Pre-emptive Action Revision by Environmental Feedback for Embodied Instruction Following Agents

Anonymous Author(s)
Affiliation
Address
email

Abstract: When we, humans, conduct a task, we consider changes in environments such as objects' arrangement due to interactions with objects and other reasons; *e.g.*, when we find a mug to clean, but it is already clean. Then we skip cleaning it. The plasticity of the human brain allows us to adapt to environmental states but current embodied agents often ignore the changed environments when conducting a task, leading to failure of task completion or executing unnecessary actions. Here, we propose Revising actions by Environmental feeDback (RED) that allows an embodied agent to revise their action in response to perceived environmental status "before it makes mistakes." We empirically validate our RED and observe that our RED outperforms prior arts on two challenging benchmarks, TEACh and ALFRED, by noticeable margins in most metrics, including unseen success rates, with shorter execution length, *i.e.*, an efficiently behaving agent.

Keywords: Replanning, Environmental Feedback, Brain plasticity, Embodied AI

1 Introduction

Building robotic assistants that can understand natural language and the surroundings and perform the desired tasks has long been an ambitious goal in the research community. For these assistants, recent advances in related domains such as computer vision [1, 2, 3] and natural language processing [4, 5, 6, 7] have been actively integrated into the learned agents. Subsequently, these learned agents engage in various tasks [8, 9, 10, 11, 12, 13, 14] within diverse environments [15, 16, 17]. To complete the desired tasks, agents typically generate their initial plans based on the anticipated environmental states at the beginning [18, 19] and execute them.

However, the environments may change, making it difficult for embodied agents to complete a task due to the discrepancy between the environment that the agent expects and the actual environment. This discrepancy results in misperception and incomplete exploration, *etc*. In contrast to the artificial agents, humans and animals can effectively adapt to these environmental changes through *brain plasticity* [20, 21, 22]. This plasticity rewires their brains based on experiences with varying environmental conditions, allowing them to adjust their behaviors to those performed previously to prevent mistakes in advance. In light of this, we pose the question: *Can artificial embodied agents also derive benefits from this brain plasticity for environmental discrepancies?*.

Drawing inspiration from neuroscience, we propose Revising actions by Environmental feeDback (RED), an instruction following embodied agent that can adjust their behaviors by perceiving environmental discrepancies as environmental feedback by common sense learned in large language models (LLM) to review the current plan based on this feedback. For environmental discrepancies, considering that object perception plays an important role in numerous embodied tasks, we particularly focus on four distinct environmental discrepancies concerning objects: 1) object presence [23, 24], 2) object appearance [25], 3) object attributes [26, 27], and 4) object-object relationships [28, 29].

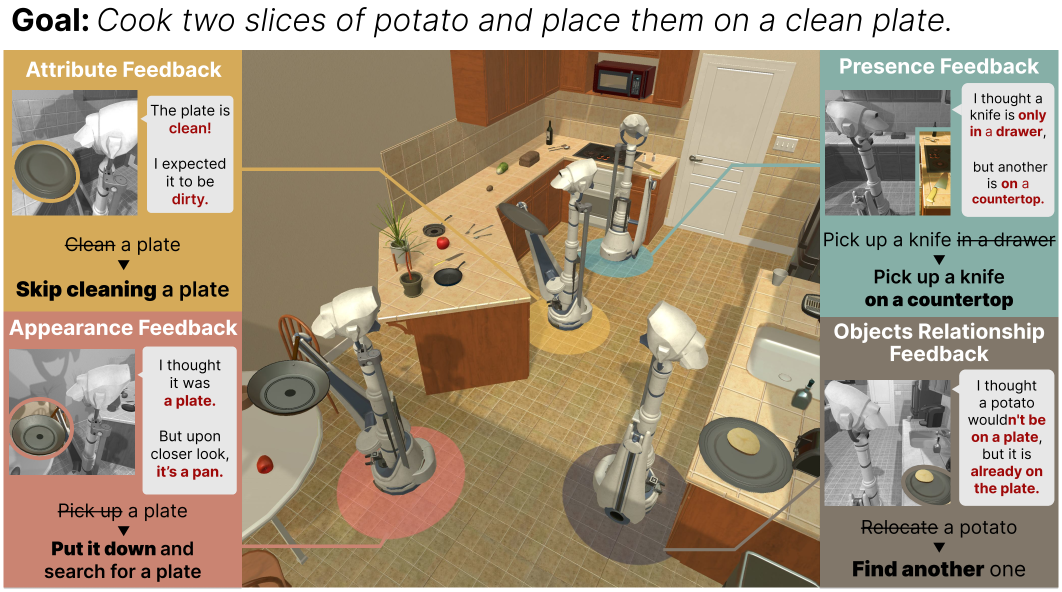
Goal: Cook two slices of potato and place them on a clean plate.
Attribute Feedback
The plate is clean! I expected it to be dirty.
~~Clean~~ a plate
Skip cleaning a plate
Appearance Feedback
I thought it was a plate. But upon closer look, it's a pan.
~~Pick up~~ a plate
Put it down and search for a plate
Presence Feedback
I thought a knife is only in a drawer, but another is on a countertop.
Pick up a knife ~~in a drawer~~
Pick up a knife on a countertop
Objects Relationship Feedback
I thought a potato wouldn't be on a plate, but it is already on the plate.
~~Relocate~~ a potato
Find another one

Figure 1: **Overview of the proposed method (RED).** The agent adapts to unexpected environmental discrepancies: it skips cleaning an already clean plate, puts down a pan when a plate is needed, grabs a knife from the countertop without further searching, and ignores moving a potato that is already on its destination plate. Guided by LLMs, the agent effectively achieves the goal of serving two slices of potato on a clean plate.

To address these discrepancies, we propose four components: Dynamic Target Adaptation (DTA) that dynamically modifies navigation targets using object presence discrepancies, Object Heterogeneity Verification (OHV) that verifies whether an interacted object is intended by examining object appearance discrepancies between the initial and subsequent perceptions of the object, Attribute-Driven Plan Modification (APM) that modifies the original state-changing actions, such as cleaning an object, using object attribute discrepancy, and Action Skipping by Relationship (ASR) that refrains agents from taking unintended actions using object relationship discrepancy. In contrast to previous approaches [26, 30] that revise their original plans after encountering failures, we preemptively revise them to avoid nonrecoverable failures such as irreversible state transitions [12, 13, 31].

We empirically validate our RED in two challenging benchmarks, TEACh [13] and ALFRED [12], for embodied instruction following. We observe that our RED outperforms prior arts notably in most metrics, including the unseen success rates, which is the main metric of [12, 13].

2 Related Work

Embodied instruction following agents. Developing agents that can achieve the desired goals by understanding natural language has been a daunting challenge. To develop such agents, previous benchmarks [32, 33, 34] require agents to understand natural language and navigate to a designated target location. For example, [32] requires a robot agent to infer proper next steps towards the goal with a given natural language history, and [33] requires allowing a tourist to reach a specified destination. However, they primarily focus on learning navigation agents without object interaction, hindering the deployability of an agent to complex tasks (*e.g.*, an agent needs to prepare breakfast by heating a bread slice with a toaster and making a cup of coffee using a coffee machine).

To address more complex tasks beyond navigation, recent benchmarks [13, 14] incorporate object interaction into their task setups to require agents to complete tasks by understanding free-form language that describes the tasks and interacting with relevant objects. For these tasks, early approaches [35, 14] learn a direct mapping from multimodal input (*i.e.*, egocentric observation, and language instructions) to the corresponding actions and object locations in the egocentric frames.

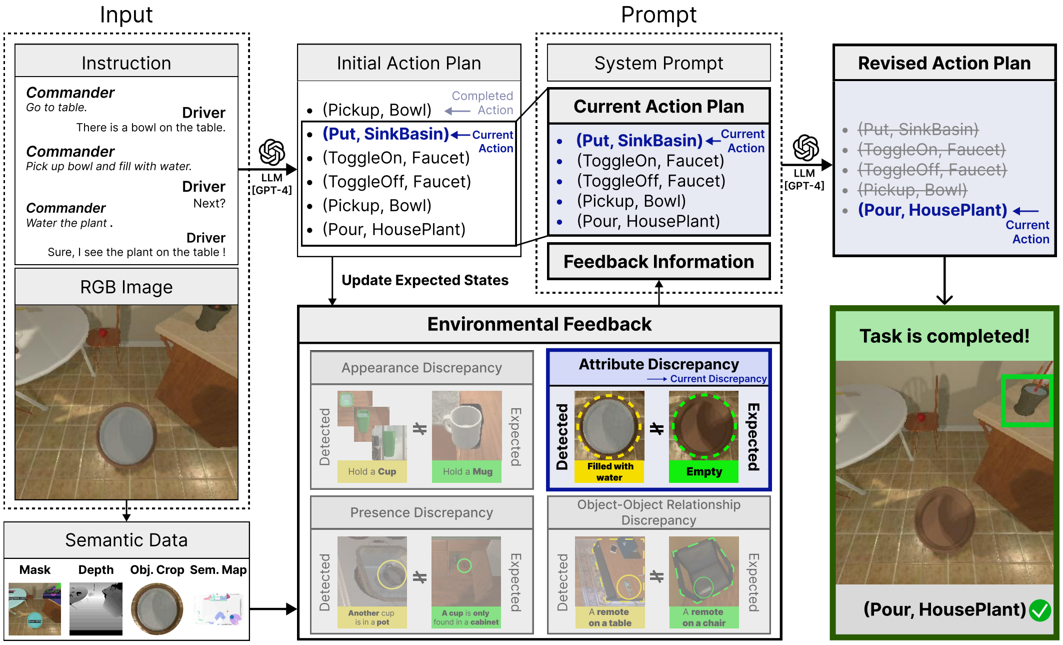

Figure 2: **Workflow of the proposed Revising actions by Environmental feeDback (RED).** We first generate an initial plan based on the language instruction using a large language model (LLM) and anticipate initial and target object states to achieve a desired goal. At the same time, we predict semantic information and obtain the actual environmental states from these semantics. We then compare the anticipated and actual environmental states and maintain the discrepancies between them. If discrepancies are detected, we use them as environmental feedback, denoted by 'Feedback Information,' to revise the original plan, denoted by 'Current Action Plan.'

However, these learning-based approaches usually require a large number of training episodes for good performance, but collecting them is expensive and sometimes impossible.

To address this data-scarcity issue, recent approaches [36, 37, 26, 30] use deterministic algorithms, such as A* or FMM [38], for accurate obstacle-free navigation on semantic spatial maps [39, 18, 37, 30], significantly improving their performance. Inspired by this, we also exploit [38] for navigation.

Planning and revising using large language models. By the help of large language models (LLMs) for task planning or revision of plans, recent work [30, 40, 41] is studying plan revision. They correct their original plans when agents fail to execute actions or determine whether they fail to achieve desired outcomes. For example, [30] retrieves the most relevant top-K examples of error correction for a failure case, and an LLM generates corrected programs for action generation. [40] condenses a robot's previous experiences to identify failures and devises new plans to rectify them. While these approaches correct the action sequence by the failure, we preemptively predict failures before they occur and revise the action sequence, saving unnecessary actions.

Other work [42, 43, 44] uses LLMs for self-corrective ability by receiving environmental feedback, but depends on ground-truth feedback from environments associated with a small number of actions. For instance, a recent approach [45] exploits either ground truth or expensive human feedback. However, this may pose limited deployability in environments without such ground-truth information.

Another line of work [46, 47] tries to solve task planning and failure correction by focusing on errors. [47] proposes to generate complete plans using a structured programming language prompt, with each action considering conditions for error-free execution. [46] proposes correcting the error or preventing the failure using LLMs with predefined preconditions. However, both have not explored the strategy of adaptively skipping unnecessary actions to enhance mission efficiency. Our approach is designed to allow agents to behave efficiently in the environment, akin to humans, by adapting actions to suit the environmental context without the information about why the task failed.

3 Approach

Recent approaches in building an embodied agents use LLMs [30, 41, 48] for commonsense reasoning and semantic spatial maps [37, 39, 18] for path planning. However, these agents often encounter scenarios where their anticipated environmental states differ from the ones they actually perceive. These scenarios can result in unintended or incorrect actions and consequently lead to inefficiency or even failure in completing tasks. Inspired by humans and animals that adapt to changing environments, we propose RED that pre-emptively adjusts plans using environmental discrepancies as feedback. Figure 2 illustrates the workflow of our RED.

3.1 Revising Actions using Environmental Feedback

We aim to learn a policy that generates a modified action sequence for embodied agents [12, 13] by perceiving environmental changes. Specifically, we learn a function $f_s : \mathcal{V} \times \mathcal{A} \rightarrow \mathcal{A}$, that maps a visual observation set, $\{v_m\}_{m=1}^{M}$, and an (N-length) action sequence, $\{a_n\}_{n=1}^{N}$, into a modified (K-length) action sequence, $\{a'_k\}_{k=1}^{K}$. $\mathcal{V}$ and $\mathcal{A}$ denotes all possible observation and action sets.

We consider an embodied agent, a function, $\pi_\theta : \mathcal{V} \times \mathcal{X} \rightarrow \mathcal{A}$, that maps a natural language instruction, $x \in \mathcal{X}$, and the current visual observation, $v_t \in \mathcal{V}$, into an action, $a_t \in \mathcal{A}$. We expect the generated action sequence, $\{a_t\}_{t=1}^{T}$, to transform the initial environment state, $s_i \in \mathcal{S}$, to the desired state, $s_f \in \mathcal{S}$, where each $\mathcal{X}$ and $\mathcal{S}$ denotes a set of all possible natural language instructions and environment states and T a time budget. We detail our agent's architecture in the supplementary.

In particular, when an agent encounters unexpected scenarios caused by 'differences' between those inferred from a language description and observed in the environment, referred to as 'environmental discrepancies,' RED revises the agent's original plan, $\{a_n\}_{n=1}^{N}$, by querying a large language model, $\mathcal{L}$, with a prompt, $\mathcal{P}$. $\mathcal{P}$ concatenates a system prompt, $\mathcal{P}_s$, for a general description and guide of the task, the original plan, and a feedback prompt, $\mathcal{P}_f$, which describes the discrepancy encountered as environmental feedback. Then $\mathcal{L}$ receives $\mathcal{P}$ and produces a revised plan, $\{a'_k\}_{k=1}^{K}$ as:

$$\{a'_k\}_{k=1}^{K} = \mathcal{L}(\mathcal{P}) \quad \text{where} \quad \mathcal{P} = [\mathcal{P}_s; \{a_n\}_{n=1}^{N}; \mathcal{P}_f]. \tag{1}$$

To build a feedback prompt, we consider four types of environmental discrepancies caused by the presence, appearance, attributes, and relationships of objects based on visual information that occupies a large proportion of sensory information perceived by humans [49, 50]. We propose four modules for respective discrepancies: Dynamic Target Adaptation (DTA), Object Heterogeneity Verification (OHV), Attribute-Driven Plan Modification (APM), and Action Skipping by Relationship (ASR). We detail the proposed modules below and provide their examples of system prompt, current action plan, feedback information, and an LLM output in listings in the supplementary.

3.1.1 Presence Discrepancy → Dynamic Target Adaptation

When an agent is conducting a task, it may encounter scenarios where a target object is present in unexpected places. For example, in the task, 'boil a potato in the refrigerator using a pot,' the agent may expect the refrigerator, possibly containing the potato, as its navigational target. However, the agent may find another potato on different objects, such as a table. In this case, revising the action plan by adapting to environments where target objects can be easily found may improve the efficiency and effectiveness of navigation. Here, we denote by an object presence discrepancy the difference between an expected location and an actual one for a target object to be found.

To address this, we propose Dynamic Target Adaptation (DTA) which detects an object presence discrepancy and provides a feedback prompt describing it. For this, the agent first compares an inferred place of a target object, o, with one perceived and maintained in the agent's memory (see the supplementary for more details), Z_t, at the current time step, t. If the agent has previously observed the target object in a different place from the inferred one (*i.e.*, $o \in Z_t$) before reaching it, DTA returns the presence discrepancy as a feedback prompt, $\mathcal{P}_f$, indicating that the target object is not in the expected location (*e.g.*, $\mathcal{P}_f =$ "`the object is found in another place, not in the receptacles that should be opened.`").

3.1.2 Appearance Discrepancy → Object Heterogeneity Verification

An agent often fails to interact with task-relevant objects due to misperceptions influenced by lighting, occlusions, and varying appearances from different viewpoints. This issue can be mitigated by examining objects from multiple perspectives. For example, an agent might mistake a cup for a mug due to object recognition errors from a far distance.

To address this issue, we propose Object Heterogeneity Verification (OHV), which recognizes appearance discrepancies and provides feedback accordingly. Here, we define an appearance discrepancy as the difference in an object's predicted identities (*i.e.*, classes) from the appearances observed in various viewpoints. The proposed verification requires the agent to pick up objects and change its view. That is, when the agent interacts with an object, it verifies if the object is intended one by comparing its predicted classes that can be different due to varying appearances from various viewpoints. These actions allow the agent to see the object without occlusion, thus encouraging it to properly identify the object (*i.e.*, a mug) by the observations from various viewpoints.

Formally, let c_i be a predicted object class from the appearance of a i^{th} viewpoint of an object, where $i \in \{0, \cdots, I\}$ denotes a viewpoint index among predefined I viewpoints. $i = 0$ is the viewpoint at the time of interaction (see the supplementary for more details). If the agent encounters a different predicted class (*i.e.*, $c_0 \neq c_i$) from any different viewpoint $i > 0$, contrary to the expected class (*i.e.*, $\forall i > 0 : c_0 = c_i$), OHV detects an appearance discrepancy and specifies this in a feedback prompt, $\mathcal{P}_f$, indicating that the object is not intended (*e.g.*, for the example above, $\mathcal{P}_f =$ “`It turns out that the object picked up is not the intended object.`”).

3.1.3 Attribute Discrepancy → Attribute-Driven Plan Modification

Language instructions often lack detailed environmental descriptions, potentially causing the agent to do redundant actions or miss important actions due to unknown object attributes. Here, an *attribute* refers to the physical state of an object depending on its affordances[1] [13, 26]. We can mitigate redundant or missed actions from unknown attributes by checking the target object, o, its expected attribute, $\hat{\phi_o}$, and its actual observed attribute, ϕ_o.

For example, in a task, “clean a mug and fill it with coffee,” the agent expects that o (*i.e.*, a mug) needs to be cleaned because it is dirty, so the expected attribute of the target can be represented as $\hat{\phi_o} = \{\text{Dirty}\}$. However, the agent might find a clean mug during exploration, where the detected target object's attribute is clean (*i.e.*, $\phi_o = \{\text{Clean}\}$). In this case, the agent can skip redundant actions in the original plan (*i.e.*, cleaning the mug first), allowing efficient task completion. We denote an attribute discrepancy as the difference between the expected and observed attributes of an object.

For this, we propose Attribute-Driven Plan Modification (APM) to detect attribute discrepancies and provide environmental feedback. When the agent detects o, it captures a cropped image exclusively of the target from the current view. Then, we utilize an attribute detector, $\mathcal{H}$, which takes the cropped image v_o as input and predicts ϕ_o (see the supplementary for more details). If ϕ_o from $\mathcal{H}$ does not match $\hat{\phi_o}$ (*i.e.*, $\hat{\phi_o} \neq \phi_o$), APM describes this attribute discrepancy as a feedback prompt $\mathcal{P}_f$ (*e.g.*, “`After checking, it appears that the target has already been cleaned.`”).

3.1.4 Object-Object Relationship Discrepancy → Action Skipping by Relationship

Rearranging objects [10, 51] often poses challenges when the agent relocates multiple objects, potentially with the same look, of the same class, making it difficult for the agent to decide which object to move. To address this, the agent considers the current relationship, r_o, and the expected relationship, $\hat{r}_o$, between the target object, o, and its placement object, o_p. Here, the relationship represents the spatial relationship between o and o_p. We write this as $r_o = (o, o_p)$.

For instance, if instructed to “place two pillows on the sofa,” the agent assumes that o (*i.e.*, a pillow) is not on the final place (*i.e.*, sofa) and needs to be moved. Thus, the expected relationship can

[1] We use object attributes supported by AI2-THOR [15] on which our evaluation benchmarks [12, 13] run.

Table 1: **Comparison with the state of the arts in the TEACh benchmark.** The path-length-weighted (PLW) metrics are given in the parentheses for each value. The highest and second highest values per fold and metric are shown in **bold** and underline, respectively.

Model	TfD				EDH			
	Unseen		Seen		Unseen		Seen	
	SR	GC	SR	GC	SR	GC	SR	GC
E.T [35]	0.48 (0.12)	0.35 (0.59)	1.02 (0.17)	1.42 (4.82)	7.80 (0.90)	9.10 (1.70)	10.20 (1.70)	15.70 (4.10)
JARVIS [36]	1.80 (0.30)	3.10 (1.60)	1.70 (0.20)	5.40 (4.50)	15.80 (2.60)	16.60 (8.20)	15.10 (3.30)	22.60 (8.70)
FILM [37]	2.90 (1.00)	6.10 (2.50)	5.50 (2.60)	5.80 (11.60)	10.20 (1.00)	18.30 (2.70)	14.30 (2.10)	26.40 (5.60)
DANLI [26]	7.98 (3.20)	6.79 (6.57)	4.97 (1.86)	10.50 (10.27)	16.98 (7.24)	23.44 (19.95)	17.76 (9.28)	24.93 (22.20)
HELPER [30]	13.73 (1.61)	14.17 (4.56)	12.15 (1.79)	18.62 (9.28)	17.40 (2.91)	25.86 (7.90)	18.59 (4.00)	32.09 (9.81)
RED (Ours)	**19.77** (5.16)	**16.74** (8.31)	**20.99** (4.64)	**21.55** (11.03)	**21.69** (4.44)	**26.83** (7.46)	**21.71** (4.62)	**32.78** (10.39)

be defined as the pillows that are not on the sofa, represented as $\hat{r_o} = \neg(pillow, sofa)$. However, during exploration, if the agent detects a pillow on the sofa, the current relationship becomes $r_o = (pillow, sofa)$, *i.e.*, $\hat{r}_o \neq r_o$, and a relationship discrepancy occurs. If the agent ignores such discrepancy and continues to interact with o, *i.e.*, meaninglessly moving the pillow from the sofa to the sofa, it may not be able to complete the task.

To mitigate meaningless relocation, we propose Action Skipping by Relationship (ASR), which detects the relationship discrepancy and provides a corresponding feedback prompt. To obtain r_o, we first predict the masks of objects from the current egocentric view, $\{m_i\}_{i=0}^{N}$, where m_0 denotes the mask of o. We then find the most 'adjacent' mask, m_i, of o_p, to m_0 and regard that o is currently on o_p (see the supplementary for more details). If the currently detected target object is already in the final place (*i.e.*, $\hat{r_o} \neq r_o$), ASR describes this relationship discrepancy in a feedback prompt $\mathcal{P}_f$, such as ``After checking the object and its location, it is observed that the target object is already in the desired location''

4 Experiments

We briefly explain the benchmarks, baselines, and evaluation metrics used for the experiments. For more details on the benchmarks and baselines, kindly refer to the supplementary.

Benchmarks. We employ two challenging long-horizon instruction following benchmarks for embodied agents, TEACh [13] and ALFRED [12]. In TEACh, we evaluate our RED in two sub-benchmarks, Trajectory from dialog (TfD) and Execution from Dialog History (EDH). TfD requires the agent to solve long-horizon household tasks by understanding dialogs, while EDH requires performing a session-specific portion of the TfD tasks. ALFRED provides declarative instructions consisting of a goal statement and step-by-step instructions that describe how to complete a task.

Baselines. We compare RED with recent state-of-the-art methods as baselines for both benchmarks. For the TEACh benchmark, we compare ours with FILM [37], DANLI [26], and HELPER [30]. For the ALFRED benchmark, we adopt HLSM [39], FILM [52], and CAPEAM [18] as baselines.

Metrics. The primary metric is the success rate (SR), the ratio of the completed tasks. The goal condition success rate (GC) denotes the ratio of the satisfied goal conditions. To measure efficiency, the path-length-weighted (PLW) score penalizes SR and GC by the length of the actions taken.

4.1 Comparison with State of the Art

We compare RED with prior state-of-the-art methods on the TEACh and ALFRED benchmarks summarized in Table 1 and Table 2, respectively. Additionally, we provide the results of a new validation and test set in EDH, used exclusively by DANLI [26] and re-split from the original validation set, to ensure a fair comparison (see the result table in the supplementary). RED outperforms other baseline models in both benchmarks by noticeable margins in SR and GC, demonstrating its efficacy.

In the TEACh benchmark, in TfD and EDH setups, we observe that RED outperforms the previous methods in unseen/seen environments for SR and GC, which implies the effectiveness of our proposed RED. In addition, our model shows a larger performance gap between ours and prior art on TfD, a more changeable setup than on EDH. This may be because TfD requires more interactions with objects than EDH, which performs a specific portion of the TfD tasks. Thus, our proposed methods have more opportunities to be applied, leading to greater improvement in TfD compared to EDH.

We observe that DANLI [26] achieves better PLW scores in the EDH setup. We believe that its 3D map tracking each instance's location, including height, eliminates the need for vertical scanning in navigation, unlike our top-down 2D map. In addition, its recovery plans' effectiveness is based on a lot of human-defined plans for all exceptions, improving PLW scores.

Table 2 shows the prior arts and RED's performance in the ALFRED benchmark with a few different settings. We include the 'Reproduced' section because the reproduced results of previous methods differ slightly from the originally reported ones. As shown in Table 1, we observe that our method outperforms the prior arts in all metrics, implying the effectiveness of the proposed components.

Table 2: **Comparison with the state of the arts in the ALFRED benchmark.** The path-length-weighted (PLW) metrics are given in the parentheses for each value. The highest and second highest values per fold and metric are shown in **bold** and $\underline{\text{underline}}$, respectively. 'Reported' and 'Reproduced' sections show the methods' performances from the paper and our reproduction results. 'Reproduced w/ Dynamic Initial States' shows performances in dynamic initial states (Section 4.1).

Model	**Test Unseen**		**Test Seen**	
	SR	**GC**	**SR**	**GC**
Reported				
HLSM [39]	20.27 (5.55)	30.31 (9.99)	29.94 (8.74)	41.21 (14.58)
FILM [52]	24.46 (9.67)	34.75 (13.13)	25.77 (10.39)	36.15 (14.17)
CAPEAM [18]	43.69 (17.64)	54.66 (22.76)	47.36 (19.03)	54.38 (23.78)
Reproduced				
HLSM [39]	21.32 (5.89)	31.09 (10.39)	31.90 (9.75)	43.22 (15.30)
FILM [52]	23.61 (15.10)	36.90 (12.99)	25.77 (10.58)	35.43 (14.62)
CAPEAM [18]	$\underline{41.79}$ (18.07)	$\underline{53.93}$ (23.41)	$\underline{45.14}$ (18.79)	$\underline{52.82}$ (23.25)
RED (Ours)	**46.96** (20.58)	**57.35** (24.72)	**51.40** (21.14)	**59.04** (25.52)
Reproduced w/ Dynamic Initial States				
HLSM [39]	19.03 (5.60)	28.21 (9.64)	23.27 (7.59)	31.56 (12.60)
FILM [52]	15.17 (7.43)	22.74 (12.14)	13.89 (5.68)	22.80 (10.26)
CAPEAM [18]	$\underline{24.00}$ (9.39)	$\underline{31.92}$ (14.85)	$\underline{25.18}$ (10.78)	$\underline{32.97}$ (15.96)
RED (Ours)	**32.31** (12.48)	**42.62** (17.87)	**35.09** (15.02)	**43.27** (19.95)

Dynamic initial states. In the ALFRED benchmark, the states of all objects are 'static' at the beginning of every task, indicating that the initial states of objects are always 'fixed' once a task for the agent to perform is given. For example, in the task of moving a cleaned mug, the initial states of all mugs in the environment are always set to be 'dirty.' However, this evaluation does not fully address environmental discrepancies caused by different initial states of objects (*e.g.*, cleaning a mug that is already clean), potentially resulting in unexpected scenarios.

To further address these discrepancies in the ALFRED benchmark, we intentionally modify the initial states of objects to have diverse ones and denote these modified ones as 'dynamic initial states.' For example, in the task of moving a cleaned mug above, the agent may encounter already cleaned mugs and the agent can revise its plan in this case for efficient and effective task completion. We observe that our RED outperforms all prior approaches in the dynamic setting, implying that our RED has better capability to adapt to environments with objects' dynamic initial states.

4.2 Ablation Study

To investigate the impact of each proposed component, DTA, OHV, APM, and ASR, we conduct ablation studies and summarize the results in Table 3 and Table 4. In our experiments, no simultaneous environmental discrepancies were detected, but our RED is designed to handle multiple discrepancies at once. Here, '(x) *vs.* (y),' denotes a comparison between the x and y rows in Table 3 and Table 4.

No DTA. We observe that ablating DTA ((a) *vs.* (b)) leads to noticeable drops (up to 4.97% in TfD seen) in all metrics in seen and unseen splits for TfD and EDH. We believe that our agent without DTA does not consider object presence discrepancies, causing repeated unnecessary object interaction. This can increase the chance of interaction failure and thus, reduce task success rates.

No OHV. Second, we ablate OHV ((a) *vs.* (c)) in our RED to assess the impact of considering differences in object appearances. Ablating OHV results in significant performance drops, with a 4.97% SR decrease in the TfD seen and a 3.01% SR decrease in the ALFRED test unseen. We empirically observe that the agent suffers from misperceptions such as light reflection and occlusion, resulting in the agent interacts with irrelevant objects. Unlike the agent with OHV, the ablated agent

Table 3: **Ablation study in TEACh for each proposed component.** The path-length-weighted (PLW) metrics are given in the parentheses for each value. (b) to (e) show the performances of RED without each component.

Model	**TfD**				**EDH**			
	Unseen		**Seen**		**Unseen**		**Seen**	
	SR	GC	SR	GC	SR	GC	SR	GC
(a) **RED**	19.77 (5.16)	16.74 (8.31)	20.99 (4.64)	21.55 (11.03)	21.69 (4.44)	26.83 (7.46)	21.71 (4.62)	32.78 (10.39)
(b) w/o DTA	17.65 (4.63)	13.39 (6.63)	16.02 (4.05)	16.34 (8.13)	20.67 (3.52)	25.74 (6.62)	18.75 (3.59)	30.95 (8.69)
(c) w/o OHV	15.20 (4.22)	12.58 (6.57)	16.02 (4.56)	17.46 (9.68)	20.35 (3.44)	22.89 (6.66)	18.75 (3.87)	29.87 (9.60)
(d) w/o APM	16.18 (4.14)	13.66 (6.45)	14.36 (3.70)	18.59 (9.10)	19.89 (3.09)	24.93 (5.85)	19.98 (3.67)	29.42 (7.40)
(e) w/o ASR	17.16 (4.44)	15.30 (7.41)	18.23 (5.11)	17.18 (8.92)	20.49 (4.06)	26.00 (7.42)	18.91 (2.53)	30.62 (9.38)

proceeds with the task without checking whether it interacted with the correct object. As a result, errors due to misperceptions directly lead to task failure.

No APM. When APM is absent from RED ((a) *vs.* (d)), it may not be trivial for the agent to discern changes in object attributes, potentially taking unnecessary actions or omitting key actions. The results show that ablating APM results in significant performance drops in all metrics by a large margin in TEACh (a downturn peaking at a 6.63% SR in the seen environments in both TfD and EDH) and in ALFRED (7.98% SR in test unseen). This oversight in object attributes can cause the agent to unnecessarily change an object's state (*e.g.*, cleaning an already cleaned object), even if it already meets the target state. Additionally, the agent repeatedly attempts infeasible actions to the objects that are not in an available state for interaction, which increases failures and time steps, thereby reducing the likelihood of task completion.

No ASR. Finally, we ablate ASR from our agent ((a) *vs.* (e)). Without ASR, the agent cannot comprehend the relationships between objects, which makes it attempt to move the objects again that already satisfy the goal relationship (*e.g.*, a *Plate* should be in *Sink*). This may lead to achieving insufficient goal states as the agent moved 'two' objects but actually, it did only 'one' object. Due to this, we observe that 2.76% SR of drop in TfD seen and 4.06% SR of drop in ALFRED test unseen.

Table 4: **Ablation study in ALFRED for each proposed component.** The path-length-weighted (PLW) metrics are given in the parentheses for each value. (b) to (e) show the performances of RED without each component.

Model	**Test Unseen**		**Test Seen**	
	SR	**GC**	**SR**	**GC**
(a) **RED**	32.31 (12.48)	42.62 (17.87)	35.09 (15.02)	43.27 (19.95)
(b) w/o DTA	29.63 (11.79)	39.49 (16.66)	33.59 (13.92)	42.02 (18.69)
(c) w/o OHV	29.30 (11.33)	38.95 (16.54)	33.53 (13.91)	42.75 (19.07)
(d) w/o APM	24.33 (9.13)	32.11 (13.98)	24.66 (10.01)	32.42 (15.00)
(e) w/o ASR	28.25 (11.22)	38.98 (17.52)	33.07 (14.48)	42.95 (19.63)

4.3 Qualitative Analysis

We qualitatively investigate DTA, OHV, APM, and ASR in the supplementary for the sake of space.

5 Conclusion

We propose RED that adjusts their behaviors based on perceived environmental discrepancies inspired by brain plasticity of humans and animals before failure. Given perceived environmental discrepancies, RED builds a prompt comprising its current plan, the discrepancies, and a system prompt including a task objective, and queries large language models (LLMs) to generate a revised plan.

To address these environmental discrepancies, we propose DTA, OHV, APM, and ASR for replanning for effective and efficient task completion. We observe that our RED outperforms previous methods notably in two challenging embodied instruction following benchmarks, TEACh and ALFRED.

Limitations and future work. The environmental discrepancies are perceived based on semantic information (*e.g.*, object masks, semantic spatial maps, *etc*) predicted from a single egocentric observation and therefore may not be accurate, possibly leading to inaccurate plan modification. A promising future direction is to modify a plan even with these possibly inaccurate discrepancies. In addition, we plan to extend our RED, potentially with strong foundation models [53] for richer and more accurate environmental feedback, to real-robot setups.

References

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021.

[3] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021.

[4] S. Hochreiter and J. Schmidhuber. Long short-term memory. In *Neural Computation*, 1997.

[5] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*, 2014.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[7] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

[8] P. Anderson, Q. Wu, D. Teney, J. Bruce, M. Johnson, N. Sünderhauf, I. Reid, S. Gould, and A. van den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *CVPR*, 2018.

[9] J. Krantz, E. Wijmans, A. Majumdar, D. Batra, and S. Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, 2020.

[10] L. Weihs, M. Deitke, A. Kembhavi, and R. Mottaghi. Visual room rearrangement. In *CVPR*, 2021.

[11] K. Ehsani, W. Han, A. Herrasti, E. VanderBilt, L. Weihs, E. Kolve, A. Kembhavi, and R. Mottaghi. Manipulathor: A framework for visual object manipulation. In *CVPR*, 2021.

[12] M. Shridhar, J. Thomason, D. Gordon, Y. Bisk, W. Han, R. Mottaghi, L. Zettlemoyer, and D. Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *CVPR*, 2020.

[13] A. Padmakumar, J. Thomason, A. Shrivastava, P. Lange, A. Narayan-Chen, S. Gella, R. Piramuthu, G. Tur, and D. Hakkani-Tur. Teach: Task-driven embodied agents that chat. In *AAAI*, 2022.

[14] X. Gao, Q. Gao, R. Gong, K. Lin, G. Thattai, and G. S. Sukhatme. Dialfred: Dialogue-enabled agents for embodied instruction following. *IEEE RA-L*, 2022.

[15] E. Kolve, R. Mottaghi, W. Han, E. VanderBilt, L. Weihs, A. Herrasti, D. Gordon, Y. Zhu, A. Gupta, and A. Farhadi. Ai2-thor: An interactive 3d environment for visual ai. *arXiv*, 2017.

[16] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. In *ICCV*, 2019.

[17] M. Deitke, W. Han, A. Herrasti, A. Kembhavi, E. Kolve, R. Mottaghi, J. Salvador, D. Schwenk, E. VanderBilt, M. Wallingford, et al. Robothor: An open simulation-to-real embodied ai platform. In *CVPR*, 2020.

[18] B. Kim, J. Kim, Y. Kim, C. Min, and J. Choi. Context-aware planning and environment-aware memory for instruction following embodied agents. In *ICCV*, 2023.

[19] W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In *ICML*, 2022.

[20] B. Draganski, C. Gaser, V. Busch, G. Schuierer, U. Bogdahn, and A. May. Changes in grey matter induced by training. In *Nature*, 2004.

[21] R. J. Zatorre, R. D. Fields, and H. Johansen-Berg. Plasticity in gray and white: neuroimaging changes in brain structure during learning. In *Nature neuroscience*, 2012.

[22] A. Gutchess. Plasticity of the aging brain: new directions in cognitive neuroscience. In *Science*, 2014.

[23] J. Yang, Z. Ren, M. Xu, X. Chen, D. J. Crandall, D. Parikh, and D. Batra. Embodied amodal recognition: Learning to move to perceive objects. In *ICCV*, 2019.

[24] K. Kotar and R. Mottaghi. Interactron: Embodied adaptive object detection. In *CVPR*, 2022.

[25] A. Inceoglu, E. E. Aksoy, A. C. Ak, and S. Sariel. Fino-net: A deep multimodal sensor fusion framework for manipulation failure detection. In *IROS*, 2021.

[26] Y. Zhang, J. Yang, J. Pan, S. Storks, N. Devraj, Z. Ma, K. Yu, Y. Bao, and J. Chai. DANLI: Deliberative agent for following natural language instructions. In *EMNLP*, 2022.

[27] S. Zhou, P. Yin, and G. Neubig. Hierarchical control of situated agents through natural language. In *NACCLW*, 2022.

[28] W. Yang, X. Wang, A. Farhadi, A. Gupta, and R. Mottaghi. Visual semantic navigation using scene priors. In *ICLR*, 2019.

[29] S. Y. Gadre, K. Ehsani, S. Song, and R. Mottaghi. Continuous scene representations for embodied ai. In *CVPR*, 2022.

[30] G. Sarch, Y. Wu, M. J. Tarr, and K. Fragkiadaki. Open-ended instructable embodied agents with memory-augmented large language models. *EMNLP Findings*, 2023.

[31] M. Salem, G. Lakatos, F. Amirabdollahian, and K. Dautenhahn. Would you trust a (faulty) robot? effects of error, task type and personality on human-robot cooperation and trust. In *HRI*, 2015.

[32] J. Thomason, M. Murray, M. Cakmak, and L. Zettlemoyer. Vision-and-dialog navigation. In *CoRL*, 2020.

[33] H. De Vries, K. Shuster, D. Batra, D. Parikh, J. Weston, and D. Kiela. Talk the walk: Navigating new york city through grounded dialogue. *arXiv*, 2018.

[34] K. Nguyen, D. Dey, C. Brockett, and B. Dolan. Vision-based navigation with language-based assistance via imitation learning with indirect intervention. In *CVPR*, 2019.

[35] A. Pashevich, C. Schmid, and C. Sun. Episodic transformer for vision-and-language navigation. In *ICCV*, 2021.

[36] K. Zheng, K. Zhou, J. Gu, Y. Fan, J. Wang, Z. Di, X. He, and X. E. Wang. Jarvis: A neuro-symbolic commonsense reasoning framework for conversational embodied agents. *arXiv*, 2022.

[37] S. Y. Min, H. Zhu, R. Salakhutdinov, and Y. Bisk. Don't copy the teacher: Data and model challenges in embodied dialogue. In *EMNLP*, 2022.

[38] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. In *PNAS*, 1996.

[39] V. Blukis, C. Paxton, D. Fox, A. Garg, and Y. Artzi. A persistent spatial semantic representation for high-level natural language instruction execution. In *CoRL*, 2022.

[40] Z. Liu, A. Bahety, and S. Song. Reflect: Summarizing robot experiences for failure explanation and correction. In *CoRL*, 2023.

[41] C. H. Song, J. Wu, C. Washington, B. M. Sadler, W.-L. Chao, and Y. Su. Llm-planner: Few-shot grounded planning for embodied agents with large language models. In *ICCV*, 2023.

[42] N. Shinn, B. Labash, and A. Gopinath. Reflexion: an autonomous agent with dynamic memory and self-reflection. *arXiv*, 2023.

[43] M. Skreta, N. Yoshikawa, S. Arellano-Rubach, Z. Ji, L. B. Kristensen, K. Darvish, A. Aspuru-Guzik, F. Shkurti, and A. Garg. Errors are useful prompts: Instruction guided task programming with verifier-assisted iterative prompting. *arXiv*, 2023.

[44] T. Silver, S. Dan, K. Srinivas, J. B. Tenenbaum, L. Kaelbling, and M. Katz. Generalized planning in pddl domains with pretrained large language models. In *AAAI*, 2024.

[45] W. Huang, F. Xia, T. Xiao, H. Chan, J. Liang, P. Florence, A. Zeng, J. Tompson, I. Mordatch, Y. Chebotar, et al. Inner monologue: Embodied reasoning through planning with language models. In *CoRL*, 2022.

[46] S. S. Raman, V. Cohen, E. Rosen, I. Idrees, D. Paulius, and S. Tellex. Planning with large language models via corrective re-prompting. In *Foundation Models for Decision Making Workshop @ NeurIPS*, 2022.

[47] I. Singh, V. Blukis, A. Mousavian, A. Goyal, D. Xu, J. Tremblay, D. Fox, J. Thomason, and A. Garg. Progprompt: Generating situated robot task plans using large language models. In *ICRA*, 2023.

[48] 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 *ICRA*, 2023.

[49] P. G. Zimbardo and F. L. Ruch. Psychology and life. In *Scott, Foresman*, 1975.

[50] F. Hutmacher. Why is there so much more research on vision than on any other sensory modality? In *Frontiers in psychology*, 2019.

[51] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, M. Anvari, M. Hwang, M. Sharma, A. Aydin, D. Bansal, S. Hunter, K.-Y. Kim, A. Lou, C. R. Matthews, I. Villa-Renteria, J. H. Tang, C. Tang, F. Xia, S. Savarese, H. Gweon, K. Liu, J. Wu, and L. Fei-Fei. BEHAVIOR-1k: A benchmark for embodied AI with 1,000 everyday activities and realistic simulation. In *CoRL*, 2022.

[52] S. Y. Min, D. S. Chaplot, P. Ravikumar, Y. Bisk, and R. Salakhutdinov. Film: Following instructions in language with modular methods. In *ICLR*, 2022.

[53] T. Ren, S. Liu, A. Zeng, J. Lin, K. Li, H. Cao, J. Chen, X. Huang, Y. Chen, F. Yan, Z. Zeng, H. Zhang, F. Li, J. Yang, H. Li, Q. Jiang, and L. Zhang. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv*, 2024.