each state variable to form a complete estimate of the appliance's current state. This process can be represented as:

$$s_t = f_{\text{state}}(\mathcal{PA}_t, \mathcal{R}_{t-1}, \mathcal{MA}, a_{t-1}, s_{t-1})$$

where f_{state} represents the LLM of the state agent.

4) Planning agent: The planning agent translates the perception and state-estimation outputs into step-by-step plans to achieve the target task. Taking the user's natural language instruction I as the goal and conditioning on the discovered interactive components $\mathcal{PA}_t$, the current state s_t, and historical information, an LLM infers the required sequence of actions. These actions are expressed in a human-readable form (e.g., "Power button (press once)," "Temperature knob (rotate clockwise by 60°)") and are aligned with the actual control-interface elements of the appliance. The output of the planning agent is an initial operation plan $\mathcal{PL}_t$. In prompting the model, we emphasize operational constraints (drawn from manual

cautions), incorporate historical execution traces $\mathcal{PL}_{t-1}$, and include the previous round's reflection signals $\mathcal{R}_{t-1}$, state information s_{t-1} and a_{t-1} to improve plan validity and coherence. Formally, this process is defined as

$$\mathcal{PL}_t = f_{\text{plan}}(I, \mathcal{PA}_t, \mathcal{R}_{t-1}, \mathcal{PL}_{t-1}, a_{t-1}, s_t, s_{t-1})$$

where f_{plan} represents the LLM of the planning agent.

5) Execution agent: The execution agent serves as the bridge between abstract planning and physical execution. While the planning agent produces a human-readable sequence of actions $\mathcal{PL}_t$ aligned with control-interface elements, these steps must be further grounded into precise motor commands a_t for the robotic arm. The execution agent performs this translation by mapping each planned step to executable parameters, such as button coordinates, press counts, and knob rotation angles, thereby ensuring that the planned procedure can be realized on the physical appliance. This process can be expressed as:

$$a_t = f_{\text{decide}}(\mathcal{PL}_t)$$

where f_{decide} represents the LLM of the execution agent.

6) Reflection agent: The reflection agent closes the loop of perception, planning, and execution by continuously monitoring outcomes and correcting errors. After the robot completes a series of actions a_t according to the plan, the agent obtains the latest appliance state s_{t+1} (by re-invoking the perception and state agents) and compares it with the pre-execution state s_t. If state estimation goes wrong (e.g., a perceived numeric value exceeds the upper bound, or historical state is forgotten) or the expected change fails to materialize or anomalies occur (e.g., pressing a button does not alter the appliance state), the reflection agent analyzes likely causes and returns the findings as textual feedback $\mathcal{R}_t$ to the state agent and the planning agent, which then revises the subsequent state estimation and plan. This reflect–replan loop may iterate multiple times until the task is successfully completed or deemed infeasible. Formally, this process is represented as:

$$\mathcal{R}_t = f_{\text{reflect}}(\mathcal{PA}_t, \mathcal{PA}_{t+1}, s_t, s_{t+1}, a_t)$$

where f_{reflect} is the LLM of the reflection agent.

By doing so, the reflection agent endows the system with fault tolerance and self-correction: when the initial plan is imperfect, the robot can use feedback to converge toward a correct solution, improving success rates on complex tasks.

IV. Experiments

A. Dataset

1) Public simulation dataset: We conduct preliminary validation on the CheckManual dataset [4], which covers 10 common household-appliance categories (coffee makers, cameras, dishwashers, monitors, microwaves, ovens, printers, refrigerators, toasters, washing machines), comprising 182 appliance models and 1,107 task instances. Each instance pairs an appliance manual with a concrete operation description. Based on these manuals, we build simulators that correspond one-to-one with the 182 models to provide real-time state feedback during task execution, thereby evaluating the robot's ability to learn operations from manuals.

2) Self-collected dataset: To validate the system's adaptability to real control panels and manuals, we collect panel images and manuals for 6 appliance categories (air fryer, oven, toaster, rice cooker, microwave oven, washing machine) from e-commerce platforms (e.g., Amazon). Each category includes 5 different brands/models, and for each model, we collect 10 natural language operation commands (e.g., setting temperature and time). We also build a corresponding simulator for each model to test execution accuracy and operability. This dataset complements categories not covered by CheckManual, increasing the diversity and realism of the evaluation scenarios.

B. Implementation details

In the manual-summarization module, we utilize all-MiniLM-L6-v2 model [31] to embed the names of appliance components. For the visual perception module of the perception agent, we use the OCR model [32] and Grounding DINO [33] for object detection. GPT-4o [34] serves as the LLM and VLM driving all agents.

C. Evaluation

We pre-generate one Python simulator per appliance to deliver deterministic, fast feedback. The simulators contain control definitions, state variables, primitive actions, and macro-action composition. The shared interface lets agents interact identically across appliances. To assess the effectiveness of our framework, we define 3 evaluation metrics.

- Success Rate (SR): Percentage of instructions successfully completed across scenarios.
- Completion Rate (CR): Completed necessary steps divided by the total steps required for the task.
- Average Steps (AS): Average number of reasoning steps to solve instructions.

To comprehensively assess the performance of our multi-agent home-appliance operation system, we conduct experiments in two settings: the public dataset and the self-collected manuals/panel dataset. The specific settings are as follows.

1) Simulation experiment 1: We evaluate our system performance by comparing our method against baselines with the CheckManual dataset. Specifically, we consider the following methods:

- Ours w/ original manuals: HAA with original manuals (appliance-specific manuals) provided by CheckManual.
- Ours w/ general manuals: HAA with generalized manuals produced by the manual-summarization module.
- Ours w/o manual acquisition: HAA reasoning solely with the LLMs' built-in knowledge (without manual knowledge acquisition).
- ManualPlan: A single-agent home appliance operation system proposed in CheckManual with original manuals.
- ApBot: A multi-agent home appliance operation system with original manuals.

TABLE I: The results of our method compared to baselines.

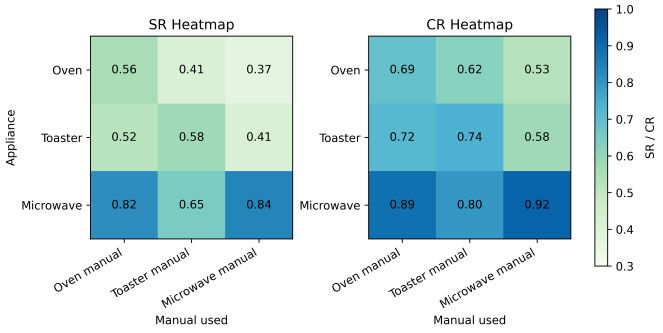
	Simulation Experiment 1			Simulation Experiment 2		
	SR	CR	AS	SR	CR	AS
ours w/ original manuals	0.69	0.82	5.75	0.82	0.89	6.35
ours w/ general manuals	0.58	0.73	4.78	0.77	0.83	6.72
ours w/o manual acquisition	0.22	0.42	5.38	0.37	0.60	6.51
ManualPlan	0.35	0.58	5.94	0.52	0.67	7.20
ApBot	0.62	0.78	6.35	0.75	0.82	6.67

2) Simulation experiment 2: Using our self-collected dataset instead of CheckManual within the same simulator framework, we conduct comparative experiments with the same methods of simulation experiment 1.

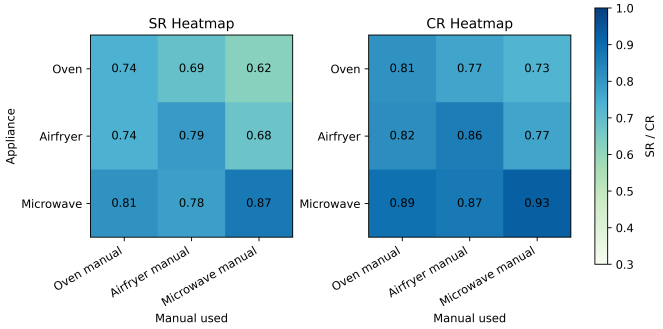
D. Result

Table I summarizes the results of simulation experiments 1 and 2, comparing our method under three modes of manual knowledge acquisition (original manual, generalized manual, and without manual knowledge acquisition) against two baselines (ManualPlan and ApBot). Table I indicates that our method with the original manual achieves the best performance in both experiments. Notably, our method with the generalized manual remains strong, which surpasses ManualPlan and differs only slightly from ApBot, highlighting its robustness. It trails the original by 11% in SR and 9% in CR in simulation experiment 1, but the gap narrows to 5% in SR and 6% in CR in simulation experiment 2. This trend suggests that the generalized manual can serve as an effective substitute for appliance-specific documentation. The smaller gap in experiment 2 is expected because CheckManual includes many instructions with implicit steps, for example "bake cookies" or "brew an espresso". These steps are explicitly listed in the original manuals but must be inferred with a generalized manual, which makes the tasks more challenging. As a consequence, using the generalized manuals in experiment 1 can omit certain required steps, leading to task failures and yielding a shorter AS. In experiment 2, the instructions are explicit and free of implicit steps, so the generalized manual guides planning almost as effectively as the original. The configuration without manual knowledge acquisition yields the worst outcomes in both experiments, indicating that the robot struggles to infer complete procedures without manual knowledge. Overall, the results validate the proposed method and show that when instructions are clearly specified, strong performance is maintained even with only a generalized manual.

To further evaluate the system's manual generalization ability, we consider the oven, microwave, and toaster from simulation experiment 1, and the air fryer, oven, and microwave from simulation experiment 2. In each case, a single category-level generalized manual from one appliance is applied to operate the other two. All other settings are identical to those in the two simulation experiments. The results of the manual generalization experiment are summarized in Fig. 4. Using category matched manuals, the diagonal entries are consistently the highest, showing that appliance-specific documentation yields the most reliable



(a) In simulation experiment 1 setting



(b) In simulation experiment 2 setting

Fig. 4: The generalization experiment result.

TABLE II: The ablation experiment result.

Stage Agent	✔	✗	✔	✔
Reflection Agent	✔	✔	✗	✔
Memory Unit	✔	✔	✔	✗
SR	0.77	0.63	0.71	0.59
CR	0.83	0.71	0.79	0.70
AS	6.72	8.55	6.48	8.67

procedures. Category crossing substitution is feasible with modest degradation. Using the most similar manual reduces performance by roughly 5%–15% SR and 4%–9% CR. Among substitutes, the oven manual transfers best to the other two appliances in both 2 experiments settings and outperforms the converse, reducing only 2%–6% SR and 2%–4% CR. We attribute this to control richness coverage, which means oven manuals typically enumerate a superset of operations (modes, temperature, and time), making them stronger proxies for related appliances. These observations validate our manual selection strategy: when an exact match is unavailable, prefer the nearest category with higher functional coverage rather than a simpler manual. This choice aligns with our manual-selection criteria and the setup of the manual

E. Ablation Experiments

We conduct ablations on the state agent, reflection agent, and memory unit using our self-collected dataset. All other settings are identical to those in simulation experiment 2 with the summarized manual.

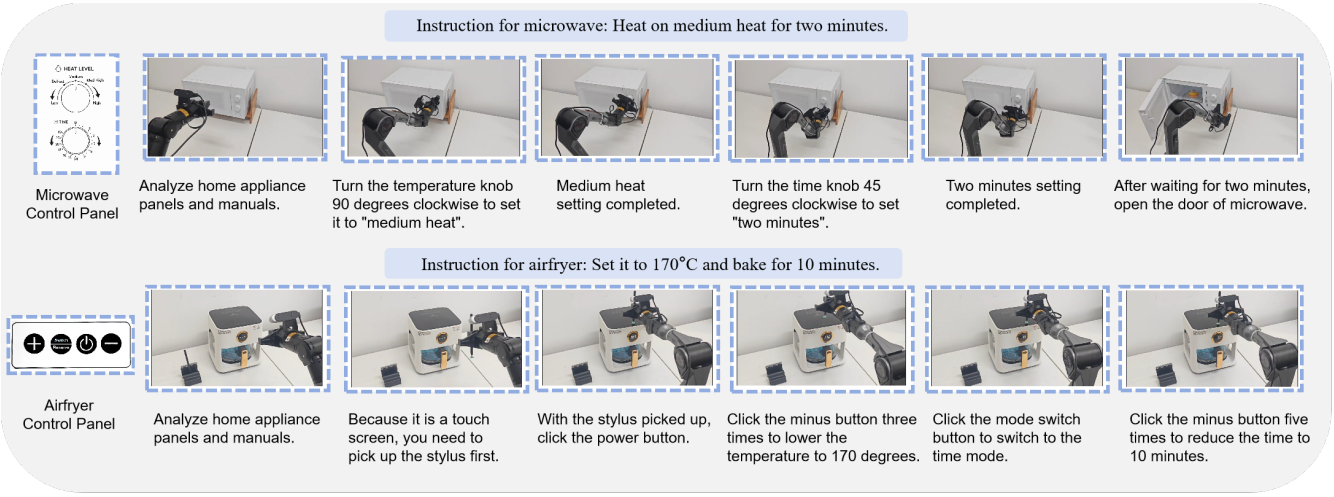


Fig. 5: Task execution workflow for the microwave and the air fryer.

TABLE III: The real-world experiment results.

	SR	CR	AS
ours w/ original manual	0.78	0.84	6.2
ours w/ general manual	0.70	0.79	6.5
ours w/o manual acquisition	0.35	0.52	5.3

As summarized in Table II, the experimental results demonstrate that both the state agent and the memory unit have a substantial impact. When either is removed, HAA frequently oscillates among appliance states, inflating AS and degrading reliability. This result reflects the importance of historical context and a globally consistent state representation for instructions that require multi-parameter adjustments. The reflection agent introduces a slight increase in AS but improves SR and CR, particularly on real control panel images, by correcting misrecognition and unexpected execution effects through closed-loop feedback.

V. Physical Validation

We conduct physical validation experiments to assess the reliability and robustness of our robotic system on real household appliances. Following the design in [6], the robot uses an RGB-D camera on the end effector to capture surface images and depth information. For button pressing, the system extracts the point cloud, computes the 3D centroid and surface normal, aligns the end effector, and advances 0.1 cm along the normal beyond the estimated surface using either the gripper or the capacitive stylus tip to complete the press. For knob rotation, the system detects the rotation axis, center, and normal, aligns and advances 0.1 cm, then closes the gripper and rotates the terminal joint to perform the adjustment.

The evaluation involves two microwaves and two air fryers, each with ten representative user instructions (e.g., "set the air fryer to 200°C for 10 minutes"). The appliances are used to obtain the results reported in Table III. For each instruction, task success is determined by verifying whether the final appliance state matches the target configuration, from which SR and CR are computed over all 40 tasks. Fig. 5 illustrates the execution workflow for microwave and air fryer tasks. The robot executes button presses and knob adjustments, enabling assessment of system reliability and stability in real-world settings.

As reported in Table III, the results on real-world appliance tasks indicate modest declines of 2–8 percentage points in SR and CR compared to simulation, while preserving relative performance trends: any manual guidance consistently outperforms the no-manual-acquisition configuration. The lower AS in the no-manual-acquisition case reflects early task termination due to failures rather than higher efficiency. Two main error sources are identified. First, illumination changes and specular glare on glossy panels intermittently corrupt OCR and numeric readouts. Second, small actuation biases when turning knobs or tapping tiny buttons lead to misalignment and occasional retries. Despite these challenges, the results confirm the feasibility, reliability, and scalability of our framework for real-world appliance interaction.

VI. Conclusions

In this work, we present Home Appliance Agent (HAA), a foundation-model-driven multi-agent framework for robust robotic operation of household appliances. HAA integrates specialized agents for perception, state estimation, planning, decision-making, and reflection, enabling end-to-end task execution. By combining manual summarization, category-level generalization, and on-demand web search, the system can operate both seen and unseen appliances without relying on appliance-specific manuals. Extensive experiments on public benchmarks, self-collected manuals, and real-world appliances demonstrate that HAA significantly outperforms prior methods in terms of success rate and generalization. Our results highlight the potential of multi-agent collaborative frameworks powered by LLMs and VLMs for autonomous and adaptive robot manipulation in diverse home environments. Future work will extend the application of HAA

beyond household appliances to a broader range of appliances, aiming to enable robust and adaptive robotic operation across diverse environments.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Fund for Distinguished Young Scholars under Grant 62025304, National Natural Science Foundation Project under Grant 62273054, and the Fundamental Research Funds for the Central Universities.

REFERENCES

[1] W. Huang, C. Wang, Y. Li, R. Zhang, and L. Fei-Fei, "ReKep: Spatio-temporal reasoning of relational keypoint constraints for robotic manipulation," in *8th Annual Conference on Robot Learning*, 2024.

[2] M. Pan, J. Zhang, T. Wu, Y. Zhao, W. Gao, and H. Dong, "OmniManip: Towards general robotic manipulation via object-centric interaction primitives as spatial constraints," *arXiv preprint arXiv:2501.03841*, 2025.

[3] Y. Chen, W. Li, S. Wang, H. Zhuang, and Q. Wu, "T-rex: Task-adaptive spatial representation extraction for robotic manipulation with vision-language models," *arXiv preprint arXiv:2506.19498*, 2025.

[4] Y. Long, J. Zhang, M. Pan, T. Wu, T. Kim, and H. Dong, "Checkmanual: A new challenge and benchmark for manual-based appliance manipulation," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 22 595–22 604.

[5] A. Yuguchi, T. Nakamura, M. Toyoda, M. Yamada, P. Tulathum, M. Aubert, G. A. Garcia Ricardez, J. Takamatsu, and T. Ogasawara, "Toward robot-agnostic home appliance operation: a task execution framework using motion primitives, ontology, and gui," *Advanced Robotics*, vol. 36, no. 11, p. 548–565, 2022.

[6] J. Zhang, H. Zhang, A. Xiao, and D. Hsu, "Robot operation of home appliances by reading user manuals," *arXiv preprint arXiv:2505.20424*, 2025.

[7] S. Yang, O. Nachum, Y. Du, J. Wei, P. Abbeel, and D. Schuurmans, "Foundation models for decision making: Problems, methods, and opportunities," *arXiv preprint arXiv:2303.04129*, 2023.

[8] B. Ichter, A. Brohan, Y. Chebotar, C. Finn, K. Hausman, A. Herzog, D. Ho, J. Ibarz, A. Irpan, E. Jang, R. Julian, D. Kalashnikov, S. Levine, Y. Lu, C. Parada, K. Rao, P. Sermanet, A. T. Toshev, V. Vanhoucke, F. Xia, T. Xiao, P. Xu, M. Yan, N. Brown, M. Ahn, O. Cortes, N. Sievers, C. Tan, S. Xu, D. Reyes, J. Rettinghouse, J. Quiambao, P. Pastor, L. Luu, K.-H. Lee, Y. Kuang, S. Jesmonth, N. J. Joshi, K. Jeffrey, R. J. Ruano, J. Hsu, K. Gopalakrishnan, B. David, A. Zeng, and C. K. Fu, "Do as i can, not as i say: Grounding language in robotic affordances," in *Proceedings of The 6th Conference on Robot Learning*, 2023, pp. 287–318.

[9] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, P. Sermanet, N. Brown, T. Jackson, L. Luu, S. Levine, K. Hausman, and B. Ichter, "Inner monologue: Embodied reasoning through planning with language models," *arXiv preprint arXiv:2207.05608*, 2022.

[10] J. Liang, W. Huang, F. Xia, P. Xu, K. Hausman, B. Ichter, P. Florence, and A. Zeng, "Code as policies: Language model programs for embodied control," in *2023 IEEE International Conference on Robotics and Automation*, pp. 9493–9500.

[11] K. Lin, C. Agia, T. Migimatsu, M. Pavone, and J. Bohg, "Text2motion: from natural language instructions to feasible plans," in Autonomous Robots vol. 47, no. 8, pp. 1345–1365, 2023.

[12] W. Huang, C. Wang, R. Zhang, Y. Li, J. Wu, and F. Li, "Voxposer: Composable 3d value maps for robotic manipulation with language models," in *Proceedings of the 7th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, vol. 229, 2023, pp. 540–562.

[13] H. Huang, F. Lin, Y. Hu, S. Wang, and Y. Gao, "CoPa: General robotic manipulation through spatial constraints of parts with foundation models," in *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2024, pp. 9488–9495.

[14] A. Goldberg, K. Kondap, T. Qiu, Z. Ma, L. Fu, J. Kerr, H. Huang, K. Chen, K. Fang, and K. Goldberg, "Blox-net: Generative design-for-robot-assembly using vlm supervision, physics simulation, and a robot with reset," *arXiv preprint arXiv:2409.17126*, 2024.

[15] C. Tie, S. Sun, J. Zhu, Y. Liu, J. Guo, Y. Hu, H. Chen, J. Chen, R. Wu, and L. Shao, "Manual2skill: Learning to read manuals and acquire robotic skills for furniture assembly using vision-language models," *arXiv preprint arXiv:2502.10090*, 2025.

[16] A. A. Abdulla, H. Liu, N. Stoll, and K. Thurow, "A robust method for elevator operation in semi-outdoor environment for mobile robot transportation system in life science laboratories," in *2016 IEEE 20th Jubilee International Conference on Intelligent Engineering Systems (INES)*, 2016, p. 45–50.

[17] D. Zhu, T. Li, D. Ho, T. Zhou, and M. Q.-H. Meng, "A novel ocr-rcnn for elevator button recognition," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, p. 3626–3631.

[18] F. Wang, G. Chen, and K. Hauser, "Robot button pressing in human environments," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, p. 7173–7180.

[19] V. Sukhoy and A. Stoytchev, "Learning to detect the functional components of doorbell buttons using active exploration and multimodal correlation," in *2010 10th IEEE-RAS International Conference on Humanoid Robots*, 2010, p. 572–579.

[20] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems 35*, 2022.

[21] S. Yao, J. Z. Yu, D. Yang, R. Cui, I. Shafran, T. Griffiths, and K. N. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[22] N. Shinn, W. Labash, and A. Gopinath, "Reflexion: Language agents with verbal reinforcement learning," in *Advances in Neural Information Processing Systems 36*, 2023.

[23] J. Wang, H. Xu, H. Jia, X. Zhang, M. Yan, W. Shen, J. Zhang, F. Huang, and J. Sang, "Mobile-agent-v2: Mobile device operation assistant with effective navigation via multi-agent collaboration," in *Advances in Neural Information Processing Systems 37*, 2024.

[24] Z. Wang, H. Xu, J. Wang, X. Zhang, M. Yan, J. Zhang, F. Huang, and H. Ji, "Mobile-agent-e: Self-evolving mobile assistant for complex tasks," *arXiv preprint arXiv:2501.11733*, 2025.

[25] J. Ye, X. Zhang, H. Xu, H. Liu, J. Wang, Z. Zhu, Z. Zheng, F. Gao, J. Cao, Z. Lu, J. Liao, Q. Zheng, F. Huang, J. Zhou, and M. Yan, "Mobile-agent-v3: Fundamental agents for gui automation," *arXiv preprint arXiv:2508.15144*, 2025.

[26] C. Zhang, Z. Yang, J. Liu, Y. Han, X. Chen, Z. Huang, B. Fu, and G. Yu, "Appagent: Multimodal agents as smartphone users," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 2025.

[27] W. Tan, W. Zhang, X. Xu, H. Xia, Z. Ding, B. Li, B. Zhou, J. Yue, J. Jiang, Y. Li, R. An, M. Qin, C. Zong, L. Zheng, Y. Wu, X. Chai, Y. Bi, T. Xie, P. Gu, X. Li, C. Zhang, L. Tian, C. Wang, X. Wang, B. F. Karlsson, B. An, S. Yan, and Z. Lu, "Cradle: Empowering foundation agents towards general computer control," *arXiv preprint arXiv:2403.03186*, 2024.

[28] H. Singh, R. J. Das, M. Han, P. Nakov, and I. Laptev, "Malmm: Multi-agent large language models for zero-shot robotics manipulation," *arXiv preprint arXiv:2411.17636*, 2024.

[29] J. A. Hartigan, *Clustering Algorithms*. New York: John Wiley & Sons, 1975.

[30] L. McInnes, J. Healy, and S. Astels, "hdbscan: Hierarchical density based clustering," *Journal of Open Source Software*, vol. 2, no. 11, p. 205, 2017.

[31] Sentence-Transformers, "all-minilm-l6-v2: A fast and compact sentence-transformer model," `https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2`, 2024.

[32] JaidedAI, "Easyocr," `https://github.com/JaidedAI/EasyOCR`, 2020.

[33] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su *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.

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