A. Implementation Details

We use the pre-trained weights from the original TEACh codebase for the ACTIONER and select the confusion thresholds via the TEACh validation set. For clarity and for keeping hyper-parameters minimal, we use the same threshold across both action and object distributions when using entropy-based confusion. Our ablation studies show that using a common hyper-parameter does not substantially affect performance. The confusion threshold is set to 0.9 for the entropy-based method and 1.2 for the gradient-based method. Moreover, we train the QA EVALUATOR on the question-answer pairs extracted from TEACh and the oracle question-answer pairs generated using the QA GENERATOR. For PLANNER, we finetune the pre-trained T5 model [30] using Adam optimizer with the learning rate of $3e-5$ and batch size of 6. We construct the training data for PLANNER by converting the training trajectories of TEACH into sequences of subgoals. We treat all interaction actions as subgoals. For navigation actions, we create subgoals by replacing sequences of navigation actions with an abstract "Find" action with the destination as the next `object` manipulated. We evaluate the performance of PLANNER via Rouge-L [20], which measures the longest common subsequence (LCS) between the ground truth sub-goal sequence and the generated sub-goal sequence. For the QA EVALUATOR, we use a global batch size of 32, AdamW optimizer [22] with the weight decay of 0.33 and learning rate of $1e-5$. Our code is based on PyTorch [28] and Huggingface Transformers [46]. We train our models on a machine equipped with two RTX 8000 with 40GBs of memory.

B. Method

B.1. Pseudocode for Entropy-based Confusion

We provide the pseudocode for our entropy-based confusion module in Algorithm 1. For clarity, we simplify the question-answer generation and selection by referring to the combination of the QA GENERATOR and QA EVALUATOR steps as QUESTIONER.

Algorithm 1 Entropy-based Confusion

Input: Entropy function $H(\cdot)$; Action distribution threshold ϵ_α ; Object distribution threshold ϵ_o; Interaction action set A^I; State information $s_{t-1} = (x_{1:t-1}, v_{1:t-1}, \alpha_{1:t-1})$; Selected question and answer pair (q_t^*, a_t^*) at time step t.

$p_t^\alpha, p_t^o \leftarrow$ ACTIONER(s_{t-1}) # Select next action
$\hat{\alpha}_t = \arg\max_\alpha p_t^\alpha$
$\hat{o}_t = \arg\max_o p_t^o$
if $\left(H(p_t^\alpha) > \epsilon_\alpha\right)$ or $\left(\hat{\alpha}_t \in A^I \text{ and } H(p_t^o) > \epsilon_o\right)$ **then**
    # Generate question-answer pair
    $(q_t^*, a_t^*) \leftarrow$ QUESTIONER(s_{t-1})
    # Augment state information
    $\tilde{s}_{t-1} \leftarrow (s_{t-1}, q_t^*, a_t^*)$
    # Select next action given question-answer pair
    $(\tilde{p}_t^\alpha, \tilde{p}_t^o) \leftarrow$ ACTIONER$(\tilde{s}_{t-1})$
    # Compute action and object entropy difference
    $\Delta_\alpha \leftarrow H(\tilde{p}_t^\alpha) - H(p_t^\alpha)$
    $\Delta_o \leftarrow H(\tilde{p}_t^o) - H(p_t^o)$
    if $\left(\Delta_\alpha < 0\right)$ or $\left(\hat{\alpha}_t \in A^I \text{ and } \Delta_o < 0\right)$ **then**
        # If entropy decreases, ask the question
        $\pi_\theta(\tilde{s}_{t-1}) = (\tilde{p}_t^\alpha, \tilde{p}_t^o)$
        $\hat{\alpha}_t = \arg\max_\alpha \tilde{p}_t^\alpha$
        $\hat{o}_t = \arg\max_o \tilde{p}_t^o$
    end if
end if

B.2. QA EVALUATOR

B.3. Sub-goal Generator

We further evaluate the sub-goal generator on the seen and unseen test sets employing ROUGE-L and BERTScore as our evaluation metrics. For the immediate next subgoal, ROUGE-L is 66.1 (seen) and 64.3 (unseen). When considering the entire sequence of all forthcoming subgoals, the scores were 46.2 (seen) and 44.1 (unseen). ROUGE-L measures the maximum exact matching subsequence between generated and reference sentences and is considerably high given that our generator produces free-form text. Additionally, utilizing BERTScore, which assesses cosine similarity between contextual embeddings, we observe high scores of 95.2/91.5 (seen) and 95.0/91.0 (unseen) for the next subgoal and all subgoals, respectively. This indicates a robust performance in capturing semantic similarity. Manual inspection further corroborated the quality of the generated subgoals, affirming their coherence and logical soundness.

B.4. Generated QA Pairs

To assess the quality of the generated question-answer (QA) pairs, we measure perplexity on the TEACh test split. The generated QA pairs exhibit a lower perplexity of 137.62, in contrast to the higher perplexity of 316.59 observed in human-generated QA pairs. This decrease in perplexity indicates an enhanced generalization performance in the generated QA pairs. The higher perplexity in human QA pairs is likely a result of the presence of typos and abbreviations commonly encountered in online text conversations.

Furthermore, we conduct experiments to understand the effect of mismatched QA pairs on the model's efficacy. These experiments involve altering the questions in two specific ways: for the "Empty Question" variant, the question is replaced with an empty string, and for the "<UNK> Question" variant, it is substituted with '<UNK>'. The results, detailed in Table 4, reveal a noticeable decline in performance when the question is substituted with an empty string

Table 4. **Impact of Mismatched QA Pairs**

Model	SR [TLW]	GC [TLW]
ELBA w/E - Oracle QA	16.0 [1.5]	19.4 [4.4]
- Empty Question	14.4 [2.4]	17.6 [4.6]
- <UNK> Question	13.4 [1.5]	16.8 [4.0]

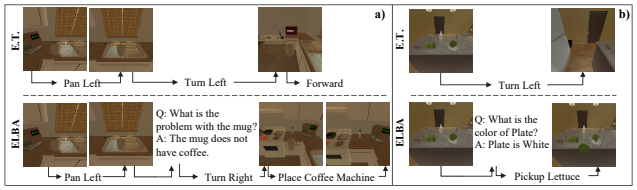

Figure 7. **Qualitative Examples.** The predicted trajectory of E.T. and ELBA. In each example, the top row shows the predicted trajectory by the E.T. model, and the bottom row shows the predicted trajectory of ELBA. Examples (a) `Make coffee` and (b) `Make breakfast` show successful cases of ELBA.

or '<UNK>', underscoring the critical role and importance of valid QA pairs.

C. Assessing QA Relevance

Due to the lack of ground truth in both subgoal actions and QAs, assessing the appropriateness of timing and relevance of questions generated by the agent along the trajectories can be challenging. The ideal evaluation would involve a human expert evaluating each question and answer generated across the agent's trajectory, leading to infeasible labor demands. Therefore, we instead resort to qualitative analysis, with a few examples shown in Figure 7, and a small-scale user study to evaluate the relevance and correctness of the generated questions for 6 different subgoal tasks.

Figure 7 showcases ELBA 's ability to generate QA pairs related to objects critical to the task at hand, thereby guiding the embodied agent to perform actions that are relevant to successfully completing the task. For instance, by querying information about the mug or the color of a plate, the model demonstrates an understanding of the task context required to determine subsequent actions, such as placing the mug in a coffee machine or transferring lettuce to the plate. In contrast, the baseline struggles to discern the most relevant actions and resorts to an exploration of the room.

The user study investigates the relevance of the question-and-answer (QA) dialogue in relation to task completion. Participants were presented with six example sub-trajectories depicting ELBA's process of completing various household tasks, with the specific task name and goal condition for each sub-trajectory, and a series of evaluative questions regarding the QA dialogues' relevance to task steps, overall task relevance, grammatical correctness, and any identified issues. The instructions provided to the participants, as shown in Figure 8, outline the intent of the user study. Additionally, Figures 9 and 10 present two example trajectories featured in the user study and the corresponding task and goal condition presented to the user.

Table 5. Summarized user study scores - ELBA's QA evaluation.

Questions	Percentage
Relevance to Task Steps (↑)	61.19% ± 15.56 %
Overall Task Relevance (↑)	80.89% ± 18.80%
Grammatical Correctness (↑)	100% ± 0%
Issues Identified (↓)	62.41% ± 18.08%

Instructions:
We present several example trajectories illustrating an agent's process of completing various household tasks. Each example showcases a sequence of images that capture the agent's first-person view of achieving the designated subgoals of the task. On top of each image is text indicating the agent's next action. In some of the steps, there is an additional question-and-answer dialogue representing the agent's inquiries at these given time steps during task execution. For each example trajectory, please answer the following questions:

1. For each question, is it relevant to the specific time step? If not, please identify the time steps for which a question is irrelevant.
2. Overall, are the questions posed by the agent relevant to the task?
3. Are the questions grammatically correct? Please answer 'Yes' or 'No'.
4. Do you identify any issues with the questions or answers? Please specify.

Figure 8. A snapshot of the user study instructions outlining the objectives and questions.

Table 5 presents the summarized results of the user study. We compute the percentage of QAs recognized as relevant to the overall task for each instance and average across all examples and participants. This method was similarly applied to the issues flagged by participants. Participants generally found the QA dialogues relevant to the overarching tasks, with a promising average relevance score of 80.89%. However, participants indicated a moderate average score of 61.19% regarding QA relevance to specific task steps, indicating that the question asked might not be directly timely to the next actions to be taken. Despite occasional discrepancies in immediate relevance, the overall task relevance scores show that the ELBA's QA capabilities effectively contribute to task understanding and execution. All participants confirmed the grammatical correctness of the QA dialogues, underscoring ELBA's ability to generate clear and accurate dialogues. Most of the issues identified are about the repetition of QAs or the relevance of QAs towards specific timesteps. Some users indicated that the question could

be relevant to nearby or earlier time steps, suggesting a potential avenue in improving the temporal relevance of QA dialogues during task execution. These findings highlight both strengths and areas of improvement for future research in task-driven interactive QA for embodied agents.

D. Additional Quantitative Results

The primary objective of our work is to demonstrate the benefits of enabling an embodied AI agent to ask questions when encountering uncertainty or confusion during task execution. This capability is expected to enhance the performance of an agent by facilitating more effective feedback and decision-making. To validate the general applicability of our approach to different ACTIONER agents, we extend our methodology to HELPER [34]. For this purpose, we integrate a Question-Answering (QA) module within HELPER, that is designed to prompt the agent to ask targeted questions about errors it encounters during task execution, thus providing an opportunity for real-time correction and learning. In Table 6, we observe a notable improvement in the performance of HELPER with QA capabilities, suggesting that being able to ask relevant questions can potentially enhance the effectiveness of various ACTIONER models, which are orthogonal contributions to this field.

E. Additional Qualitative Analysis

We also analyze the failure cases of ELBA and categorize possible errors into the following limitations:

Color Detection: The generated oracle QAs sometimes contain errors regarding the appearance of objects. Our model might detect a wrong color, especially when there is a shadow on objects. For example, our model could detect the color of the table as "black" while it is supposed to be a "white" table under the shadow. Currently, we use a simple dictionary-based approach that first defines a color dictionary that contains the HSV range for each color and then determines the color of an object by looping through the color dictionary and using the color that can cover the largest area as the object color. Thus, there is room for improvement in color detection, *e.g.*, by employing vision models.

Ill-Formed Model-Generated QAs: In some cases, the model-generated question-answer pairs might not be well-formed, *e.g.*, when the generated question does not match the candidate answer (*e.g.*, "`Q: How is the bowl on the self arranged? A: Place potato in bowl.`"). This issue could potentially be solved by including an evaluator model that measures the relevance between the question and the answer.

Ill-timed QAs: We find that the generated question and answer pair at a certain time-step could be ill-timed. For example, when the agent is performing a certain sub-goal (*e.g.*, `Find Potato`) given a high-level task (*e.g.*, `Make potato salad`), our model will sometimes generate an ill-timed question on a task-irrelevant sub-goal (*e.g.*, `Pickup Dish Sponge`) or a sub-goal that follows one or more time steps after the completion of the current sub-goal (*e.g.*, `Find Plate`). These errors are caused by the fact that we use all future sub-goals predicted by the PLANNER as candidate answers rather than constructing candidate answers from the next sub-goal instruction only. The latter approach requires the model to track the completion status of the current sub-goal so that the model can decide when to ask questions about the next sub-goal. While our current model bypasses the challenge of tracking sub-goal status by treating all future sub-goals as candidate answers, this leads to ill-timed questions during inference and potentially increases the number of steps needed to complete the task.

Table 6. Effect of enabling QA in HELPER.

Model	SR [TLW]	GC [TLW]
HELPER (reported)	9.48 [1.21]	10.05 [3.68]
HELPER + QA	11.05 [1.78]	13.52 [4.99]

F. Broader Impact

Our work highlights the need for a more natural way of interaction for agents to operate in human spaces. Future extensions of this work include developing more robust QA Evaluators and multimodal QA Generators. While ELBA is a step forward towards truly interactive agents, there remain several open challenges, including but not limited to better contextual understanding and temporal reasoning, handling unexpected or ambiguous feedback, incorporating memory mechanisms to remember and adapt QAs to dynamic changes in the environment during task execution, and automated methods for evaluating timeliness and relevance of task-driven interactive embodied question answering. In future research, we also hope to explore unified generative approaches.

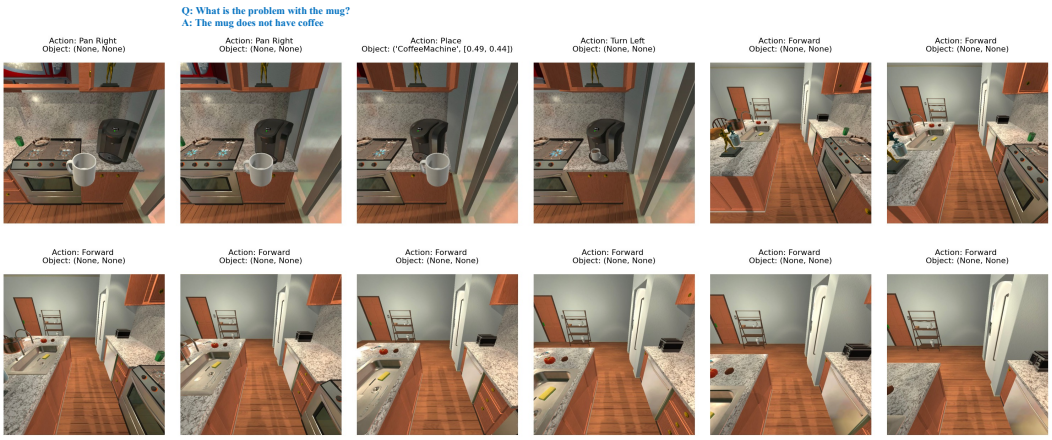

Figure 9. **User Study Example 1.** Task: `Coffee`. Goal Condition: `Place the mug on the coffee machine.`

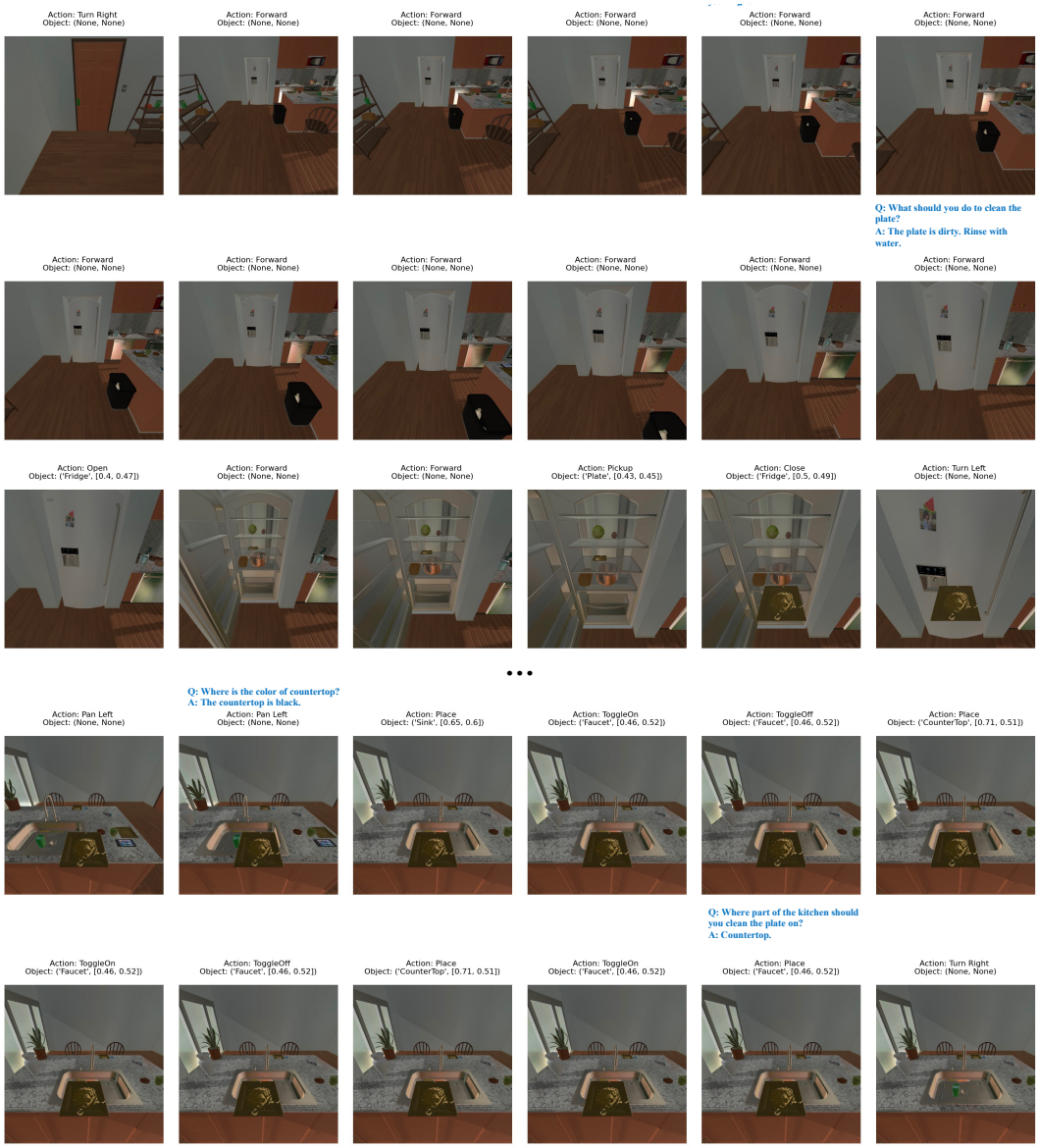

Figure 10. **User Study Example 2.** Task: `Clean All X`. Goal Condition: `Clean the plate.`