# RePlan-Bot: Multi-Level Replanning for Embodied Instruction Following

Xicheng Gong[1] Guozheng Sun[2] Peiran Xu[1] Yadong Mu[1,*]
[1]Peking University [2]Tsinghua University
gongxicheng@stu.pku.edu.cn sgz25@mails.tsinghua.edu.cn xpr820@pku.edu.cn myd@pku.edu.cn
*Corresponding author

## Abstract

*Embodied instruction following (EIF) requires agents to understand and execute complex natural language commands within interactive 3D environments. Despite recent advances, existing methods often fail in long-horizon planning and handling irreversible state changes, resulting in low task success rates. To address these challenges, we introduce **RePlan-Bot**, a novel EIF agent that performs multi-level, continuous replanning throughout task execution. RePlan-Bot integrates a high-level LLM-based auditor for dynamic sub-goals adjustments guided by environmental feedback, a commonsense-guided search mechanism based on a multi-layered instance map for precise and structured object localization, and a lightweight ViT-based corrector to preemptively fix risky low-level actions. Evaluated on the ALFRED benchmark, RePlan-Bot achieves state-of-the-art performance in both seen and unseen environments, demonstrating superior adaptability and reliability.*

## 1. Introduction

Embodied instruction following (EIF) tasks require autonomous agents to understand and execute natural language commands within visually rich, interactive environments. Typical EIF scenarios, exemplified by benchmarks such as ALFRED [31], involve sequences of navigation and manipulation actions guided by instructions like "Clean the mug and place it on the countertop." These tasks pose significant challenges, as agents must handle long-horizon planning, cope with partial observability, and manage irreversible state changes. Consequently, even the most advanced EIF systems [12, 19, 35, 41] often exhibit low task success rates, highlighting fundamental shortcomings of existing methods.

Previous EIF solutions [9, 11, 12, 19, 21, 23, 29, 36] typically follow one of two strategies. End-to-end approaches directly map sensory inputs to actions, but frequently fail to generalize beyond training scenarios due to limited reasoning capability and insufficient interpretability; Modular methods, represented by systems such as FILM [19] and CAPEAM [12], decompose EIF tasks into high-level planning, mid-level object searching, and low-level action execution. While modularity improves interpretability, these approaches remain vulnerable because their modules typically lack adaptability: high-level planners produce static action templates disconnected from real-time visual feedback; mid-level search mechanisms use coarse semantic heatmaps inadequate for precise localization of small or partially occluded objects; and low-level controllers rely on fixed, primitive actions, susceptible to errors arising from inaccurate perception or faulty actuation. Address-

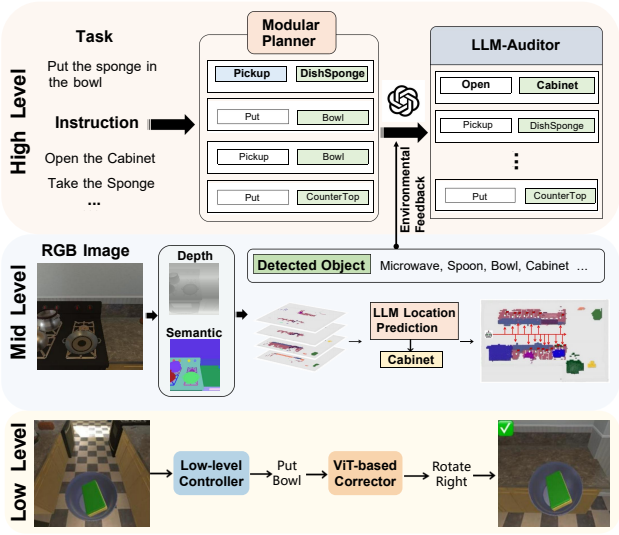


Figure 1. **Overview of the proposed RePlan-Bot**. It consists of three components: High Level Replanning, an LLM-based Auditor dynamically replans high-level goals; Mid Level Searching, a commonsense-driven module guides object search via multi-layered instance map; and Low Level Replanning, a ViT-based Corrector monitors low-level actions to prevent execution failures.

ing these critical limitations, we propose RePlan-Bot, a novel EIF agent that integrates multi-level and continuous replanning. As shown in Figure 1, at the high level, we introduce an environment-aware auditor powered by a large language model (LLM) that dynamically adjusts subgoals by considering both textual instructions and current semantic context. For mid-level exploration, we present a commonsense-driven search mechanism leveraging a multi-layered instance map segmented by object instances and vertical spatial information, enabling targeted and structured exploration for unseen objects. At the low level, inspired by the intuition that humans continuously adjust their actions based on visual observations in the real world, we propose a lightweight vision transformer (ViT)-based [7] action corrector trained on a dataset constructed from execution trajectories. The corrector monitors each primitive action, evaluates its feasibility using egocentric images, and proactively substitutes risky actions with safer alternatives. This comprehensive framework significantly improves the robustness and overall performance of the EIF agent. Experimental results on the challenging ALFRED benchmark [31] demonstrate that RePlan-Bot substantially surpasses prior state-of-the-art methods, achieving marked improvements in both seen and unseen scenarios. Extensive ablations further confirm the individual and collective contributions of our proposed multi-level replanning approach.

Our main contributions can be summarized as follows:

- We design a **high-level environment-aware auditor** powered by LLM, which refines task plans on the fly by jointly leveraging natural language instructions and real-time environmental feedback.
- We propose a **commonsense-guided search mechanism** based on a multi-layered instance map, enabling precise object localization and structured exploration, especially under partial occlusion or cluttered scenes.
- We develop a **ViT-based low-level action corrector** trained on visual observations captured from execution trajectories, capable of predicting and correcting potential failures in primitive actions, thereby improving execution accuracy and safety.
- We conduct extensive experiments on the ALFRED benchmark [31], where RePlan-Bot achieves new state-of-the-art performance in both seen and unseen environments. Detailed ablations validate the effectiveness of each proposed component.

## 2. Related Work

**Embodied Instruction Following** Embodied Instruction Following (EIF) tasks require agents to translate natural-language commands into multi-step behaviors in 3D, partially observed environments. Early end-to-end approaches directly map egocentric observations and language to actions [11, 22, 23, 34, 36, 46]. For example, EmBERT [36] fuses panoramic RGB and step-wise text via a multimodal transformer to predict the next action. However, such models generalize poorly, relying heavily on annotated data and often mis-grounding objects in cluttered scenes.

To overcome these limitations, recent modular pipelines decouple high-level planning from low-level control [5, 6, 9, 12, 19, 21, 48], inferring abstract action plans and executing them via semantic or instance-level maps. FILM [19] segments instructions into sub-goals and incrementally builds a semantic map, while CAPEAM [12] leverages a context-aware planner and memory for robust long-horizon manipulation.

**LLMs for Embodied Instruction Following** Large language models (LLMs) have emerged as powerful high-level planners in embodied AI, demonstrating strong generalization across tasks such as vision–language navigation [3, 17, 24, 28, 39, 47], object-centric manipulation [10, 37, 42, 44], and long-horizon household planning [1, 4, 16, 20, 25, 32, 40]. Typically, LLMs generate sub-goals or action sketches from language instructions, and then low-level controllers handle execution. Despite strong zero-shot generalization via commonsense priors, LLMs lack explicit spatial grounding and struggle with fine-grained control due to large action spaces. To address this, hybrid architectures assign high-level reasoning to LLMs and delegate perception and control to specialized modules. For instance, ThinkBot [16] completes sparse instructions via self-reasoning and grounds plans using a multimodal object localizer.

On this basis, to address execution failures stemming from incomplete perception, ambiguous instructions, or dynamic environmental changes, recent methods integrate adaptive replanning modules into embodied agents [2, 4, 14, 16, 18, 30, 33, 35, 38, 43]. This enhances robustness in long-horizon and partially observable tasks. For instance, FLARE [14] substitutes undetected objects with semantically similar alternatives using language-encoder similarity (e.g., "TrashCan"→"GarbageCan"). These methods enable real-time and language-aware replanning, allowing agents to adapt fluidly in open-world scenarios.

## 3. Approach

The typical EIF pipeline [12, 19] struggles when language is sparse or ambiguous, which often results in logically incomplete plans. In addition, single-layer semantic map routinely misses small or occluded items. Moreover, rigid controller that ignores ongoing perceptual feedback cannot predict or recover from execution failures, leading to aimless exploration, collisions, and cascading errors.

To overcome these limitations, we introduce a continuous multi-level replanning approach, as shown in Figure 2.

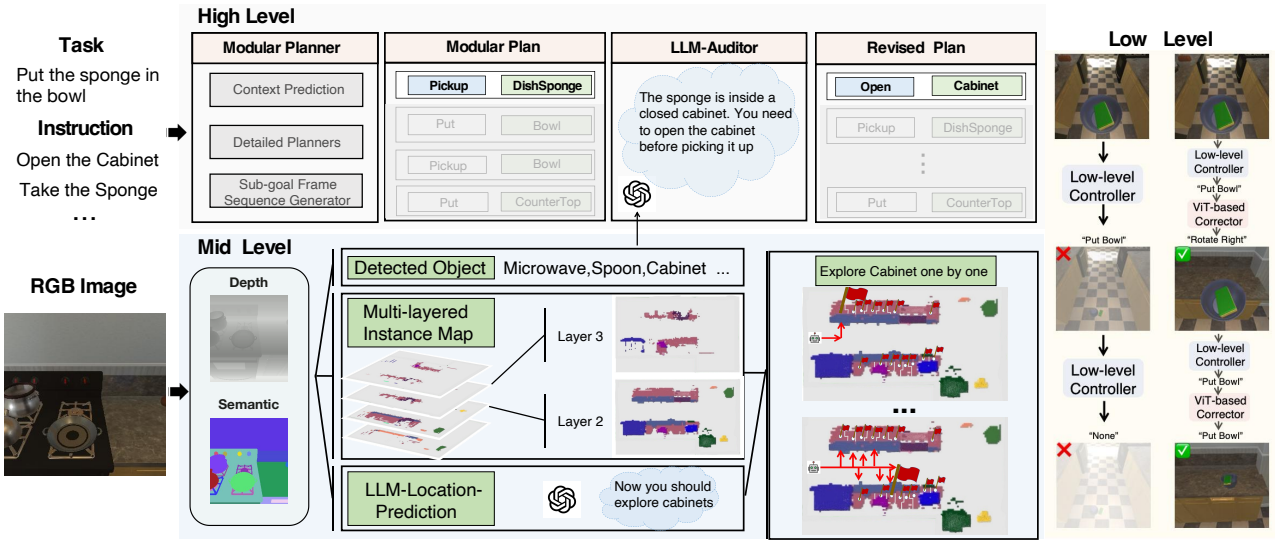


Figure 2. **The detailed pipeline of RePlan-Bot.** At the high level, upon receiving natural-language commands, the Modular Planner generates an initial plan. The LLM-Auditor then refines this plan to make it more rational. During task execution, the LLM-Auditor continuously optimizes the plan based on environmental feedback. At the mid level, the commonsense-guided search mechanism uses a multi-layered instance map and an LLM to predict object locations ("🚩") for precise and efficient search. At the low level, the ViT-based corrector adjusts the originally planned low-level actions based on the current RGB input to ensure correct execution.

At the high level, an LLM-based auditor refines under-specified sub-goal sequences by integrating instruction and environmental feedback, enabling real-time adjustment to dynamic scenes. At the mid level, a commonsense-guided search mechanism leverages a multi-layered instance map, which enables thorough and efficient exploration, particularly in environments containing multiple instances of the same object category. At the low level, a ViT-based corrector continuously monitors egocentric observations to anticipate failures and proactively substitute risky actions with safer alternatives to enhance overall task reliability. This hierarchical replanning mechanism enables the agent to revise flawed high-level plans, navigate purposefully under partial observability, and recover robustly from execution failures. Below is the detailed explanation of each component.

## 3.1. High-Level Replanning: Environment-Aware Sub-goals Auditing

Large language models excel at inferring latent structure from language and chaining symbolic steps, both of which naturally align with the high-level planning demands of embodied instruction following [9, 35]. We exploit these strengths by inserting an LLM auditor after the Modular High-Level Planner. The auditor receives a structured prompt containing three streams of information: System Message, including the role explanation, task definition, and response format; Agent Message, which indicates the task instruction and the given sparse step-by-step instruction sentences; Environmental Feedback, which includes the observed objects and the current step index in sub-goals list. It must emit a revised sub-goals list that conforms to the same schema yet is fully grounded in the observed scene. We detail the language model setup and provide the full prompt structure in Appendix 1.2.

Specifically, high-level replanning is triggered under these two conditions: (1) at the beginning of each task, an initial audit is introduced to ensure that the sub-goals list is consistent with the scene configuration and task instruction; (2) when the agent takes an excessive number of steps to complete a sub-goal, the auditor re-evaluates the sub-goal in context to stay aligned with the evolving environment. This hybrid trigger ensures that replanning is both proactive and anticipatory, making it capable of adapting in real time. The details are described as follows.

### 3.1.1. Pre-Execution Sub-Goals Refinement

In many modular approaches such as CAPEAM [12], high-level sub-goals are generated by matching objects with predefined action templates, without reasoning about their spatial state. This may produce infeasible plans: for instance, the agent may attempt (PickUp, DishSponge) even when the object is inside a closed cabinet, without opening it first.

To address this limitation, we introduce an LLM-based auditor that refines the sub-goals prior to the execution of each task. At the beginning of each task, the auditor performs an initial review of the sub-goals list generated by the Modular High-Level Planner, supplementing missing

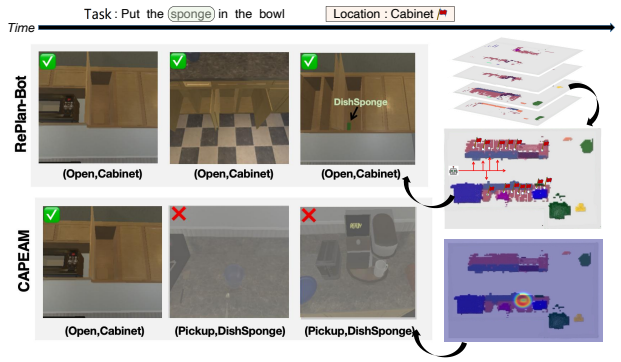


Figure 3. **Comparison between RePlan-Bot and conventional EIF methods (CAPEAM [12]).** RePlan-Bot predicts the sponge is in a cabinet (" 🚩 ") and successfully finds it after exploring multiple cabinets. In contrast, CAPEAM checks only one cabinet and moves on without verifying the sponge's location, resulting in task failure.

actions and correcting object reference errors arising from semantic ambiguity. For instance, in the above case, the auditor augments the plan with (Open, Cabinet) before attempting the pickup. By explicitly reasoning over spatial visibility and containment, the auditor identifies and corrects missing preconditions that would otherwise cause execution to fail.

#### 3.1.2. Vision-Grounded Object Disambiguation

Existing instruction-following systems often select object references purely based on textual similarity, without grounding them in the visual context. This can result in selecting semantically relevant but visually absent targets, leading to invalid or unnecessary search behaviors, especially in cluttered or visually ambiguous environments.

To address this issue, we leverage the LLM-based auditor to refine sub-goals based on the detected-object set and the original sub-goals list. For instance, when the agent takes too many steps to execute the action (PickUp, Cup), the auditor may revise this sub-goal to (PickUp, Mug), guided by the fact that Mug is present in the detected object set. This adjustment ensures successful task completion. By grounding symbolic references in actual visual observations, the auditor substitutes unsupported predictions with visually available alternatives. This alignment between symbolic planning and egocentric perception enhances the robustness of the overall plan and reduces failures caused by semantic misalignment.

### 3.2. Mid-Level Searching: Commonsense-Guided Search Mechanism

Given a global semantic map $M_t \in \{0,1\}^{C\times H\times W}$ and a subgoal-defined target object $o^\star$, the agent must determine where to begin its search when $o^\star$ is unobserved. Prior approaches such as FILM [19] and CAPEAM [12] compress $M_t$ into a fixed-resolution heatmap $H_t \in \mathbb{R}^{8\times 8}$ using a trained U-Net [26], which guides the agent toward a coarse likelihood peak. However, such predictions are spatially imprecise and often unreliable for small or novel items.

To facilitate a more precise and object-aware exploration search strategy, we construct a **multi-layered instance map** $M_t \in \{0,1,\ldots,N\}^{C\times H\times W\times Z}$, where $C$ denotes the number of semantic categories such as `Cabinet` and `Fridge`, $H$ and $W$ represent the height and width of the map grid respectively, and $Z$ indicates discrete height slices obtained by partitioning the 3D voxelized observations vertically; meanwhile, $\{0,1,\ldots,N\}$ correspond to the instance IDs assigned to each pixel within the same semantic category. At each timestep, the RGB image is processed into a depth map and instance segmentation mask using Mask R-CNN [8], from which we recover a 3D point cloud. Each 3D point is associated with a predicted semantic category and discretized into a voxel grid along the vertical axis. This mapping strategy enables object localization across the scene layout while supporting instance-level differentiation along the vertical axis, such as distinguishing upper and lower cabinet compartments.

To update the multi-layered semantic map, we project the 3D voxelized observations into the appropriate height slice based on their vertical coordinates. Each pixel is labeled with an instance *ID*, and contributes to the corresponding $(c,z)$ slice of the map. In each slice, we compute :

$$ID(I_t) = \begin{cases} \arg\max\limits_{i\in\mathcal{I}_{t-1}} \mathrm{IoU}(I_t,i), & \text{if } \max \mathrm{IoU}(I_t,i) > \theta \\ \text{NewID}, & \text{otherwise} \end{cases} \quad (1)$$

where $i$ indexes a candidate instance in the maintained instance set $\mathcal{I}_{t-1}$, $\mathrm{IoU}(I_t,i)$ denotes the intersection-over-union between the current mask $I_t$ and instance $i$, and $\theta$ is a predefined threshold that determines whether the current mask $I_t$ sufficiently overlaps with any existing instance. This process enables instance-level tracking conditioned on both semantic and height-based spatial constraints.

Building upon this spatially grounded map, we now describe how the agent performs target-directed search. When the target object is small or visually occluded, direct localization from egocentric perception is often unreliable. Humans naturally resolve this challenge by using semantic priors, inferring object locations based on their typical co-occurrence with larger, stable support objects. Inspired by this, we introduce a LLM that predicts a high-level host category $h^\star$ given the task description, step-by-step instruction, and the set of currently detected objects. The prediction acts as a semantic anchor to narrow the search space, while visited locations are dynamically masked to avoid redundancy. This targeted search strategy is triggered only when the goal object has not yet been observed in the environment. The language model configuration, along with

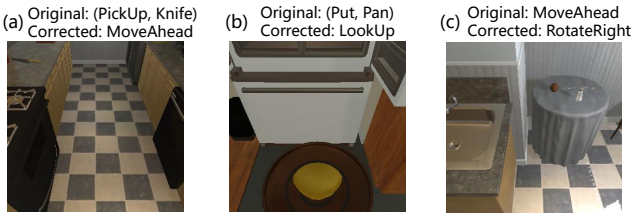


Figure 4. **Example visualization of the RGB image and the corresponding low-level action.** (a) The agent is too far to `PickUp Knife`, so the action is corrected to `MoveAhead`. (b) The viewpoint is too low to `Put Pan in Fridge`, so the action is corrected to `LookUp`. (c) The agent is blocked when trying to `MoveAhead`, so the action is corrected to `RotateRight`.

detailed listings of small object classes and host category definitions, are presented in Appendix 1.3.

Once $h^\star$ is predicted, the agent queries the multi-layered instance map to enumerate all instances of the host category. Instead of relying on coarse likelihood heatmaps, the agent conducts a structured, instance-wise search by explicitly visiting each candidate in sequence. As shown in Figure 3, based on the LLM's prediction that the sponge is inside a `Cabinet`, the agent explores multiple cabinet instances one by one and successfully locates the hidden sponge, which remains fully occluded throughout. By aligning language-driven semantic inference with spatial instance reasoning, our approach yields robust performance in cluttered and partially observable environments.

### 3.3. Low-Level Replanning: Vision-Driven Action Correction

In most modular EIF systems, once the FMM planner [27] issues a short-term goal, the low-level controller adjusts its heading using basic "MoveAhead" and "Rotate" actions based on the semantic map. Upon reaching the goal region, it checks for a segmented mask of the target in the current RGB frame and executes a predefined action, such as PickUp, without judging whether the action is feasible. As a result, while executing actions, this template-driven approach struggles to handle real-world imperfections: the collision map may misrepresent free space due to coarse resolution; segmentation may miss the object under occlusion; and actuation errors such as grasp misalignment can send the agent off course. As a result, fixed primitives often attempt infeasible moves, such as driving into undetected obstacles, over-rotating, or grasping at empty air, which leads to repeated failures and wasted time.

In the real world, humans continuously adjust their actions based on current visual information. Inspired by this, we interleave a lightweight ViT-based corrector into each primitive step. Specifically, the ViT module takes the current egocentric RGB image and the originally planned low-level action as input, evaluates its feasibility, and outputs a potentially better alternative. Formally, we denote the RGB observation as $O_t$ at time $t$, and $a_t^{\text{plan}}$ the action proposed by the low-level controller. The ViT model computes:

$$\hat{a}_t = \mathcal{F}_{\text{ViT}}(I_t, a_t^{\text{plan}}), \tag{2}$$

where $\mathcal{F}_{\text{ViT}}$ is a transformer-based policy head that outputs a corrected action $\hat{a}_t \in \mathcal{A}$, with $\mathcal{A}$ denoting the discrete low-level action space (e.g., `MoveAhead`, `RotateLeft`, `RotateRight`, `LookDown`, and `LookUp`). When $\hat{a}_t = a_t^{\text{plan}}$, the agent proceeds with the original plan; otherwise, the system treats $\hat{a}_t$ as a more suitable alternative and executes it instead.

To train the ViT module, each training sample consists of an RGB image $O_t$ and the planned action $a_t^{\text{plan}}$, where the action is embedded as a discrete token and concatenated with the image tokens before being passed to the transformer encoder. A softmax classification head predicts the corrected action over the space $\mathcal{A}$, supervised by the relabeled ground truth. This setup enables the ViT to function both as an action validator and a corrector, learning to recognize execution failures and predict better alternatives in visually ambiguous or error-prone situations. The training dataset and details are presented in Appendix 1.4.

## 4. Experiments

### 4.1. Experimental Setup

**Dataset.** We evaluate our method on the ALFRED benchmark [31], which contains 25,743 instruction-demonstration pairs for interactive 3D household tasks in AI2-THOR [15]. The dataset spans seven task types involving long-horizon, multi-step navigation and manipulation under partial observability. It is split into training, validation, and test sets, with validation and test environments further divided into seen and unseen folds to assess generalization.

**Evaluation Metrics.** We report official ALFRED metrics: Success Rate (SR, percentage of episodes with all subgoals completed), Goal-Condition Success Rate (GC, proportion of individual goal conditions achieved), and their path length-weighted variants (PLWSR, PLWGC), which penalize inefficient trajectories. These metrics jointly evaluate both the effectiveness and efficiency of instruction following. Additional details are provided in Appendix 1.1.

### 4.2. Comparison with the State of the Art

We compare RePlan-Bot with several recent state-of-the-art models on the ALFRED benchmark, including FILM [19], LGS-RPA [21], Prompter [9], CAPEAM [12], and so on. Since our approach is based on CAPEAM, we use the reproduced results for CAPEAM in our evaluation, while the results for other baselines are taken from the reported values.

Table 1. Comparison with state-of-the-art methods on the ALFRED benchmark (Test Seen/Unseen).

| Method | Test Seen | | Test Unseen | |
|---|---|---|---|---|
| | GC(PLWGC) | SR(PLWSR) | GC(PLWGC) | SR(PLWSR) |
| *Task Description Only* | | | | |
| HLSM | 35.79(11.53) | 25.11(6.69) | 27.24(8.45) | 16.29(4.34) |
| FILM | 36.15(14.17) | 25.77(10.39) | 34.75(13.13) | 24.46(9.67) |
| LGS-RPA | 41.71(24.49) | 33.01(16.65) | 38.55(20.01) | 27.80(12.92) |
| Prompter | 56.98(**28.42**) | 47.95(23.29) | 56.57(25.80) | 41.53(18.84) |
| CAPEAM | 53.14(24.32) | 43.64(19.76) | 54.17(23.68) | 40.42(19.34) |
| **RePlan-Bot** | **57.81**(25.23) | **49.23**(**23.37**) | **57.02**(**26.31**) | **44.79**(**21.89**) |
| *Task Description + Step-by-step Instructions* | | | | |
| ABP | 51.13(4.92) | 44.55(3.88) | 24.76(2.22) | 15.43(1.08) |
| FILM | 39.55(15.59) | 28.83(11.27) | 38.52(15.13) | 27.80(11.32) |
| LGS-RPA | 48.66(28.97) | 40.05(21.28) | 45.24(22.76) | 35.41(**22.76**) |
| Prompter | 60.22(**30.21**) | 51.17(**25.12**) | 56.57(25.80) | 45.32(20.79) |
| CAPEAM | 57.32(26.31) | 47.61(22.82) | 56.52(24.26) | 43.17(20.13) |
| **RePlan-Bot** | **61.21**(26.69) | **52.05**(24.60) | **60.29**(**26.31**) | **47.61**(21.89) |

Notably, the reproduced results of CAPEAM differ slightly from those originally reported [13]. All methods are evaluated with task description only and with both task description and step-by-step instructions as input, using PLWGC, GC, PLWSR, and SR on both Test Seen and Test Unseen splits.

**Task Description Only.** In the setting where only the task description is provided, RePlan-Bot achieves 44.79% SR and 57.02% GC on the Test Unseen split, outperforming all baselines by a clear margin (e.g., CAPEAM: 40.42% SR, 54.17% GC). On the Test Seen split, our method also leads with 49.23% SR and 57.81% GC. This substantial improvement demonstrates strong generalization to novel environments, even with limited high-level instructions. Such robustness makes RePlan-Bot practical for real-world scenarios where step-by-step guidance may be unavailable.

**Both Task Description and Step-by-Step Instructions.** When both the task description and step-by-step instructions are provided, RePlan-Bot further improves across both SR and GC, reaching 47.61% SR and 60.29% GC on the Test Unseen split, and 52.05% SR and 61.21% GC on the Test Seen split. These results highlight the model's ability to leverage detailed instructions for more accurate and complete task execution, which indicates that our approach is both robust to instruction sparsity and highly efficient in utilizing additional information.

### 4.3. Ablation Study

To evaluate the contributions of each proposed component in our RePlan-Bot, we conduct a quantitative ablation study, with the results summarized in Table 2, where both the task description and step-by-step instructions are provided.


Figure 5. **Comparison between RePlan-Bot with and without high-level replanning.** The top row shows correct actions guided by high-level replanning, while the bottom row shows failed actions without it.

**Without High-Level Replanning.** Removing the high-level replanner consistently degrades performance: SR and GC are reduced by 3.19% and 2.40%, respectively, on the Test-Seen split, and by 2.87% and 2.42% on Test-Unseen. Without this layer, the agent tends to generate incomplete sub-goals or mis-referenced objects. High-level replanning promptly rectifies such errors, ensuring that actions remain aligned with task instructions and real-time environmental feedback throughout the episode. This global, adaptive oversight is essential for robust and efficient long-horizon execution, distinguishing our approach from conventional modular planners.

**Without Mid-Level Searching.** Table 2 shows that removing the mid-level searching module leads to a noticeable performance drop on the Test Unseen set, where the SR decreases by 1.70% and GC drops by 0.31%. Although the decline on the Test Seen set is less pronounced (SR drops by 0.91% and GC by 0.28%), the sharper degrada-


Figure 6. **Comparison between RePlan-Bot with and without mid-level searching.** The top row shows correct actions guided by commonsense-guided search mechanism, while the bottom row shows failed actions based on Unet.

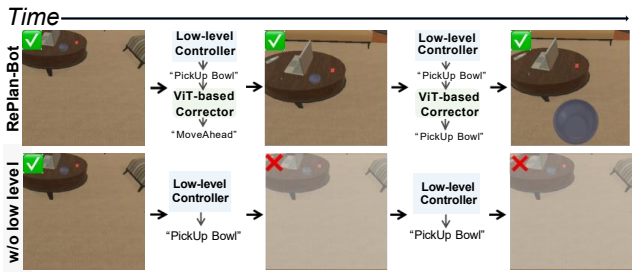


Figure 7. **Comparison between RePlan-Bot with and without low-level replanning.** With the low-level action corrector, the agent successfully picks up the bowl on the table. Without the low-level action corrector, the agent fails to do so due to being positioned too far from the table.

tion on unseen scenarios highlights the importance of mid-level reasoning when generalizing to unfamiliar environments. Meanwhile, in real-world scenarios where partial observability, occlusions, and cluttered scenes are much more prevalent, the mid-level search mechanism may be essential for reliably locating and distinguishing objects.

**Without Low-Level Replanning.** Table 2 indicates that removing the low-level action corrector leads to a clear drop in all metrics. Specifically, the SR decreases by 2.08% and GC drops by 2.19% on the Test Seen set; on the Test Unseen set, SR decreases by 2.41% and GC drops by 1.94%. This demonstrates that although predefined actions can execute simple plans, they are prone to failure in the absence of visual feedback. The vision-based corrector allows the agent to dynamically adjust actions in response to current visual feedback, preventing repeated failures such as unsuccessful pickup or unnecessary collisions. As a result, the agent achieves more efficient and robust task execution, reflected by notable improvements over both the baseline and the ablated variant in all evaluation settings.

### 4.4. Qualitative Analysis

**High-Level Replanning** Figure 5 qualitatively demonstrates the effectiveness of our high-level replanning module on representative instruction-following tasks. In the "chill a tomato and put it inside the microwave" scenario, the baseline agent without high-level replanning keeps pushing the tomato against a closed microwave and never opens the door. In contrast, our method employs an LLM auditor that reviews the initial plan, adds essential missing steps including opening the microwave and selecting the correct object, and aligns the revised sequence with the observed scene before execution. These qualitative cases highlight that high-level replanning is essential for robust execution in complex environments, especially when instructions contain sequential dependencies or linguistic ambiguity.

| Method | Test Seen | | Test Unseen | |
|---|---|---|---|---|
| | GC(PLWGC) | SR(PLWSR) | GC(PLWGC) | SR(PLWSR) |
| **RePlan-Bot** | 61.21(26.69) | 52.05(24.60) | 60.29(26.31) | 47.61(21.89) |
| w/o High-Level | 58.81(25.10) | 48.86(22.63) | 57.87(24.33) | 44.74(20.24) |
| w/o Mid-Level | 60.93(26.54) | 51.14(23.11) | 59.98(25.64) | 45.91(20.95) |
| w/o Low-Level | 59.02(24.79) | 49.97(21.76) | 58.35(23.93) | 45.20(19.22) |

Table 2. Ablation study for each proposed component.

**Mid-Level Searching** Figure 6 highlights the value of our commonsense-guided mid-level search for object retrieval. Charged with placing a credit card on the desk, the agent first receives the language model's guess that a sofa may host the card. RePlan-Bot then consults its instance-level semantic map and inspects each sofa in sequence, quickly uncovering the hidden card and completing the task (top row). Without the mid-level module, the ablated agent relies on a U-Net likelihood heat map whose coarse spatial cues misdirect the search and leave the card unfound. The example shows that instance-wise exploration delivers greater precision and robustness than heat-map methods, particularly for small or occluded objects.

**Low-Level Replanning** Figure 7 presents an example where our ViT-based low-level action corrector significantly improves execution success rates compared to the baseline without the corrector. For the "Put Bowl" action, the corrector similarly recommends corrective moves to ensure proper placement, while the baseline often fails due to poor positioning or actuation errors. These results highlight that our vision-driven low-level replanning provides crucial robustness against spatial misalignment, sensor noise, and control errors, enabling more reliable object manipulation than naive template-based controllers.

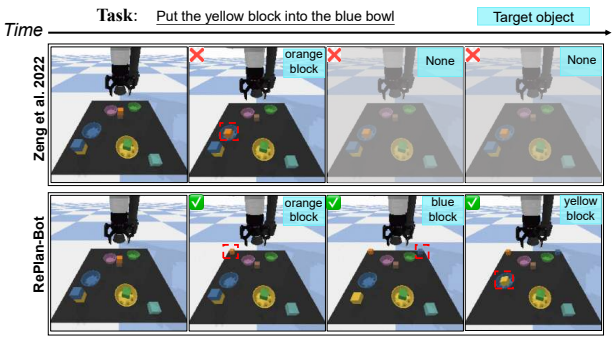


Figure 8. **An example of multi-step planning for a complex manipulation task**. The baseline model [45] fails due to a static plan that cannot handle the occluded target object. In contrast, RePlan-Bot formulates a dynamic plan to first clear the occluding blocks and successfully completes the task.

### 4.5. Application in Generalizable Robotic Manipulation

To demonstrate the generalization of RePlan-Bot beyond the ALFRED benchmark, we use a simulated tabletop environment and illustrate a comparison between RePlan-Bot and the baseline model in Figure 8. We choose [45] as the baseline model for its effectiveness in few-shot robot planning. A common methodological ground is shared between the two systems, as both utilize GPT-4o as a high-level LLM for sub-goal generation and rely on an identical privileged low-level policy for execution. Additional details are provided in Appendix 1.5.

We observe that RePlan-Bot successfully completes the complex pick-and-place task, demonstrating its ability to solve occlusion problems through high-level replanning and mid-level searching. In contrast, the baseline model fails as its static plan is incapable of generating the necessary sub-goals to first move the occluding orange and blue blocks to access the target yellow block.

## 5. Conclusion

In this paper, we propose **RePlan-Bot**, a novel EIF agent that integrates multi-level, continuous replanning to address long-horizon planning and complex environment challenges. The high-level LLM-based auditor dynamically adjusts sub-goals using environmental feedback, while the mid-level commonsense-guided search mechanism enables precise object localization with multi-layered instance maps. At the low level, the ViT-based corrector preemptively fixes risky actions using visual observations. Evaluated on the ALFRED benchmark, RePlan-Bot delivers exceptional performance across both seen and unseen environments, showcasing its strong adaptability and robustness.

**Limitations and Future Work** In real-world applications, agents often encounter unexpected events or adversarial conditions, which are not fully reflected in current simulation benchmarks. As a result, bridging the sim-to-real gap remains a significant challenge, so our future work may explore integrating reinforcement learning, domain adaptation, or more realistic simulators to enhance the agent's robustness in real-world scenarios.

## References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, Chuyuan Fu, Keerthana Gopalakrishnan, Karol Hausman, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022. 2

[2] Valts Blukis, Chris Paxton, Dieter Fox, Animesh Garg, and Yoav Artzi. A persistent spatial semantic representation for high-level natural language instruction execution. In *Conference on Robot Learning*, pages 706–717. PMLR, 2022. 2

[3] Jiaqi Chen, Bingqian Lin, Ran Xu, Zhenhua Chai, Xiaodan Liang, and Kwan-Yee K Wong. Mapgpt: Map-guided prompting with adaptive path planning for vision-and-language navigation. *arXiv preprint arXiv:2401.07314*, 2024. 2

[4] Yaran Chen, Wenbo Cui, Yuanwen Chen, Mining Tan, Xinyao Zhang, Dongbin Zhao, and He Wang. Robogpt: an intelligent agent of making embodied long-term decisions for daily instruction tasks. *arXiv preprint arXiv:2311.15649*, 2023. 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019. 2

[6] Mingyu Ding, Yan Xu, Zhenfang Chen, David Daniel Cox, Ping Luo, Joshua B Tenenbaum, and Chuang Gan. Embodied concept learner: Self-supervised learning of concepts and mapping through instruction following. In *Conference on robot learning*, pages 1743–1754. PMLR, 2023. 2

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[8] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4

[9] Yuki Inoue and Hiroki Ohashi. Prompter: Utilizing large language model prompting for a data efficient embodied instruction following. *arXiv preprint arXiv:2211.03267*, 2022. 1, 2, 3, 5

[10] Youngjoon Jeong, Junha Chun, Soonwoo Cha, and Taesup Kim. Object-centric world model for language-guided manipulation, 2025. 2

[11] Byeonghwi Kim, Suvaansh Bhambri, Kunal Pratap Singh, Roozbeh Mottaghi, and Jonghyun Choi. Agent with the big picture: Perceiving surroundings for interactive instruction following. In *Embodied AI Workshop CVPR*, page 12, 2021. 1, 2

[12] Byeonghwi Kim, Jinyeon Kim, Yuyeong Kim, Cheolhong Min, and Jonghyun Choi. Context-aware planning and environment-aware memory for instruction following embodied agents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10936–10946, 2023. 1, 2, 3, 4, 5

[13] Jinyeon Kim, Cheolhong Min, Byeonghwi Kim, and Jonghyun Choi. Pre-emptive action revision by environmental feedback for embodied instruction following agents. In *8th Annual Conference on Robot Learning*, 2024. 6

[14] Taewoong Kim, Byeonghwi Kim, and Jonghyun Choi. Multi-modal grounded planning and efficient replanning for learning embodied agents with a few examples. In *AAAI*, 2025. 2

[15] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Matt Deitke, Kiana Ehsani, Daniel Gordon, Yuke Zhu, et al. Ai2-thor: An interactive 3d environment for visual ai. *arXiv preprint arXiv:1712.05474*, 2017. 5

[16] Guanxing Lu, Ziwei Wang, Changliu Liu, Jiwen Lu, and Yansong Tang. Thinkbot: Embodied instruction following with thought chain reasoning. *arXiv preprint arXiv:2312.07062*, 2023. 2

[17] Yujie Lu, Pan Lu, Zhiyu Chen, Wanrong Zhu, Xin Eric Wang, and William Yang Wang. Multimodal procedural planning via dual text-image prompting. *arXiv preprint arXiv:2305.01795*, 2023. 2

[18] Aoran Mei, Guo-Niu Zhu, Huaxiang Zhang, and Zhongxue Gan. Replanvlm: Replanning robotic tasks with visual language models, 2024. 2

[19] So Yeon Min, Devendra Singh Chaplot, Pradeep Ravikumar, Yonatan Bisk, and Ruslan Salakhutdinov. Film: Following instructions in language with modular methods. *arXiv preprint arXiv:2110.07342*, 2021. 1, 2, 4, 5

[20] Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. Embodiedgpt: Vision-language pre-training via embodied chain of thought, 2023. 2

[21] Michael Murray and Maya Cakmak. Following natural language instructions for household tasks with landmark guided search and reinforced pose adjustment. *IEEE Robotics and Automation Letters*, 7(3):6870–6877, 2022. 1, 2, 5

[22] Van-Quang Nguyen, Masanori Suganuma, and Takayuki Okatani. Look wide and interpret twice: Improving performance on interactive instruction-following tasks. *arXiv preprint arXiv:2106.00596*, 2021. 2

[23] Alexander Pashevich, Cordelia Schmid, and Chen Sun. Episodic transformer for vision-and-language navigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15942–15952, 2021. 1, 2

[24] Yanyuan Qiao, Yuankai Qi, Zheng Yu, Jing Liu, and Qi Wu. March in chat: Interactive prompting for remote embodied referring expression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15758–15767, 2023. 2

[25] Shreyas Sundara Raman, Vanya Cohen, Eric Rosen, Ifrah Idrees, David Paulius, and Stefanie Tellex. Planning with large language models via corrective re-prompting. In *NeurIPS 2022 Foundation Models for Decision Making Workshop*, 2022. 2

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 4

[27] James A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996. 5

[28] Dhruv Shah, Błażej Osiński, Sergey Levine, et al. Lm-nav: Robotic navigation with large pre-trained models of language, vision, and action. In *Conference on robot learning*, pages 492–504. PMLR, 2023. 2

[29] Suyeon Shin, Sujin Jeon, Junghyun Kim, Gi-Cheon Kang, and Byoung-Tak Zhang. Socratic planner: Inquiry-based zero-shot planning for embodied instruction following. *CoRR*, 2024. 1

[30] Suyeon Shin, Sujin jeon, Junghyun Kim, Gi-Cheon Kang, and Byoung-Tak Zhang. Socratic planner: Self-qa-based zero-shot planning for embodied instruction following, 2025. 2

[31] Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10740–10749, 2020. 1, 2, 5

[32] Ishika Singh, Valts Blukis, Arsalan Mousavian, Ankit Goyal, Danfei Xu, Jonathan Tremblay, Dieter Fox, Jesse Thomason, and Animesh Garg. Progprompt: Generating situated robot task plans using large language models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11523–11530. IEEE, 2023. 2

[33] Marta Skreta, Zihan Zhou, Jia Lin Yuan, Kourosh Darvish, Alán Aspuru-Guzik, and Animesh Garg. Replan: Robotic replanning with perception and language models. *arXiv preprint arXiv:2401.04157*, 2024. 2

[34] Chan Hee Song, Jihyung Kil, Tai-Yu Pan, Brian M Sadler, Wei-Lun Chao, and Yu Su. One step at a time: Long-horizon vision-and-language navigation with milestones. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15482–15491, 2022. 2

[35] Chan Hee Song, Jiaman Wu, Clayton Washington, Brian M Sadler, Wei-Lun Chao, and Yu Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2998–3009, 2023. 1, 2, 3

[36] Alessandro Suglia, Qiaozi Gao, Jesse Thomason, Govind Thattai, and Gaurav Sukhatme. Embodied bert: A trans-

former model for embodied, language-guided visual task completion. *arXiv preprint arXiv:2108.04927*, 2021. 1, 2

[37] Fangyuan Wang, Shipeng Lyu, Peng Zhou, Anqing Duan, Guodong Guo, and David Navarro-Alarcon. Instruction-augmented long-horizon planning: Embedding grounding mechanisms in embodied mobile manipulation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 14690–14698, 2025. 2

[38] Zihao Wang, Shaofei Cai, Guanzhou Chen, Anji Liu, Xiaojian Ma, and Yitao Liang. Describe, explain, plan and select: Interactive planning with large language models enables open-world multi-task agents. *arXiv preprint arXiv:2302.01560*, 2023. 2

[39] Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Jiwen Lu, and Haibin Yan. Embodied task planning with large language models. *arXiv preprint arXiv:2307.01848*, 2023. 2

[40] Zhenyu Wu, Ziwei Wang, Xiuwei Xu, Hang Yin, Yinan Liang, Angyuan Ma, Jiwen Lu, and Haibin Yan. Embodied instruction following in unknown environments. *arXiv preprint arXiv:2406.11818*, 2024. 2

[41] Yuxiao Yang, Shenao Zhang, Zhihan Liu, Huaxiu Yao, and Zhaoran Wang. Hindsight planner: A closed-loop few-shot planner for embodied instruction following. *arXiv preprint arXiv:2412.19562*, 2024. 1

[42] Hang Yin, Xiuwei Xu, Zhenyu Wu, Jie Zhou, and Jiwen Lu. Sg-nav: Online 3d scene graph prompting for llm-based zero-shot object navigation. *Advances in Neural Information Processing Systems*, 37:5285–5307, 2024. 2

[43] Takuma Yoneda, Jiading Fang, Peng Li, Huanyu Zhang, Tianchong Jiang, Shengjie Lin, Ben Picker, David Yunis, Hongyuan Mei, and Matthew R. Walter. Statler: State-maintaining language models for embodied reasoning, 2024. 2

[44] Bangguo Yu, Hamidreza Kasaei, and Ming Cao. L3mvn: Leveraging large language models for visual target navigation. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3554–3560. IEEE, 2023. 2

[45] Andy Zeng, Maria Attarian, Brian Ichter, Krzysztof Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. *arXiv*, 2022. 8

[46] Yichi Zhang and Joyce Chai. Hierarchical task learning from language instructions with unified transformers and self-monitoring. *arXiv preprint arXiv:2106.03427*, 2021. 2

[47] Gengze Zhou, Yicong Hong, and Qi Wu. Navgpt: Explicit reasoning in vision-and-language navigation with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 7641–7649, 2024. 2

[48] Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Yu Qiao, Zhaoxiang Zhang, and Jifeng Dai. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144*, 2023. 2