Supplementary Material – MaP-AVR: Meta-action Planner for Agent Powered by VLM and RAG

In this document, we provide some extra experimental results to demonstrate the effect of the error correction bias and the trade-off between model complexity and performance.

1. The Execution of the Meta-Action

2. The Interface to Prepare the RAG Database and Evaluate the Feasibility of the Planner

2.1. Prepare the initial samples in the task database

In order to use Retrieval-augmented generation(RAG) technology, we need to prepare a task database. The database will contain some pre-planned scenarios and tasks as initial examples. To ensure the accuracy of these initial examples, we designed an interface to involve human intervention for verification and correction, as shown in Fig. 1. Through this interface, humans can efficiently participate in correcting the entire process of interactions with the Visual Language Model (VLM) for any images and any tasks.

2.2. Evaluate the meta-action planner with human annotators

In addition to preparing the initial examples for the task database, human involvement is also needed to evaluate the effectiveness of the task planner's planning result. We developed another interface to assist human annotators in efficiently and quickly evaluating the task planning results. This interface also supports comparing the planning results with and without In-Context Learning. This interface is demonstrated in Fig. 2. Annotators can freely select data sources, upload images, and specify any daily tasks that humans believe a robot might be able to complete in the corresponding scenario depicted in the image. Once the submit button is clicked, the back-end will concurrently run a task planner with In-Context Learning (ICL) and another without ICL. After execution, the results of both will be displayed in the corresponding sections on the front-end page. Next, the annotators need to assume they are playing the role of a robot, evaluating whether they can successfully complete the task by following the planned instructions. They should then click the appropriate button to record the evaluation results. We further have an area dedicated to double-checking, which allows annotators to review and correct the evaluation results at any time. Finally, once the annotators have confirmed the evaluation results, they simply need to click the save button, and the entire evaluation record will be stored in a centralized location.

3. How the RAG and ICL can Help for Daily Task Planning

3.1. Some typical failure patterns without ICL

3.2. The self-augmented feature of the task database is important for general applicability

For the RAG database, it can also grow autonomously as more tasks are completed. We calculate two metrics for every task in the database with method based VLM: the similarity of relevant objects and the similarity of sequences from the task planner. When a new task is executed completely and correctly, these two metrics are calculated across all tasks in the database. If the metrics are not sufficiently high, suggesting that the new task has low cross-correlation with the existing database, the new task will be added to the task database. As shown in Fig. 4, a new task is denoted as yellow star mark. The new task will be evaluated to add to task database.

4. The Whole CoT Prompts of the Meta-Action

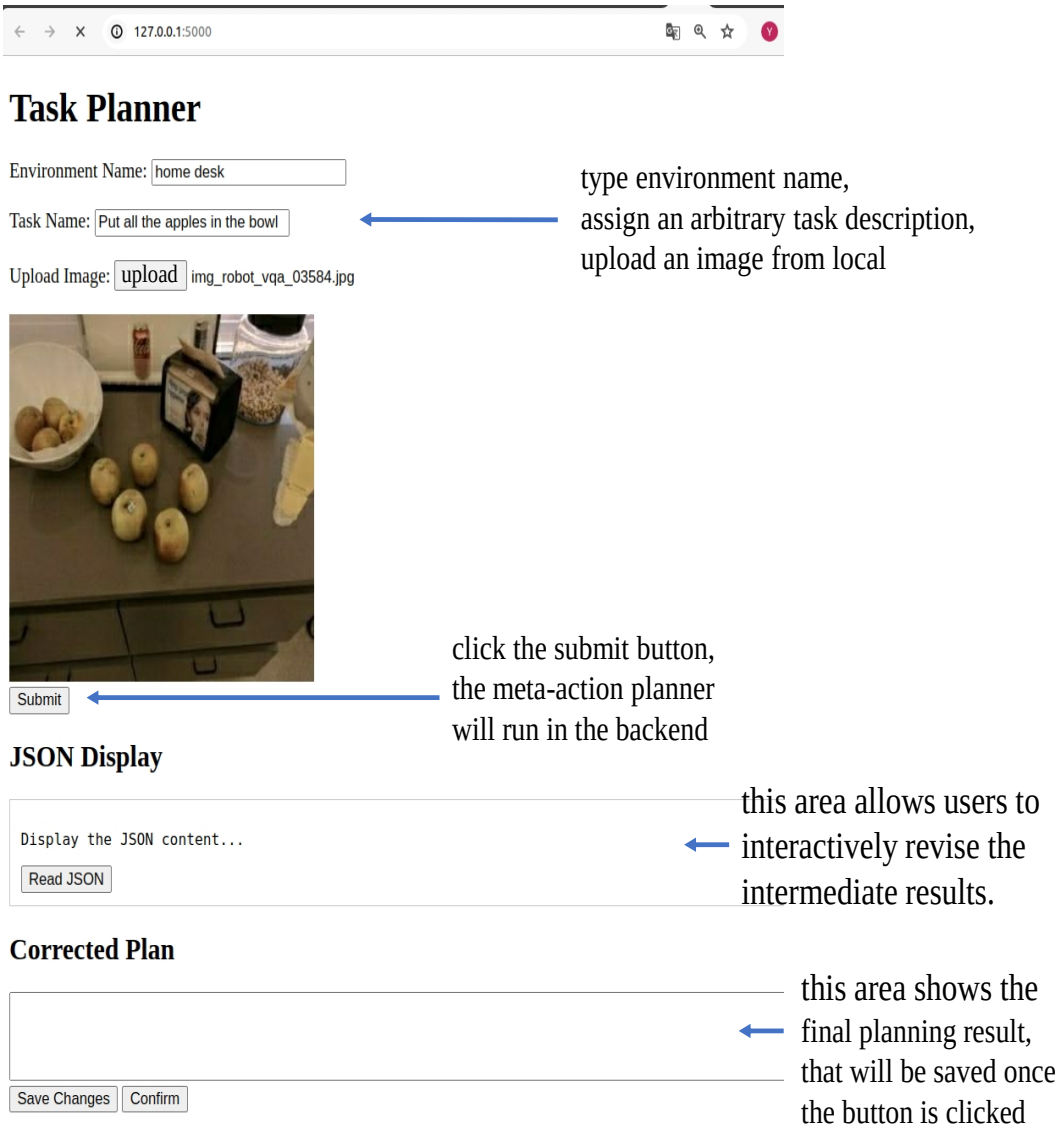

Figure 1: This image shows the front-end interface used for introducing human intervention to prepare correctly planned examples for the database.

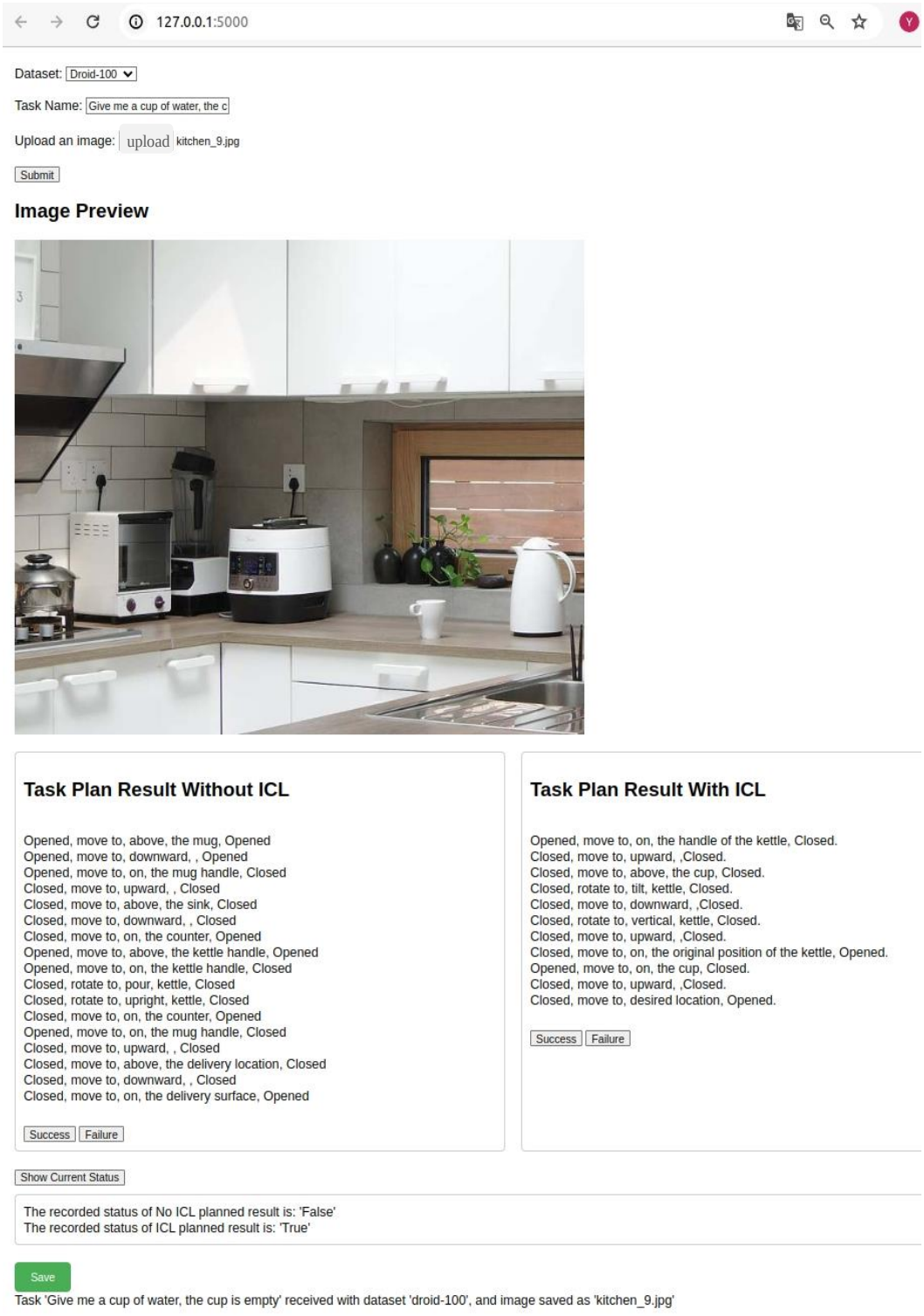

Figure 2: This image shows the interface used to involve human annotators in evaluating the task planner.

(a) (b) (c)

Figure 3: Comparison of the evolution of the loss between networks with the error correction and DT combined (red) vs networks with DT only (blue). From right to left, (a) SimpleNet, (b) SparsityCNN and (c) ResNet18

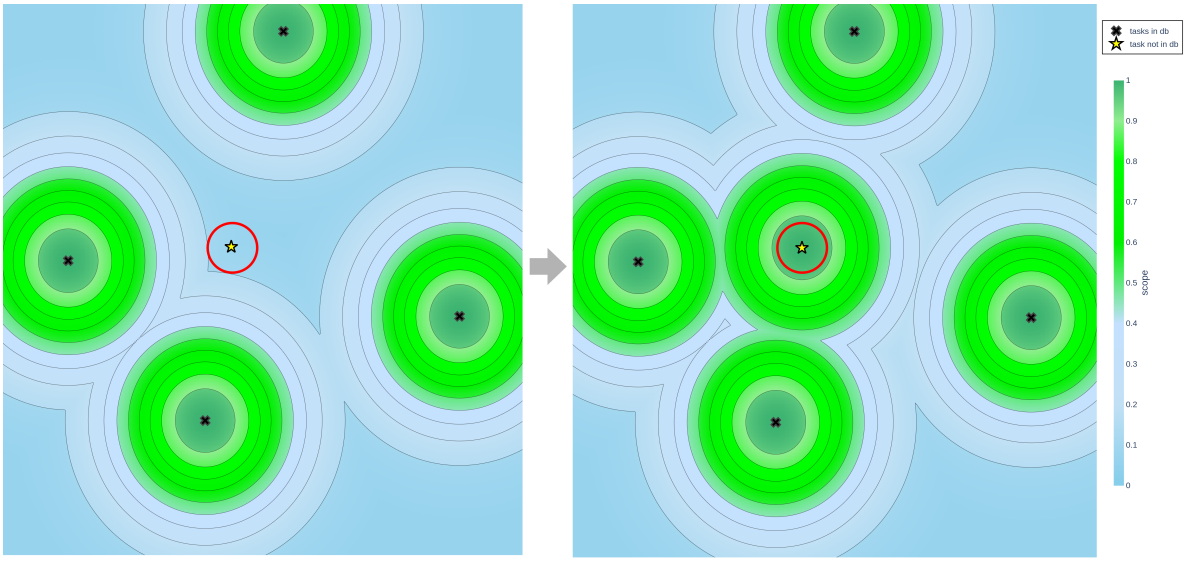

Figure 4: The change of the scope of task database