REFLECT: Summarizing Robot Experiences for FaiLure Explanation and CorrecTion

7th Conference on Robot Learning (CoRL 2023), Atlanta, Turning on microwav

Tenglong Liu
National University of Defense Technology
2023.11.15

1 Introduction

- The failure explanations can either help a human user to debug the robotic system without having to read through the tedious execution logs, or guide the robot to correct the failure by itself.
- An effective failure reasoning framework requires several key components:
  - First, a component to summarize “what happened”
  - Second, a component to reason “what was wrong”
  - Finally, the ability to plan “what to do”
- Challenge: how to generate a textual summary of robot sensory data and systematically query LLMs for failure reasoning.

1 Introduction

Two important attributes of a good robot summary

- Multisensory.
  - The summary should cover all sensory modalities the robot has access to, such as visual, audio, contact, etc.

- Hierarchical.
  - The highest summary level
    - focus on identifying misalignment between the robot high-level plan and execution outcomes
  - The lower summary level
    - maintain enough environmental context for LLMs to generate an informative explanation that is useful for correction planning

2 Method: the REFLECT Framework

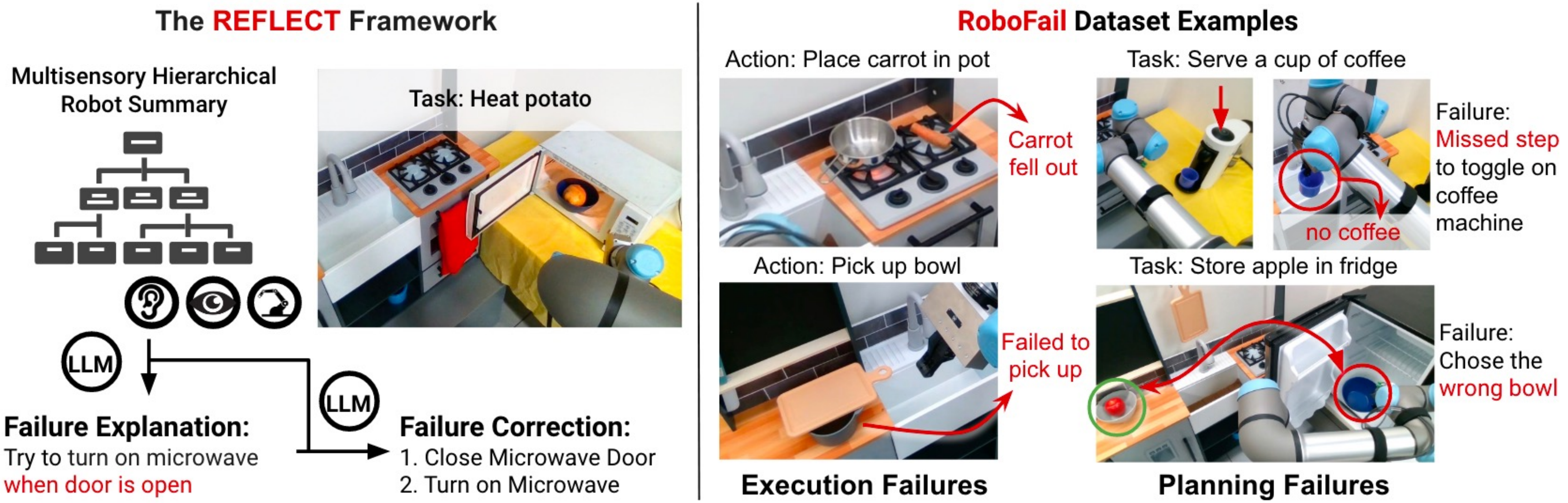


Fig 1: **A framework for robot failure explanation and correction.** On the left, we show the REFLECT framework that converts multisensory observations (RGB-D, audio, robot states) to a hierarchical summary of robot experiences. The summary is then used to query a Large Language Model (LLM) for failure explanation and correction. The right shows a few example failure cases in the RoboFail dataset.

2 Method: the REFLECT Framework

- hierarchical robot summary module → summarizes multisensory robot data with three levels of abstraction
- progressive failure explanation module → queries LLM to detect and explain the failure
- failure correction planner → generates an executable correction plan

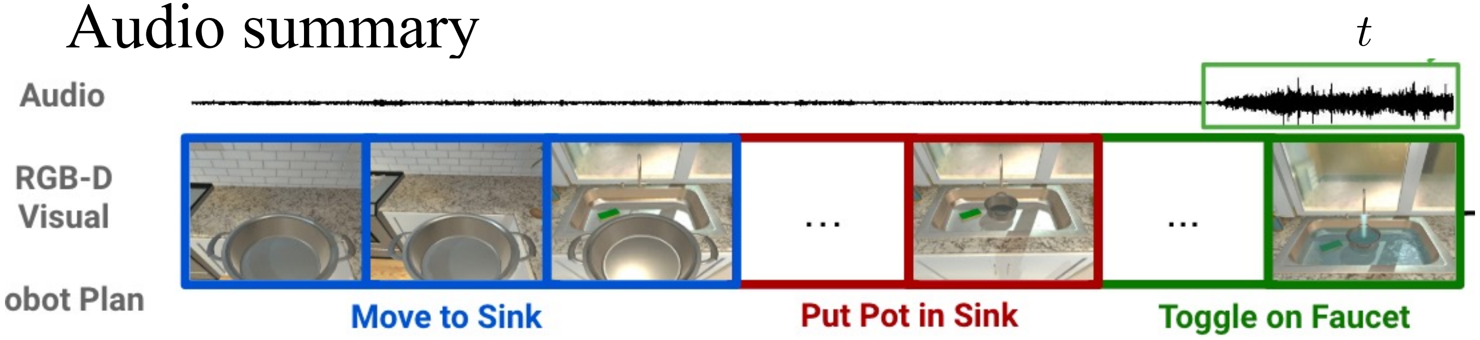
2.1 Hierarchical Robot Summary

1) aggregate and convert robot sensory data over time into a unified structure;
2) summarize the robot experiences for efficient failure localization and explanation.

2.1.1 Sensory-Input Summary

Visual summary with task-informed scene graphs

Audio summary



semantic segmentation I_t^S

cosine similarity of the CLIP embedding

X. Li, D. Guo, H. Liu, and F. Sun. Embodied semantic scene graph generation. In A. Faust, D. Hsu, and G. Neumann, editors, Proceedings of the 5th Conference on Robot Learning, volume 164 of Proceedings of Machine Learning Research, pages 1585–1594. PMLR, 08–11 Nov 2022.

2.1 Hierarchical Robot Summary

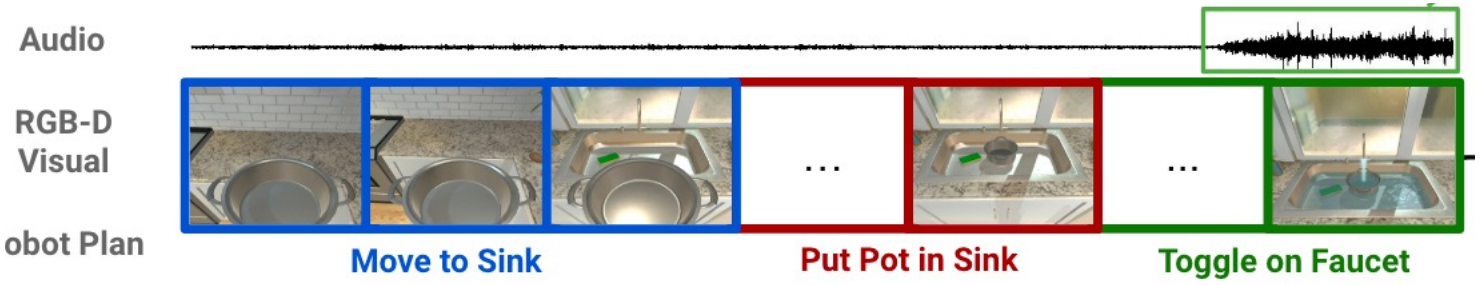
1) aggregate and convert robot sensory data over time into a unified structure;
2) summarize the robot experiences for efficient failure localization and explanation.

2.1.1 Sensory-Input Summary

Visual summary with task-informed scene graphs

Audio summary

s



AudioCLIP, Wav2CLIP

audio-lingual embedding

the cosine similarity between the audio embedding and the CLIP embeddings for a list of candidate audio event labels L

$$l^* = \text{argmax}_{l \in L}[C(s,l)], \ C = \frac{f_1(s) \cdot f_2(l)}{||f_1(s)|| f_2(l)||}$$

2.1 Hierarchical Robot Summary

2.1.2 Event-Based Summary

key frame selection mechanism

- The task-informed scene graph of the current frame is different from the previous frame
- The frame is the start or end of an audio event
- The frame marks the end of a subgoal execution

convert the scene graph into text

```
[timestep] Action: [robot action]
Visual observation: object1 [state], object2, object3 [state] ... # objects and states
object1 is [spatial relation] object2 ... # inter-object relations
object3 is inside robot gripper. # robot-object relations
Auditory observation: [audio summary].
```

2.1 Hierarchical Robot Summary

2.1.3 Subgoal-Based Summary

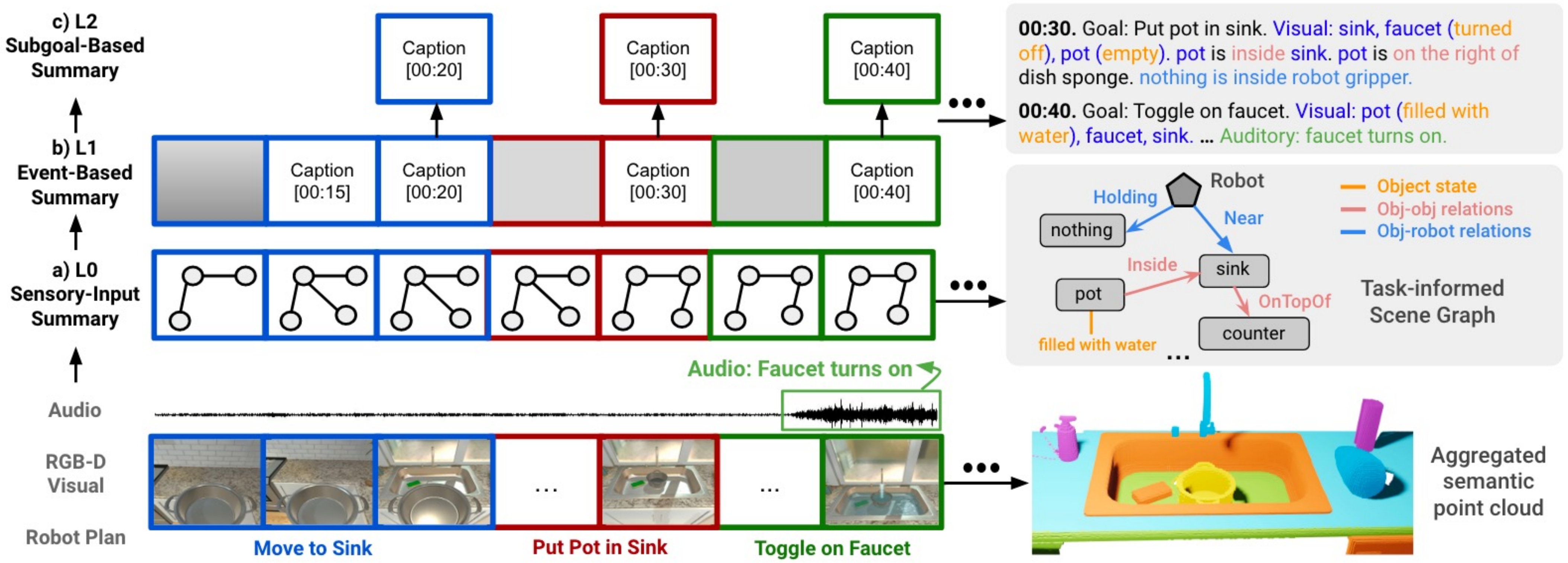


Fig 2: **Hierarchical robot summary** is composed of: a) a sensory-input summary that converts multisensory robot observations (RGB-D, sound, robot states) into task-informed scene graphs and audio summary; b) an event-based summary that generates captions for key event frames; c) a subgoal-based summary that contains the end frame of each subgoal.

2.2 Progressive Failure Explanation

Execution failure:

- Action-level observation details

Planning failure:

- Task-level information
- task description and robot plan

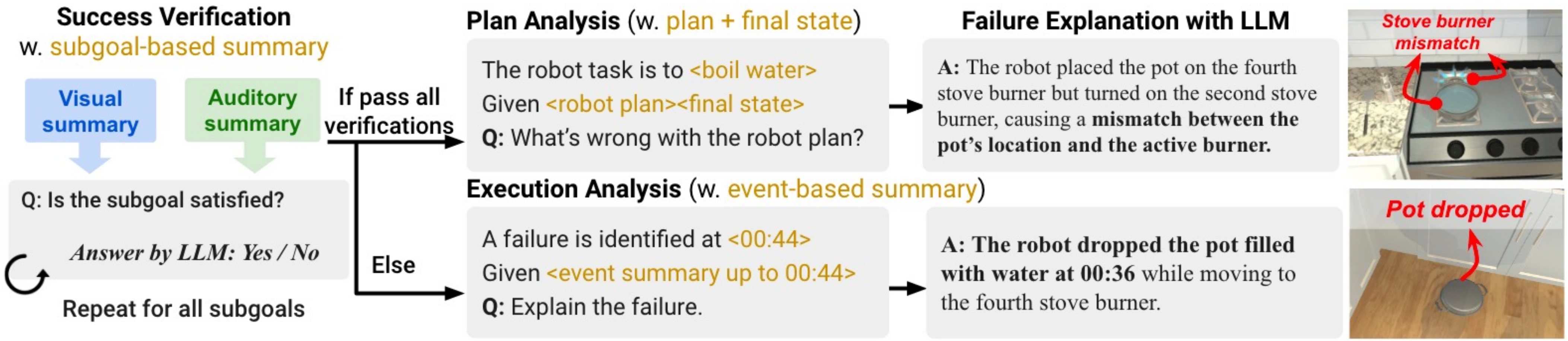


Fig 3: **Progressive failure explanation** verifies success for each subgoal. If a subgoal fails, the algorithm enters the *execution analysis* stage for detailed explanation. If all subgoals are satisfied, the algorithm enters *planning analysis* stage to check errors in the robot plan.

2.2 Progressive Failure Explanation

- **first iterates through the subgoals and verifies success**

> The robot subgoal is [robot subgoal at time t]. Given [subgoal-based summary at time t]
> Q: Is the subgoal satisfied? A: Yes

- **event-based summary for failure explanation**

> The robot task is to [task name]. A failure is identified at t. Given [event-based summary up to t]
> Q: Briefly explain what happened at t and what caused the failure?
> A: At 00:44, the robot attempted to put the pot on the fourth stove burner, but the pot was not in its gripper. The failure was caused by the robot dropping the pot filled with water at 00:36 while moving to the fourth stove burner.

- **all subgoals are achieved but the task still failed**

> The robot task is to [task name]. The task is successful if [goal state].
> The robot plan is [original robot plan]. Given [final state]
> Q: What's wrong with the robot plan that caused the robot to fail?
> A: The robot placed the pot on the fourth stove burner but turned on the second stove burner, causing a mismatch between the pot's location and the active burner.
> Q: Which time step is most relevant to the above failure?
> A: 00:49

2.3 Failure Correction Planner

the failure explanation can also guide a language planner to generate a high-level correction plan that leads to task success

The robot task is to [task name]. The task is successful if [goal state].
The robot plan is [original robot plan]. Given [final state]
Q: What's wrong with the robot plan that caused the robot to fail?
A: The robot placed the pot on the fourth stove burner but turned on the second stove burner, causing a mismatch between the pot's location and the active burner.
Q: Which time step is most relevant to the above failure?
A: 00:49

The robot task is to [task name]. The original robot plan is [original robot plan].
Given [failure explanation] [final state] and [goal state]
Correction plan: toggle_off (stoveburner-2), toggle_on (stoveburner-4)

W. Huang, P. Abbeel, D. Pathak, and I. Mordatch. Language models as zero-shot planners: Extracting actionable knowledge for embodied agents. In International Conference on Machine Learning, pages 9118–9147. PMLR, 2022.

3 The RoboFail Dataset

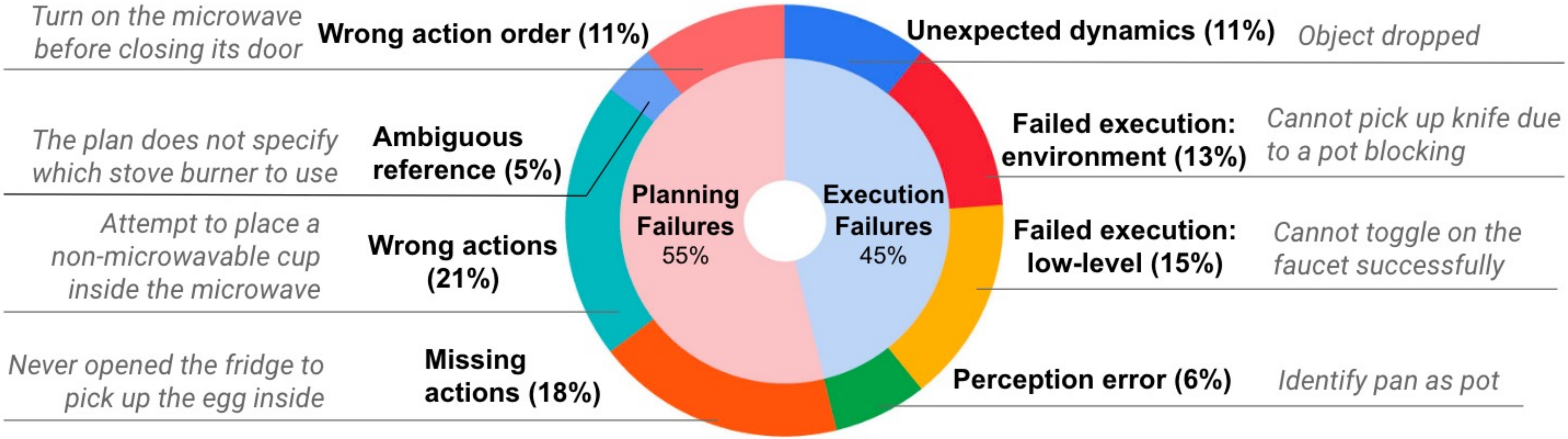


Fig 4: **RoboFail Failure Taxonomy**

4 Evaluation

- Exp (explanation): percentage of predicted failure explanations that are correct and informative as determined by human evaluators
- Loc (localization): percentage of predicted failure time that align with actual failure time.
- Co-plan (correction planning success rate): percentage of tasks that succeed after executing the correction plan.

	Execution failure			Planning failure		
Method	**Exp**	**Loc**	**Co-plan**	**Exp**	**Loc**	**Co-plan**
w/o progressive	46.5	62.8	60.5	61.4	70.2	64.9
Subgoal only	76.7	74.4	51.2	71.9	73.7	75.4
LLM summary	55.8	67.4	65.1	57.9	54.4	66.7
w/o explanation	-	-	41.9	-	-	56.1
REFLECT	**88.4**	**96.0**	**79.1**	**84.2**	**80.7**	**80.7**

Table 1: Result in Simulation Environments

	Execution failure		Planning failure	
Method	**Exp**	**Loc**	**Exp**	**Loc**
BLIP2 caption	6.25	25.0	35.7	57.1
w/o sound	50.0	68.8	78.6	78.6
w/o progressive	43.8	81.3	71.4	78.6
Subgoal only	56.3	62.5	71.4	78.6
LLM summary	37.5	75.0	64.3	71.4
REFLECT	**68.8**	**93.8**	**78.6**	**78.6**

Table 2: Result in Real-world Environments

4 Evaluation

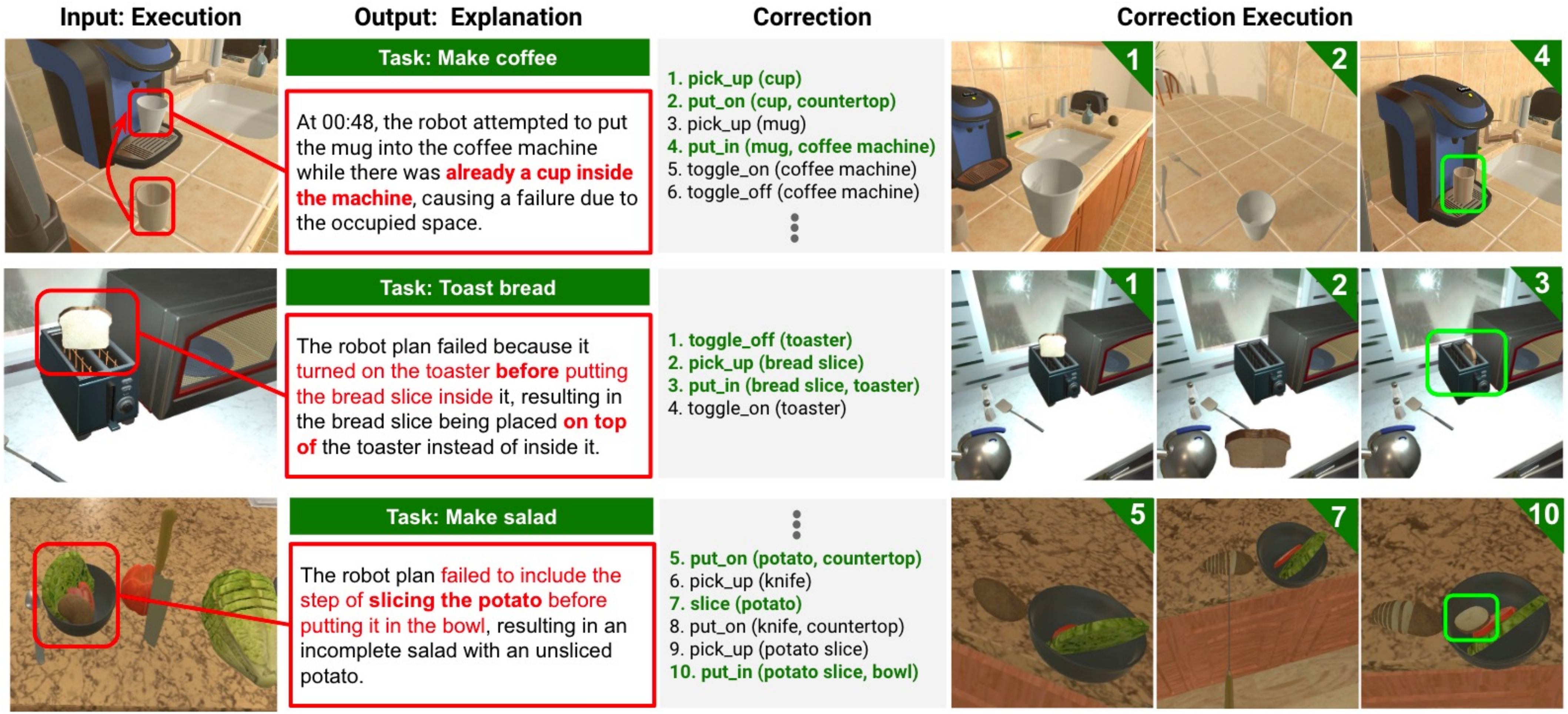


Fig 5: **Qualitative results in simulation.** Given a failed robot task execution, REFLECT is able to generate informative failure explanations for both execution and planning failures. Conditioned on the explanation, a language planner can generate a high-level plan for the robot to correct the failure and complete the task.

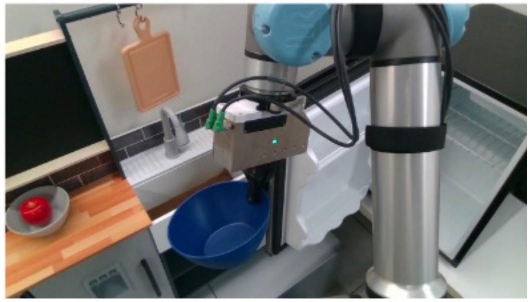
4 Evaluation

Method	Execution failure Exp	Execution failure Loc	Planning failure Exp	Planning failure Loc
BLIP2 caption	6.25	25.0	35.7	57.1
w/o sound	50.0	68.8	78.6	78.6
w/o progressive	43.8	81.3	71.4	78.6
Subgoal only	56.3	62.5	71.4	78.6
LLM summary	37.5	75.0	64.3	71.4
REFLECT	**68.8**	**93.8**	**78.6**	**78.6**

Table 2: Result in Real-world Environments

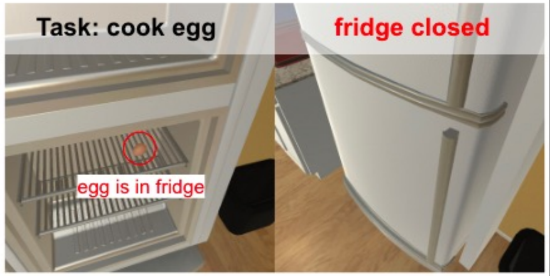
The captions generated by BLIP2 lack necessary information for failure explanation.

The progressive algorithm helps with more accurate localization and informative explanation.



BLIP2 caption: a robot is holding a bowl in a kitchen.

REFLECT: fridge (with door open), apple, white bowl, dark blue bowl. apple is inside white bowl. dark blue bowl is inside robot gripper.



w/o progressive: The robot failed to cook an egg because it did not have the egg in its gripper when attempting to crack it and put it in the pan. The visual observations show that the robot's gripper was empty during these actions.

REFLECT: At 00:14, the robot attempted to pick up an egg while the fridge was still closed, causing the failure. The robot should have opened the fridge before trying to pick up the egg.

LLM summary:
Summary:
00:14 - The robot tried to pick up the egg, but the gripper remained empty.
...
01:52 - The robot attempted to crack the egg, but the gripper was still empty.
01:53 - The robot tried to put the cracked egg in the pan, but the gripper remained empty.

Explanation:
The robot failed to cook an egg because it was unable to pick up the egg with its gripper, resulting in an empty gripper throughout the task and preventing the egg from being cracked and placed in the pan.

Fig 6: [w/o progressive] vs. [LLM summary] vs. Ours

Explanation: At 00:48, the robot attempted to put the mug into the coffee machine while there was **already a cup inside the machine**, causing a failure due to the occupied space.

REFLECT:
1. pick_up (cup)
2. put_on (cup, countertop)
3. pick_up (mug)
4. put_in (mug, coffee machine)
5. toggle_on (coffee machine)
6. toggle_off (coffee machine)
7. pick_up (mug)
8. put_on (mug, countertop)

w/o explanation:
1. pick_up (mug)
2. put_on (mug, sink basin)
3. toggle_on (faucet)
4. toggle_off (faucet)
5. pick_up (mug)
6. pour_liquid_from (mug, sink)
7. put_in (mug, coffee machine)
8. toggle_on (coffee machine)
9. toggle_off (coffee machine)
10. pick_up (mug)
11. put_on (mug, countertop)

Fig 8: Failure explanation helps correction planning.

4 Evaluation

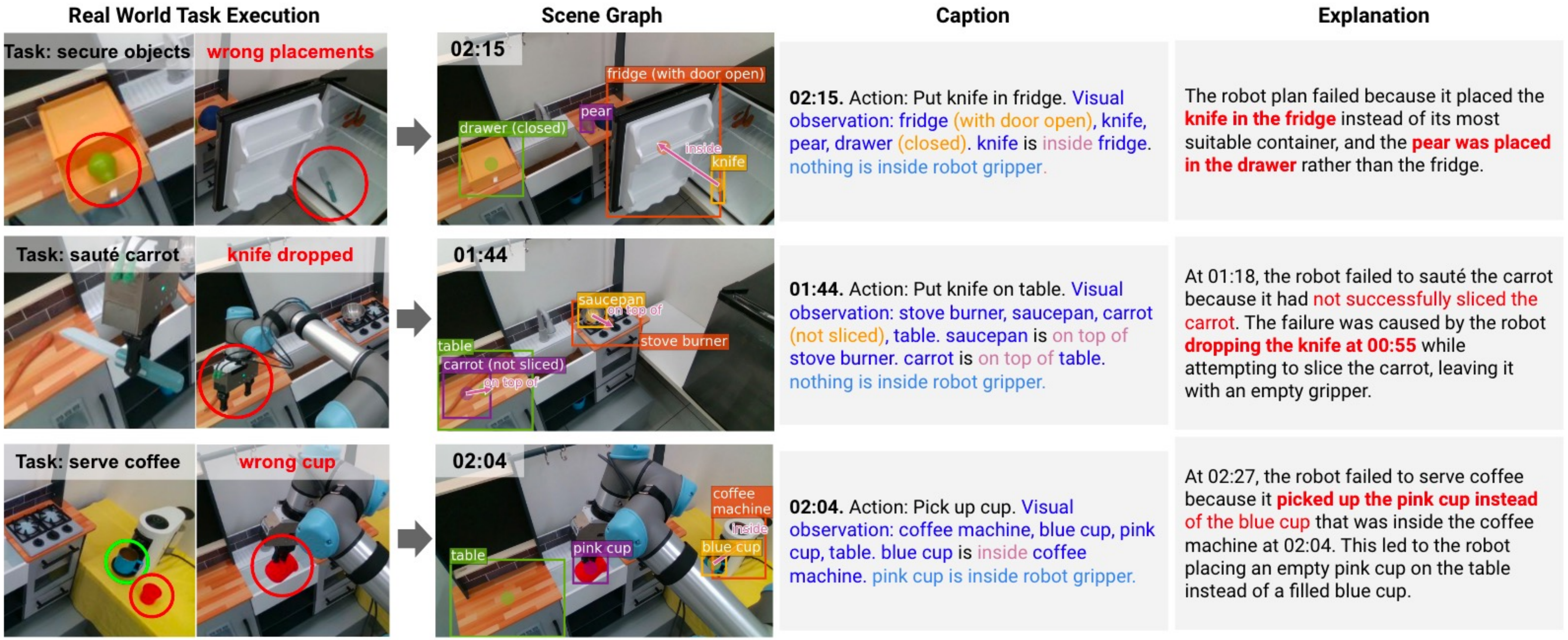


Fig 7: **Qualitative results in real world.** REFLECT is able to summarize and generate informative failure explanations for real-world robot executions. The above figure shows three failed task executions on the left, the corresponding scene graph and caption for one key frame in the middle, and the LLM-generated failure explanation on the right.

5 Limitations

- Even though the heuristics used to generate scene graphs is sufficient for scenarios studied in the paper, it may fall short in more complex environments.

- The object state detection method assumes a given list of candidate object states.

- The framework also assumes the rest of the environment will remain static throughout the robot task execution.

- It is less effective for handling low-level control failures.

- Either training a large spatial reasoning model or fine-tuning an existing model on robotics data could be a promising solution
- Future work may consider developing better perception methods that capture more low-level state information.

Thanks

Tenglong Liu
National University of Defense Technology
2023.11.15

[2]text color: blue: visual , green: audio , light blue: contact, yellow: summary, orange: final state, failure explanation, brown: timestep, task name, robot subgoal, original robot plan, goal state, blue highlight: LLM output

[3]Examples of full prompts are shown in the appendix.