# AGILE: A Novel Reinforcement Learning Framework of LLM Agents


**Peiyuan Feng**[*1]    **Yichen He**[*1]    **Guanhua Huang**[*†2]    **Yuan Lin**[*1]
**Hanchong Zhang**[*†3]    **Yuchen Zhang**[*1]    **Hang Li**[1]

[1]ByteDance Research    [2]University of Science and Technology of China
[3]Shanghai Jiao Tong University

{fpy,hyc,linyuan.0,zhangyuchen.zyc,lihang.lh}@bytedance.com,
guanhuahuang@mail.ustc.edu.cn, zhanghanchong@sjtu.edu.cn


## Abstract


We introduce a novel reinforcement learning framework of LLM agents named AGILE (**AG**ent that **I**nteracts and **L**earns from **E**nvironments) designed to perform complex conversational tasks with users, leveraging LLMs, memory, tools, and interactions with experts. The agent possesses capabilities beyond conversation, including reflection, tool usage, and expert consultation. We formulate the construction of such an LLM agent as a reinforcement learning (RL) problem, in which the LLM serves as the policy model. We fine-tune the LLM using labeled data of actions and the PPO algorithm. We focus on question answering and release a dataset for agents called ProductQA, comprising challenging questions in online shopping. Our extensive experiments on ProductQA, MedMCQA and HotPotQA show that AGILE agents based on 7B and 13B LLMs trained with PPO can outperform GPT-4 agents. Our ablation study highlights the indispensability of memory, tools, consultation, reflection, and reinforcement learning in achieving the agent's strong performance. Datasets and code are available at https://github.com/bytarnish/AGILE.


## 1   Introduction

Large Language Models (LLMs) have exhibited remarkable capabilities such as instruction following, reasoning, and zero-shot learning [3, 45, 20, 26], which have greatly catalyzed the development of autonomous agents based on LLMs [28, 30, 2], also known as LLM agents. Recent works propose several essential components or workflows to enhance the abilities of LLM agents, such as planning [45, 51, 39], reflection [21, 40], tool-use [29, 36, 48] and life-long learning [42]. However, it remains unclear how to integrate all components into a unified framework and optimize them end-to-end.

In this paper, we introduce a novel reinforcement learning framework for LLM agents to unify various components and streamline their learning and operation processes. As shown in Figure 1(a), the architecture of the agent system, named AGILE, comprises four modules: LLM, memory, tools, and executor. Furthermore, the agent can interact with both users and experts. The LLM, functioning as the predictor of all actions, generates instructions and processes responses. The executor, working as the controller of all actions, interprets the LLM instructions to activate the corresponding modules and collects their responses for the LLM. For example, the executor can fetch a text from the memory and append it to the context of LLM, or extract an excerpt from the context and append it to the memory.

---


[*]Equal contribution. Alphabet order.
[†]Work done during ByteDance Research internship.

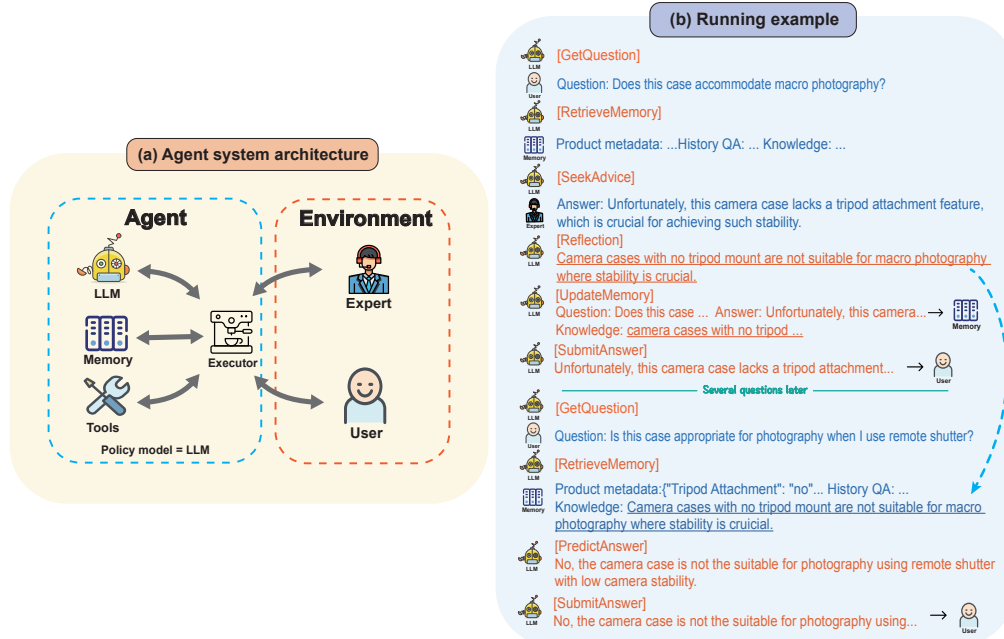
38th Conference on Neural Information Processing Systems (NeurIPS 2024).


Figure 1: (a) Architecture of our agent system, including LLM, memory, tools, and executor. (b) A running example of AGILE in a customer service QA environment. The tokens (actions) generated by the LLM are in orange color and the tokens appended by the executor are in blue color.

The executor can also follow instructions of the LLM to utilize a search tool. In addition to skills such as reasoning, planning, and reflection, we propose a new ability called *seeking advice*, which means that the agent proactively consults human experts when it encounters a problem unsolvable. The agent can reflect on the expert feedback and memorize it for future use. Furthermore, we propose a training method based on reinforcement learning (RL), which simultaneously trains the policy of invoking different modules and the reasoning, planning, reflection, and seeking advice abilities of the LLM agent in an end-to-end fashion.

While the proposed agent framework is general, in this paper, we evaluate it in complex question answering (QA). It is a task an LLM agent has the potential of outperforming existing solutions such as the use of an LLM alone. However, existing QA benchmarks [12, 49, 11, 27] are designed for specific subsets of capabilities (e.g., reflection, memory retrieve, etc.) which cannot simultaneously investigate the ability to combine all modules and capabilities of the agent.

To address this, we have developed a new benchmark called ProductQA. ProductQA comprises 88,229 question-answer pairs in customer service divided into 26 QA tasks, each corresponding to a distinct Amazon product category. This benchmark is based on real Amazon user queries and includes fact-based questions, reasoning questions, and product recommendation queries. It comprehensively evaluates agents' abilities to handle historical information and accumulated knowledge, leverage tools, interact with humans, perform self-evaluation, and conduct reflection. Additionally, the training and testing tasks are made disjoint to assess the agent's ability to adapt to new product categories.

We evaluate our agent framework on three tasks, ProductQA, MedMCQA [27] and HotPotQA [49]. For ProductQA, we use a two-stage training method based on Vicuna-13b [6]. In the first stage, imitation learning is employed to create `agile-vic13b-sft`. In the second stage, the policy gradient algorithm of PPO [37] produces `agile-vic13b-ppo`. Experimental results show that `agile-vic13b-ppo` improves the relative total performance score by 9.2% over GPT-4 and by 90.8% over GPT-3.5. Ablation studies confirm that all modules in Figure 1 are indispensable. Specifically, removing tools or memory usage negatively impacts the agent's performance, leading to a 25.9% or 17.4% increase in seeking advice, respectively, or a 9.3% or 4.0% relative decrease in the total score, respectively. Disabling the seeking advice function results in a 10.7% decrease in accuracy. Finally, `agile-vic13b-ppo` achieves a 2.3% relative increase in total score compared to `agile-vic13b-sft`, demonstrating the necessity of PPO training. On MedMCQA, we train an `agile-mek7b-ppo` agent, initialized from Meerkat-7b [17], following the same two-stage procedure.

2

Our agent improves the base LLM's accuracy from 53.4% to 85.2% by seeking advice on 31.6% instances. This accuracy surpasses the SOTA accuracy of 79.1% by GPT4-MedPrompt [25]. When all agents are able to seek advice, our agent also outperforms the GPT-4 agent in terms of the total score. For HotPotQA, we use the same two-stage method to train `agile-vic13b-ppo` from Vicuna-13b. Our agent achieves 67.5% accuracy, surpassing the strongest baseline of 48.2%, by seeking advice on 15.6% of instances. When advice-seeking is enabled for all agents, our agent outperforms GPT-4 by 10.8% in total score.

The main contributions of this paper are summarized as follows:

- We propose a novel reinforcement learning framework of LLM agents. It facilitates end-to-end learning of agents. Notably, this framework enables the agent to proactively seek advice from human experts, providing two advantages: 1) It ensures high-level accuracy when dealing with complex and challenging questions, and 2) it fosters learning from humans, thereby enhancing its abilities to adapt to new tasks.
- We develop a benchmark, ProductQA, to comprehensively evaluate the agent's capabilities in complex question answering.
- We perform experiments on multiple tasks to verify our framework and show that AGILE agents based on 13B and 7B LLMs trained with PPO can surpass GPT-4 agents.

## 2 Methods

### 2.1 RL formulation of agent

Our agent framework comprises four elements: LLM, memory, tools, and executor, see Figure 1(a). The LLM possesses a *context*, defined as the sequence of tokens it utilizes to generate the next token. In RL terminology, the agent conducts a token-level Markov decision process (MDP). The action space $\mathcal{A}$ corresponds to the LLM's vocabulary, with each token representing an action. Hence, the LLM serves as the *policy model*. The agent's state consists of the (context, memory) pair. Upon predicting a new action $a_t$ (i.e., a new token), the LLM transfers control to the executor. The executor applies predefined logic to transition from the current state $s_t$ to the next state $s_{t+1}$, implementing the state transition function $\mathcal{S} \times \mathcal{A} \to \mathcal{S}$ in RL, and then returns control to the LLM to predict the next action. Concurrently, the environment issues a reward $r(s_t, a_t)$.

Let us examine the state transition more closely. For each action, the executor's first operation is to append the token to the context, preparing the LLM for generating the next token. Then, the executor checks a registered list of *functions*. Each function is designed to execute a set of operations, including memory I/O, tool usage, and interaction with the environment. If the action (i.e., the token) matches a function name, the executor will execute the associated function implementation, further mutating the agent state. For instance, if the token is `[GetQuestion]`, the executor will prompt the user for a new question and append it to the context; if the token is `[UpdateMemory]`, the executor will write a specific segment of the context into the memory; if the token is `[ClearContext]`, the executor will reset the context to `[BOS]`. In summary, the LLM interacts with the memory and tools by predicting function names, relying on the executor to execute these functions. See Table 1 for a full list of functions defined for a QA agent and see Figure 1(b) for a running example.

### 2.2 Policy learning

We frame the policy learning problem as a task of training a language model. Consider an agent trajectory $\tau = (s_1, a_1, ..., s_n, a_n)$, we derive a *training sequence* denoted as $(e_1, ..., e_n)$, where $e_i$ represents the tokens that the executor appends to the context at step $i$. If $a_i$ is a function name token, then $e_i$ is the concatenation of $a_i$ and extra tokens appended by the function execution; otherwise, $e_i = a_i$. In this sequence, $\{a_1, ..., a_n\}$ (the first token of each $e_i$) are referred to as action tokens. The LLM context at step $i$, denoted by $c_i$, is a subsequence of the prefix $(e_1, ..., e_{i-1})$; $c_i$ may be shorter than $(e_1, ..., e_{i-1})$ because the executor can delete context tokens.

In Imitation Learning (IL), we generate trajectories by observing human experts or more proficient agents, then we derive the training sequences to fine-tune the LLM. It is important to point out that (1) the loss is calculated on the action tokens only, and (2) $c_i$ should serve as the attention mask for tokens in $e_i$, as it reflects the true context perceived by the LLM at the time of action prediction. In

3

Table 1: Functions for an exemplary customer service QA agent. Among them, [Reflection] and [PredictAnswer] are trivial functions, as the executor passes control immediately back to the LLM to start generating result tokens.

| Function name | Function implementation |
|---|---|
| [GetQuestion] | Prompt the user for a question and append it to the context. |
| [RetrieveMemory] | Retrieve relevant entries from the memory and append them to the context. |
| [SeekAdvice] | Ask human experts for advice and append it to the context. |
| [Reflection] | $\emptyset$ |
| [UpdateMemory] | Write a specific segment of the context into the memory. |
| [SearchProduct] | Extract a search query from the context, then invoke the search tool and append results to the context. |
| [PredictAnswer] | $\emptyset$ |
| [SubmitAnswer] | Extract a predicted answer from the context and submit it to the user. |
| [ClearContext] | Reset the context to a single token [BOS]. |

reinforcement learning (RL), we treat the LLM as the policy model, from which training sequences can be sampled and individual action tokens are assigned rewards. Consequently, the LLM can be optimized using policy gradient methods, such as PPO [37]. Analogous to the IL setup, we apply policy gradient updates exclusively to the action tokens and employ $c_i$ as the attention mask.

In some situations, an agent may produce very long trajectories, potentially yielding training sequences that span millions of tokens and are impractical for training. We can leverage the structure of the trajectory to partition it into smaller segments. For instance, if the agent resets its LLM context at the beginning of every QA session, then we can partition by the session boundary. Nevertheless, these sessions are not entirely independent; actions taken in earlier sessions can influence memory, creating lasting effects on subsequent sessions. To tackle this challenge of long-range dependencies, we propose a training algorithm detailed in Appendix A.

## 2.3 Interaction with human experts

Our agent framework enables the agent to proactively seek advice from human experts. For example, the agent can invoke a [SeekAdvice] function to request expert advice. This approach helps in two ways. Firstly, the agent can request the correct answer when its confidence is low, ensuring sufficient accuracy for the application. Secondly, the agent can use [Reflection] to distill general knowledge from the expert advice before storing it in memory. This accumulation of knowledge allows the agent to adapt to new tasks that it has not encountered during training.

Seeking advice involves complex decision-making. The agent must estimate its own confidence in the current session, predict the potential value of the advice for future sessions, and consider the cost of human resources. The optimal trade-off is difficult to annotate manually but aligns well with our RL framework. Specifically, the present risk, future value, and cost of action can all be represented as RL rewards, allowing this skill to be trained as part of the policy model on an end-to-end basis.

## 3 The ProductQA dataset

We believe that product question answering in a real online shopping environment offers a comprehensive challenge for evaluating LLM agents. First, it demands expert knowledge about millions of products, including their technical specifications, usage in particular scenarios, and compatibility with other products. Second, answering some questions requires the use of tools, such as a product search tool. Third, the continuous emergence of new products necessitates the adaptability of the agent. This has motivated the creation of the ProductQA dataset. Unlike existing online shopping QA datasets [38, 8], which primarily focus on questions about product metadata or page information, ProductQA features more complex queries involving reasoning, expert knowledge, and tool usage (e.g., SQL), providing a comprehensive assessment of an agent's capabilities.

The ProductQA dataset consists of 26 QA tasks, each representing a distinct group of products within a specific category. Each group encompasses 17-20 products. We collected 20 groups for training and 6 for testing, allowing for assessing the agent's adaptability to new tasks. We collected an average of 3,393 question-answer pairs for each product group. The questions within the same

4

Table 2: An example of an information table for the headphones group.

| Product ID | Title | Price | Brand | Headphone Type | Cable Type | Audio Transmission | Audio Output Mode | ... |
|---|---|---|---|---|---|---|---|---|
| B00WSLZFTK | Sennheiser RS 170 | $11.03 | Sennheiser | over-ear | bluetooth | kleer | stereo | ... |
| B003AIL2HE | JVC HAEB75B | $9.99 | JVC | earbud | 3.5mm Jack | analog | bass boost | ... |
| B01C22IJV0 | Phaiser BHS-530 | $6.04 | Phaiser | earbud | bluetooth | bluetooth | stereo | ... |
| B0013OWPV4 | JVC HARX700 | $2.00 | JVC | over-ear | 3.5mm Jack | analog | stereo | ... |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |

Table 3: Examples of Fact-QA, Search-QA and Reasoning-QA in ProductQA.

| Type | Question | Long Answer | Short Answer |
|---|---|---|---|
| Fact-QA | What is the size of the neodymium driver used in the JVC HA-EB75 headphones? | The JVC HA-EB75 headphones contain a 13.5 mm neodymium driver in each earpiece, which contributes to the enhanced sound quality. | 13.5 mm |
| Search-QA | I'm an audiophile always on the move, so I need my music non-stop. Tell me, what's the headphone with the longest playtime you have, either on-ear or in-ear? | I found a product that matches your criteria. 'ABCShopUSA Wireless Earbuds True' with asin: B00LJT2EPK | B00LJT2EPK |
| Reasoning-QA | Will these headphones deliver comparable sound quality to wired alternatives when I am editing music? | No, these headphones may not suit your needs for music editing since they are wireless and can introduce audio compression and slight latency. Such issues can impact the precise listening experience crucial for professional audio editing tasks. | no |

group are correlated, as knowledge from one answer may aid in addressing other questions. The dataset statistics are presented in Table 12.

The dataset is annotated by 20 professional annotators, each with at least a college degree, employed by a commercial data annotation company. We pay the company at market rates for professional annotation. See annotation guidelines in Appendix F.2. In addition, we will release the code for the data pre-processing before human annotation.

## 3.1 Product collection

We gather products from the Amazon Review Data [23], which includes product metadata as well as reviews. We initially filter the Amazon Review Data to retain only popular products with at least 100 reviews, then cluster them by category tags. From these clusters, we select 26 based on the size of the cluster, each defined as a *product group*. Subsequently, we sample products from each product group. See Appendix F.1 for more details about product group and product selection.

After the products are collected, annotators compile an information table for each product group. An example of such a table is presented in Table 2. To enhance the efficiency of the annotation process, we employ GPT-4 to extract as many product features as possible from the reviews. These features, together with the product metadata, are provided to the annotators for table creation.

## 3.2 QA collection

We identify three predominant types of questions in online shopping contexts: 1) **Fact-QA**: questions concerning specific product details; 2) **Search-QA**: searches for product recommendations tailored to user preferences; 3) **Reasoning-QA**: questions whose answers require domain-specific reasoning, such as the implications of a product feature. Accordingly, we annotate question-answer pairs for these types. Each question is annotated with both a detailed paragraph-long answer and a concise short answer. The long answer should resemble a response from human customer service, while the short answer consists of a few words. We train the model to predict both answer types. The accuracy of the long answers is evaluated using GPT-4 (see Appendix J for the prompt); the short answers are assessed by exact match and are used for defining rewards for RL training.

**Fact-QA** Fact-QAs are constructed from product reviews. For each product, we provide GPT-4 with a batch of 30 reviews, prompting it to generate 20 questions and their corresponding answers

5

before moving on to the next batch. We encourage GPT-4 to create diverse questions. The results are then given to annotators to refine and finalize the question-answer pairs.

**Search-QA**  Starting with an information table for a given product group, we generate random SQL expressions using a set of predefined rules. These expressions are then translated into natural language questions by GPT-4. The answers are obtained by executing the SQL queries. Subsequently, human annotators thoroughly revise the QA pairs.

**Reasoning-QA**  As the first step, we collect professional knowledge for each product group. To enhance efficiency, we utilize GPT-4 to generate candidate knowledge entries based on the technical specifications from the information table. These entries are then curated and refined by human annotators. Here is an example of a knowledge entry: *Motherboards with the ATX form factor are ideally suited for high-performance computing tasks and gaming, due to their ample expansion slots for graphics cards and other peripherals that boost computing capabilities.* Finally, annotators develop question-answer pairs from these knowledge entries.

## 4 Experiments

### 4.1 Experimental setting

**Dataset**  We evaluate our agent on three complex QA tasks: ProductQA, MedMCQA and HotPotQA. MedMCQA [27] is a dataset for multiple-choice QA. It consists of questions from medical school entrance examinations. HotPotQA [49] features natural, multi-hop questions, which challenge an agent's ability to perform reasoning and utilize search tools. For both MedMCQA and HotPotQA, we report results on their respective full dev sets.

**Agent definition**  Our agent can invoke functions defined in Table 1. In a typical workflow, the agent prompts the user for a new question at the session start. It can then retrieve memory to get relevant information. The memory can be initialized as empty (ProdcutQA) or with domain knowledge (QA pairs from MedMCQA training dataset). The agent has the option to use external tools, such as product search in ProductQA and article search in HotPotQA), to gather more information. At last, the agent decides whether to predict an answer directly or seek human advice. If the agent seeks advice, it obtains a human answer (ground-truth answer in our setting). The agent can then optionally use a reflection round to extract general knowledge from the human answer, writing both the human answer and the reflected knowledge to its memory. Finally, the agent submits an answer to the user. In our setting, submitting a correct answer incurs a $+1$ reward, while submitting a wrong answer incurs a $0$ reward. Seeking human advice has a fixed $-c$ reward, where $c$ represents *seeking advice cost*. Assuming the human advice always contains a correct answer, then the possible total rewards are $\{0, 1, 1 - c\}$.

**Training**  The training consists of two stages. First, we construct trajectories from the training data and employ imitation learning to train the agent. Then we apply Algorithm 1 for further optimization by reinforcement learning. See Appendix B for implementation details. For ProductQA and HotPotQA, the agent's LLM is initialized from Vicuna-13b-1.5. For MedMCQA, we use Meerkat-7b [17], a medical LLM trained with high-quality CoT reasoning paths from 18 medical textbooks and diverse instruction-following datasets. We fine-tune the model for 2 epochs with a learning rate of 1e-5 and a batch size of 64. We implement PPO for 1 epoch with a learning rate of 1e-6 and a batch size of 64. The training runs on NVIDIA-H800. Training times and the number of GPUs for each experiment are reported in Table 13. The LLM is fully trained without using LoRA.

**Evaluation and baselines**  We report three metrics for the agent: (a) Advice rate: the rate of seeking human advice; (b) Accuracy: the rate of predicting the correct answer; (c) Total score: the average reward across all sessions, taking the advice rate and the accuracy both into account.

We compare our agent against two types of baselines: 1) Prompting GPT-3.5 (gpt-3.5-turbo-0301) and GPT-4 (gpt-4-0613) [26] to directly answer the question, without working in an agent manner, noted as `gpt3.5-prompt` and `gpt4-prompt`. 2) Prompting GPT-3.5 and GPT-4 within the AGILE framework, noted as `agile-gpt3.5-prompt` and `agile-gpt4-prompt`. We carefully designed prompts for all baselines and they are shown in Appendix J.

6

Table 4: Results on ProductQA. Here, `X-prompt` represents directly prompting model X; `agile-X-Y` incorporates model X within the AGILE framework, while Y represents prompting or PPO training. We report results on short and long answers, respectively. The seeking advice cost is $c = 0.3$. Results are averaged over six test tasks. See Table 14 for individual product group performance.

| Method | Advice Rate $\downarrow$ | Accuracy $\uparrow$ | | Total Score $\uparrow$ | |
|---|---|---|---|---|---|
| | | Short | Long | Short | Long |
| gpt3.5-prompt | - | 0.202 | 0.322 | - | - |
| gpt4-prompt | - | 0.464 | 0.571 | - | - |
| agile-vic13b-prompt | 0.174 | 0.174 | 0.294 | 0.122 | 0.242 |
| agile-gpt3.5-prompt | 0.323 | 0.508 | 0.644 | 0.411 | 0.547 |
| agile-gpt4-prompt | 0.208 | 0.780 | 0.809 | 0.718 | 0.747 |
| agile-vic7b-ppo (ours) | 0.179 | 0.818 | 0.800 | 0.764 | 0.746 |
| agile-vic13b-ppo (ours) | 0.233 | **0.854** | **0.854** | **0.784** | **0.784** |

## 4.2 Results on ProductQA

As Table 4 shows, our AGILE agent outperforms all baselines on ProductQA. Notably, the average total score of `agile-vic13b-ppo` across six test groups shows a relative improvement of 9.2% in short answers and 5.0% in long answers to `agile-gpt4-prompt` where the seeking advice cost is added into the prompt. Concretely, `agile-vic13b-ppo` uses a comparable number of seeking advice to achieve 7.4% higher accuracy in short answers than `agile-gpt4-prompt`, and as Figure 2 shows, this accuracy improvement is consistent across the whole trajectory. Our `agile-vic7b-ppo` agent also outperforms `agile-gpt4-prompt` in average total scores. Note that the GPT-4 agent knows the seeking advice cost from its prompt (see Figure 7).

We investigate the impact of varying the seeking advice cost. As shown in Figure 3, when the cost decreases, both the advice rate and the accuracy increase, indicating greater utilization of human assistance. Specifically, with a high cost of 0.5, the advice rate is close to 0, and at a low cost of 0.1, the accuracy is close to 1. This result demonstrates that by adjusting the cost and through RL training, we can effectively manage the trade-off between accuracy and human cost. For instance, the agent can achieve 94.1% accuracy on the Motherboards task with a seeking advice cost of $c = 0.1$ (refer to Table 16). This capability is especially important in realistic scenarios that demand high accuracy levels. In most experiments, we set the cost at a medium level with $c = 0.3$.

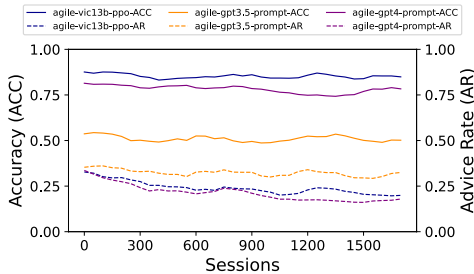



Figure 2: Accuracy and advice rate over the following 200 sessions ($c = 0.3$).

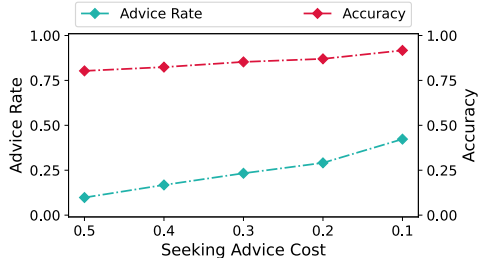



Figure 3: Advice rate, accuracy along with seeking advice cost $c$ on ProductQA.

To validate the accuracy of GPT-4 evaluator in assessing the long answer results, we randomly select 100 triplets (questions, reference long answer, model-predicted long answer) and manually labeled the correctness. The results show a 94% agreement rate between the GPT-4 evaluator and the author.

**Ablation study** We present ablation studies in Table 5 to assess the contributions of individual agent components and the effects of RL training. The table indicates that disabling the option to seek advice (w/o Advice) leads to a 10.7% drop in accuracy and a 5.0% relative reduction in total score. Forcing the agent to seek advice at the initial part of the trajectory (Non-adapt Advice) causes a 4.2% decrease in accuracy, underscoring the value of adaptive decision-making. Removing reflection and memory capabilities (w/o Memory and w/o Reflection) both increase the frequency of advice-seeking, as the agent struggles to accumulate or leverage valuable knowledge, consequently decreasing the

7

Table 5: Ablation studies for disabling reflection, memory, seeking advice, tool use, or RL training. Here, `non-adapt-advice` means that seeking advice is invoked for the first $K$ sessions of the trajectory, where $K$ equals to the number of [SeekAdvice] performed by `agile-vic13b-ppo`. See Table 15 for ablation results on individual product groups.

| Method | Advice Rate ↓ | Accuracy ↑ | Total Score ↑ |
|---|---|---|---|
| w/o Reflection | 0.270 | 0.852 | 0.771(-1.7%) |
| w/o Memory | 0.407 | 0.876 | 0.754(-4.0%) |
| w/o Advice | 0.000 | 0.747 | 0.747(-5.0%) |
| non-adapt-advice | 0.233 | 0.812 | 0.742(-5.7%) |
| w/o Tool-Use | 0.492 | 0.864 | 0.717(-9.3%) |
| w/o RL | 0.256 | 0.843 | 0.766(-2.3%) |
| agile-vic13b-ppo (ours) | 0.233 | **0.854** | **0.784** |

total score. Furthermore, disabling tool use (w/o Tool-Use) causes a substantial 25.9% increase in the advice-seeking rate because the agent's capabilities are diminished, making it more reliant on external advice. Lastly, RL training improves the relative total score by 2.3%, lowers the advice-seeking rate, and boosts accuracy, demonstrating that RL training effectively optimizes the policy. Additional results on RL training can be found in Appendix C.

In Appendix E, we present detailed examples of `agile-vic13b-ppo` illustrating how memory, tools, seeking advice, and reflection enhance the agent workflow.

**Trend of advice rate** Figure 4 demonstrates a consistent decrease in the advice rate of `agile-vic13b-ppo` as more sessions are added to the trajectory. This decline can be attributed to the agent progressively accumulating knowledge and becoming more independent. Additionally, the figure illustrates that disabling RL training or reflection leads to a significant increase in the advice rate, underscoring the importance of RL training and reflection in reducing human costs.

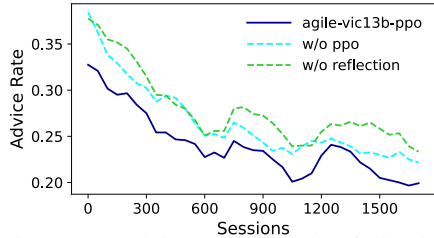



Figure 4: Advice rate over the following 200 sessions on ProductQA ($c = 0.3$).

## 4.3 Results on MedMCQA

Table 6: Results on the MedMCQA dev dataset. `X-prompt` represents directly prompting the model X; `agile-X-Y` represents incorporating the model X within the AGILE framework, while Y represents prompting, ablation studies or standard PPO training. The seeking advice cost is $c = 0.4$.

| Method | Advice Rate ↓ | Accuracy ↑ | Total Score ↑ |
|---|---|---|---|
| Meerkat-7b-prompt | - | 0.534 | - |
| gpt3.5-prompt[24] | - | 0.501 | - |
| gpt4-prompt[24] | - | 0.695 | - |
| gpt4-Medprompt[25] | - | 0.791 | - |
| agile-gpt3.5-prompt | 0.194 | 0.697 | 0.619 |
| agile-gpt4-prompt | 0.421 | **0.884** | 0.721 |
| agile-mek7b-w/o Reflection | 0.368 | 0.790 | 0.643 |
| agile-mek7b-w/o Memory | 0.506 | 0.741 | 0.539 |
| agile-mek7b-w/o Advice | 0.000 | 0.620 | 0.620 |
| agile-mek7b-w/o RL | 0.322 | 0.837 | 0.708 |
| agile-mek7b-ppo (ours) | 0.316 | 0.852 | **0.726** |

Our `agile-mek7b-ppo` agent, based on the smaller Meerkat-7b [17] model, reaches an accuracy of 85.2% with an advice rate of 31.6%. As Table 6 shows, this represents a 31.8% accuracy increase over the base model `Meerkat-7b-prompt` and a 6.1% increase over the state-of-the-art `gpt4-Medprompt` [25]. Table 6 also shows that the ability to seek advice alone contributes a 23.2% accuracy gain, meaning that each instance of seeking advice corrects an average of 0.73 prediction errors. This indicates that PPO training effectively helps the agent identify its mistakes. For a fair comparison, we also evaluate `agile-gpt3.5-prompt` and `agile-gpt4-prompt`, which incorporate GPT-3.5

8

and GPT-4 within our AGILE framework. These agents also leverage advice-seeking to enhance accuracy, but without RL training, their total scores are lower than `agile-mek7b-ppo`. Finally, through ablation studies, we confirmed the essential roles of memory, reflection, seeking advice, and RL training in achieving high performance. Removing these components leads to a significant drop in total scores, detailed in Table 6.

### 4.4 Results on HotPotQA

We compare our method against several baselines. Specifically, we found the original ReAct baseline implementation in [51] to be suboptimal. By reproducing their results with GPT-4 (ReAct-gpt4-prompt), we observed improved performance. As shown in Table 7, our agile agent outperforms all baselines in accuracy, achieving a 40.0% relative improvement over ReAct-gpt4-prompt, which is the strongest baseline. Additionally, compared to `agile-gpt4-prompt`, the trained `agile-vic13b-ppo` demonstrates both higher accuracy and a lower advice rate, leading to a 10.8% relative increase in total score. Ablation studies confirm that removing either seeking-advice or PPO training results in a significant decrease in the total score.

Table 7: Results on the HotPotQA full dev dataset. `X-prompt` represents directly prompting the model X; `agile-X-Y` represents incorporating the model X within the AGILE framework, while Y represents prompting, ablation studies or standard PPO training. The seeking advice cost is $c = 0.3$.

| Method | Advice Rate ↓ | Accuracy ↑ (Exact Match) | Accuracy ↑ (GPT-4 Evaluator) | Total Score ↑ (Exact Match) |
|---|---|---|---|---|
| ReAct [51] | - | 0.351 | - | - |
| ReAct-gpt4-prompt | - | 0.482 | - | - |
| CRITIC [9] | - | 0.443 | - | - |
| Expel [54] | - | 0.390 | - | - |
| AutoAct [32] | - | 0.384 | - | - |
| agile-gpt4-prompt | 0.194 | 0.664 | 0.842 | 0.567 |
| agile-vic13b-w/o Advice | 0.000 | 0.553 | 0.751 | 0.553 |
| agile-vic13b-w/o RL | 0.171 | 0.668 | 0.857 | 0.617 |
| agile-vic13b-ppo (ours) | 0.156 | **0.675** | **0.858** | **0.628** |

## 5 Related work

Table 8: Related work on LLM agents. AGILE stands out as the pioneering work that trains the entire agent using reinforcement learning, incorporating proactive human advice-seeking.

| LLM Agent | LLM | SFT | RL | Memory | Tools | Reflection | Proactive Human-agent Interaction |
|---|---|---|---|---|---|---|---|
| WebGPT [22] | GPT-3 | ✓ | ✓ | ✗ | ✓ | ✗ | ✗ |
| ReAct [51] | PaLM-540b | ✓ | ✗ | ✓ | ✓ | ✗ | ✗ |
| Reflexion [40] | GPT-3/3.5/4 | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ |
| ChatDev [30] | ChatGPT-turbo-16k | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ |
| RAP [14] | LLaMA-33b | ✗ | ✗ | ✓ | ✗ | ✗ | ✗ |
| AutoAct [32] | LLaMA2-70b | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| TPTU [35] | ChatGPT/InternLM | ✗ | ✗ | ✓ | ✓ | ✓ | ✗ |
| AGILE (Ours) | Vicuna-13b/Meerkat-7b | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

**LLM agents** Large Language Models (LLMs) have demonstrated substantial capabilities in following instructions, reasoning, and planning. Numerous research works, as shown in Table 8, utilizing prompt engineering, have constructed remarkable LLM agents capable of autonomously resolving complex tasks across various environments [28, 44, 2, 30, 4]. Furthermore, extensive works identify key components in the design of LLM agents, including planning [22, 39, 10, 32, 51, 35], tool-use [19, 29, 48, 36], and reflection [40, 21]. In this work, we enable the agent to utilize memory, tools and proactively learn from the environment. We then formulate the entire process within an RL framework so that all agent skills can be jointly optimized end-to-end.

9

**Human-agent interaction**   Although LLMs face practical challenges, such as hallucination [53] and a lack of long-tail knowledge [16], consulting human experts can help mitigate these issues. Several studies [52, 46] have incorporated human experts into agent workflows relying on passive feedback or predefined rules. However, these approaches do not involve proactively seeking advice, which requires more complex decision-making. While [5, 31] train models to ask questions using behavior cloning, they ignore the fact that the decision to seek advice must be based on the LLM's own knowledge and capabilities [55, 18, 13]. [34] use a calibrated version of an LLM's token probabilities as a confidence measure, yet token probabilities tend to be overconfident [47], and existing calibration methods don't generalize well to our agent setting when the LLM makes multiple decisions in sequence. Ultimately, the challenge of seeking advice is tied to the LLM's self-evaluation, which is difficult to ground truth or optimize through SFT. In our RL framework, the value and cost of seeking advice can be directly represented as RL rewards, enabling the proactive skill of seeking advice to be optimized as part of the policy model on end-to-end RL training.

**LLM agent benchmarks**   Several benchmarks have been designed to assess the capabilities of agents. For instance, the Webshop [50] and Mind2Web [7] datasets evaluate agents' tool usage and planning abilities within a web environment. HotPotQA [49] and TriviaQA [12] focus on agents' reasoning and tool usage for question answering. ALFWorld [41] examines planning and navigation skills, while ScienceWorld [43] provides an interactive text-based environment to evaluate agents' scientific aptitude. As illustrated in Table 9, despite these existing benchmarks, none comprehensively addresses all the core challenges of real-world agent applications, such as handling long-tail knowledge, human-agent interaction, long-term memory usage, tool usage, self-evaluation, and reflection. This motivated us to develop ProductQA.

Table 9: Benchmarks for evaluating LLM agents. ProductQA features long trajectories, tool use, long-term knowledge accumulation, and cross-task capabilities.

| Datasets | Type | Fields | Size | Long Trajectory | Tool Usage | Long-term Knowledge | Cross Task |
|---|---|---|---|:---:|:---:|:---:|:---:|
| Webshop [50] | Simulator | Web | 12,087 | ✗ | ✗ | ✗ | ✗ |
| Mind2Web [7] | Simulator | Web | 2,350 | ✗ | ✗ | ✗ | ✓ |
| ALFWorld [41] | Simulator | Navigation | 3,827 | ✗ | ✗ | ✗ | ✓ |
| ScienceWorld [43] | Simulator | Science | 7,207 | ✗ | ✗ | ✗ | ✗ |
| HotPotQA [49] | QA | Wikipedia | 112,779 | ✗ | ✓ | ✗ | ✗ |
| TriviaQA [12] | QA | Web | 95,956 | ✗ | ✓ | ✓ | ✗ |
| ProductQA (ours) | QA | E-commerce | 88,229 | ✓ | ✓ | ✓ | ✓ |

# 6   Conclusion and future work

In this work, we introduce a novel reinforcement learning framework of LLM agents, called AGILE. First, the whole system of AGILE is trained end-to-end by reinforcement learning. Second, AGILE has the ability of seeking advice from external human experts. In addition, we develop a challenging dataset of complex QA, ProductQA, for comprehensive evaluation of an agent's capabilities. Extensive experiments demonstrate that within our framework, an agent based on a smaller model after RL training can outperform GPT-4.

AGILE is a general agent framework and we can certainly consider multiple extensions of it. An agent can be equipped with more tools, such as multimodal perception, manipulations in physical environments, logical reasoning, among others. We posit that AGILE's activities can be categorized into two distinct types: utilizing its LLM alone, and integrating the LLM with other tools. These two approaches conceptually align with the human cognitive processes known as System 1 and System 2 [15, 1]. Furthermore, AGILE's memory serves as a repository for the accumulation of experiences and knowledge, which is crucial for self-improvement. Consequently, AGILE offers an architecture for an very powerful agent that has the potential to attain human-level intelligence.

AGILE also includes interactions between the agent and external human experts. The framework can be extended to allow interactions with humans or machine agents in various roles such as students or teachers, and in different formats such as debates or coordination. Furthermore, AGILE can be employed in multi-agent systems.

10

## Acknowledgements

The authors thank anonymous reviewers for their helpful suggestions.

## References


[1] Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. Deep learning for ai. *Communications of the ACM*, 64(7):58–65, 2021.

[2] Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. Chemcrow: Augmenting large-language models with chemistry tools. *arXiv preprint arXiv:2304.05376*, 2023.

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[4] Guangyao Chen, Siwei Dong, Yu Shu, Ge Zhang, Jaward Sesay, Börje F Karlsson, Jie Fu, and Yemin Shi. Autoagents: A framework for automatic agent generation. *arXiv preprint arXiv:2309.17288*, 2023.

[5] Xiaoyu Chen, Shenao Zhang, Pushi Zhang, Li Zhao, and Jianyu Chen. Asking before action: Gather information in embodied decision making with language models. *arXiv preprint arXiv:2305.15695*, 2023.

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023.

[7] Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36, 2024.

[8] Yang Deng, Wenxuan Zhang, Qian Yu, and Wai Lam. Product question answering in e-commerce: A survey. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11951–11964, 2023.

[9] Zhibin Gou, Zhihong Shao, Yeyun Gong, Yujiu Yang, Nan Duan, Weizhu Chen, et al. Critic: Large language models can self-correct with tool-interactive critiquing. In *The Twelfth International Conference on Learning Representations*, 2024.

[10] Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. *arXiv preprint arXiv:2305.14992*, 2023.

[11] Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421, 2021.

[12] Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*, 2017.

[13] Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

[14] Tomoyuki Kagaya, Thong Jing Yuan, Yuxuan Lou, Jayashree Karlekar, Sugiri Pranata, Akira Kinose, Koki Oguri, Felix Wick, and Yang You. Rap: Retrieval-augmented planning with contextual memory for multimodal llm agents. *arXiv preprint arXiv:2402.03610*, 2024.

[15] Daniel Kahneman. Maps of bounded rationality: Psychology for behavioral economics. *American economic review*, 93(5):1449–1475, 2003.


11


[16] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR, 2023.

[17] Hyunjae Kim, Hyeon Hwang, Jiwoo Lee, Sihyeon Park, Dain Kim, Taewhoo Lee, Chanwoong Yoon, Jiwoong Sohn, Donghee Choi, and Jaewoo Kang. Small language models learn enhanced reasoning skills from medical textbooks. *arXiv preprint arXiv:2404.00376*, 2024.

[18] Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*, 2022.

[19] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[20] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *arXiv preprint arXiv:2211.09110*, 2022.

[21] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems*, 36, 2024.

[22] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[23] Jianmo Ni, Jiacheng Li, and Julian McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197, 2019.

[24] Harsha Nori, Nicholas King, Scott Mayer McKinney, Dean Carignan, and Eric Horvitz. Capabilities of gpt-4 on medical challenge problems. *arXiv preprint arXiv:2303.13375*, 2023.

[25] Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, Renqian Luo, Scott Mayer McKinney, Robert Osazuwa Ness, Hoifung Poon, Tao Qin, Naoto Usuyama, Chris White, and Eric Horvitz. Can generalist foundation models outcompete special-purpose tuning? case study in medicine, 2023.

[26] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[27] Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pages 248–260. PMLR, 2022.

[28] Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22, 2023.

[29] Shishir G Patil, Tianjun Zhang, Xin Wang, and Joseph E Gonzalez. Gorilla: Large language model connected with massive apis. *arXiv preprint arXiv:2305.15334*, 2023.

[30] Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*, 2023.

[31] Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Yankai Lin, Zhong Zhang, Zhiyuan Liu, and Maosong Sun. Tell me more! towards implicit user intention understanding of language model driven agents. *arXiv preprint arXiv:2402.09205*, 2024.



12



[32] Shuofei Qiao, Ningyu Zhang, Runnan Fang, Yujie Luo, Wangchunshu Zhou, Yuchen Eleanor Jiang, Chengfei Lv, and Huajun Chen. Autoact: Automatic agent learning from scratch via self-planning. *arXiv preprint arXiv:2401.05268*, 2024.

[33] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks, 2019.

[34] Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*, 2023.

[35] Jingqing Ruan, Yihong Chen, Bin Zhang, Zhiwei Xu, Tianpeng Bao, Guoqing Du, Shiwei Shi, Hangyu Mao, Xingyu Zeng, and Rui Zhao. Tptu: Task planning and tool usage of large language model-based ai agents. *arXiv preprint arXiv:2308.03427*, 2023.

[36] Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36, 2024.

[37] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[38] Xiaoyu Shen, Akari Asai, Bill Byrne, and Adrià de Gispert. xpqa: Cross-lingual product question answering across 12 languages. *arXiv preprint arXiv:2305.09249*, 2023.

[39] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36, 2024.

[40] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[41] Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2020.

[42] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

[43] Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. Science-World: Is your agent smarter than a 5th grader? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics.

[44] 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.

[45] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[46] Hengjia Xiao and Peng Wang. Llm a*: Human in the loop large language models enabled a* search for robotics. *arXiv preprint arXiv:2312.01797*, 2023.

[47] Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *arXiv preprint arXiv:2306.13063*, 2023.

[48] Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Ehsan Azarnasab, Faisal Ahmed, Zicheng Liu, Ce Liu, Michael Zeng, and Lijuan Wang. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*, 2023.



13



[49] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W Cohen, Ruslan Salakhutdinov, and Christopher D Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*, 2018.

[50] Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022.

[51] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

[52] Qiang Zhang, Jason Naradowsky, and Yusuke Miyao. Ask an expert: Leveraging language models to improve strategic reasoning in goal-oriented dialogue models. *arXiv preprint arXiv:2305.17878*, 2023.

[53] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, et al. Siren's song in the ai ocean: a survey on hallucination in large language models. *arXiv preprint arXiv:2309.01219*, 2023.

[54] Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642, 2024.

[55] Kaitlyn Zhou, Dan Jurafsky, and Tatsunori Hashimoto. Navigating the grey area: Expressions of overconfidence and uncertainty in language models. *arXiv e-prints*, pages arXiv–2302, 2023.



14


# Appendix


**A  Session-level optimization algorithm**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**16**

**B  Implementation details of AGILE**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**17**

&nbsp;&nbsp;&nbsp;&nbsp;B.1  ProductQA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  17

&nbsp;&nbsp;&nbsp;&nbsp;B.2  MedMCQA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  18

&nbsp;&nbsp;&nbsp;&nbsp;B.3  HotPotQA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  18

&nbsp;&nbsp;&nbsp;&nbsp;B.4  Defining proxy reward for RL . . . . . . . . . . . . . . . . . . . . . . . . . .  19

**C  Supplementary experimental results on RL training**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**19**

&nbsp;&nbsp;&nbsp;&nbsp;C.1  Training curve . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  19

&nbsp;&nbsp;&nbsp;&nbsp;C.2  Training Robustness . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  20

&nbsp;&nbsp;&nbsp;&nbsp;C.3  Impact of PPO training . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  20

**D  Tables**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**21**

**E  Case study**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**24**

**F  Development of the ProductQA dataset**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**27**

&nbsp;&nbsp;&nbsp;&nbsp;F.1  Product collection . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  27

&nbsp;&nbsp;&nbsp;&nbsp;F.2  Annotation guidelines . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  27

**G  Broader impact**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**28**

&nbsp;&nbsp;&nbsp;&nbsp;G.1  Positive broader impact . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  28

&nbsp;&nbsp;&nbsp;&nbsp;G.2  Negative broader impact . . . . . . . . . . . . . . . . . . . . . . . . . . . . .  28

**H  Limitations**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**28**

**I  Ethical considerations**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**29**

**J  Prompt templates**&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**29**



15


# A    Session-level optimization algorithm

Assume that the entire trajectory $\tau$ can be partitioned into sub-trajectories $(\tau_1, \tau_2, \cdots, \tau_n)$, each referred to as a *session*. For session $i$, let $\mathcal{S}_i$ denote its initial state, where $c_i$ is the LLM context before the session starts, and $m_i$ is the memory before the session starts. In this section, we will explain how to transform a trajectory-level RL optimization algorithm into a session-level RL optimization algorithm.

Let $r(\tau)$ represent the total reward of trajectory $\tau$, and let $\pi_\theta$ be a policy parameterized by $\theta$. The optimization objective is to maximize the following expectation:

$$R(\theta) = \mathbb{E}_{\tau \sim \pi_\theta}[r(\tau)]. \tag{1}$$

For an arbitrary session index $i$, the trajectory $\tau \sim \pi_\theta$ can be sampled in three stages: $\tau_{1:i-1}$, $\tau_i$, and $\tau_{i+1:n}$. These stages represent the sub-trajectory from session 1 to $i - 1$, the sub-trajectory for session $i$, and the sub-trajectory from session $i + 1$ to $n$, respectively. Accordingly, we have

$$
\begin{aligned}
R(\theta) &= \mathbb{E}_{\tau_{1:i-1} \sim \pi_\theta} \left[ \mathbb{E}_{\tau_i \sim \pi_\theta(\cdot|\mathcal{S}_i)} \left[ \mathbb{E}_{\tau_{i+1:n} \sim \pi_\theta(\cdot|\mathcal{S}_{i+1})} [r(\tau_{1:i-1}) + r(\tau_i) + r(\tau_{i+1:n})] \right] \right] \\
&= \mathbb{E}_{\tau_{1:i-1} \sim \pi_\theta} \left[ r(\tau_{1:i-1}) + \mathbb{E}_{\tau_i \sim \pi_\theta(\cdot|\mathcal{S}_i)} \left[ r(\tau_i) + V_{\pi_\theta}(\mathcal{S}_{i+1}) \right] \right].
\end{aligned}
\tag{2}
$$

Here, $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$ denote the initial states of sessions $i$ and $i + 1$ respectively. The term $r(\tau_{1:i-1})$ represents the total reward accumulated from session 1 to $i - 1$, while $r(\tau_i)$ is the reward obtained in session $i$. Additionally, $V_{\pi_\theta}(\mathcal{S}_{i+1})$ represents the value function at state $\mathcal{S}_{i+1}$ with respect to policy $\pi_\theta$, indicating the expected total reward the agent expects to receive in the future. Averaging over all session indices, Eq. (2) gives:

$$R(\theta) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\tau_{1:i-1} \sim \pi_\theta} \left[ r(\tau_{1:i-1}) + \mathbb{E}_{\tau_i \sim \pi_\theta(\cdot|\mathcal{S}_i)} \left[ r(\tau_i) + V_{\pi_\theta}(\mathcal{S}_{i+1}) \right] \right]. \tag{3}$$

In Eq. (3), the parameter $\theta$ appears in three places – two expectations and a value function – making optimization challenging. To simplify the problem, we assume a base policy $\theta_k$ and define a proximal objective $R(\theta|\theta_k)$, where $\theta$ only appears in the session-level expectation:

$$R(\theta|\theta_k) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\tau_{1:i-1} \sim \pi_\theta} \left[ r(\tau_{1:i-1}) + \mathbb{E}_{\tau_i \sim \pi_\theta(\cdot|\mathcal{S}_i)} \left[ r(\tau_i) + V_{\pi_{\theta_k}}(\mathcal{S}_{i+1}) \right] \right]. \tag{4}$$

$R(\theta|\theta_k)$ is an approximation to $R(\theta)$ in the neighborhood of $\theta_k$. If we employ an iterative optimization procedure:

1. Initialize $\theta_0$ from a reference policy (obtained through SFT).
2. For $k = 0, 1, 2, \cdots$, compute $\theta_{k+1} \leftarrow \arg\max_\theta R(\theta|\theta_k)$.

Then $\theta$ will converge to an (at least locally) optimal policy.

Now we are ready to illustrate why the optimization of $R(\theta|\theta_k)$ can be solved at the session level. Notice that

$$
\begin{aligned}
R(\theta|\theta_k) &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\tau_{1:i-1} \sim \pi_{\theta_k}} \left[ \mathbb{E}_{\tau_i \sim \pi_\theta(\cdot|\mathcal{S}_i)} [r(\tau_i) + V_{\pi_{\theta_k}}(\mathcal{S}_{i+1}) - V_{\pi_{\theta_k}}(\mathcal{S}_i)] \right] \\
&\quad + \mathbb{E}_{\tau_{1:i-1} \sim \pi_{\theta_k}} [r(\tau_{1:i-1}) + V_{\pi_{\theta_k}}(\mathcal{S}_i)] \\
&= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\tau_{1:i-1} \sim \pi_{\theta_k}} \left[ \mathbb{E}_{\tau_i \sim \pi_\theta(\cdot|\mathcal{S}_i)} [r(\tau_i) + V_{\pi_{\theta_k}}(\mathcal{S}_{i+1}) - V_{\pi_{\theta_k}}(\mathcal{S}_i)] \right] + \mathbb{E}_{\tau_i \sim \pi_{\theta_k}} [r(\tau)]
\end{aligned}
$$

On the right-hand side, the first term involves two sampling steps. The first step samples $\tau_{1:i-1} \sim \pi_{\theta_k}$. The inner terms inside the expectation only depends on $\mathcal{S}_i$, thus we can replace it by $\mathcal{S}_i \sim \pi_{\theta_k}$. The second term on the right-hand side is a constant independent of $\theta$. As a result, if we define a *proxy reward*:

$$\tilde{r}_k(\tau_i) := r(\tau_i) + (V_{\pi_{\theta_k}}(\mathcal{S}_{i+1}) - V_{\pi_{\theta_k}}(\mathcal{S}_i)). \tag{5}$$

16

**Algorithm 1** Session-level optimization

---
1: Initialize $\theta_0$ from a reference policy (obtained through SFT).
2: **for** $k \leftarrow 0, 1, 2, \cdots$ **do**
3: &nbsp;&nbsp;&nbsp;&nbsp;Sample a set of trajectories from $\pi_{\theta_k}$, denote the set by $T$.
4: &nbsp;&nbsp;&nbsp;&nbsp;Define or fit a state advantage function from $T$.
5: &nbsp;&nbsp;&nbsp;&nbsp;**for** each $\tau \in T$ **do**
6: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Partition it into sessions $(\tau_1, \tau_2, \cdots, \tau_n)$.
7: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**for** each $\tau_i$ **do**
8: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;Evaluate $\tilde{r}_k(\tau_i)$ by Eq. (5) with the above state advantage function.
9: &nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;**end for**
10: &nbsp;&nbsp;&nbsp;**end for**
11: &nbsp;&nbsp;&nbsp;Treat all sessions as independent, then employ an optimization algorithm (such as PPO) to obtain a new policy $\theta_{k+1}$ by maximizing Eq. (6).
12: **end for**

---

Then, we have

$$R(\theta|\theta_k) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_{\mathcal{S}_i \sim \pi_{\theta_k}} \left[ \mathbb{E}_{\tau_i \sim \pi_\theta(\cdot|\mathcal{S}_i)} [\tilde{r}_k(\tau_i)] \right] + \text{constant}. \tag{6}$$

By Eq. (6), $R(\theta|\theta_k)$ can be optimized by maximizing the average expected proxy reward for each session. The term $A_i := V_{\pi_{\theta_k}}(\mathcal{S}_{i+1}) - V_{\pi_{\theta_k}}(\mathcal{S}_i)$ measures the advantage of state $\mathcal{S}_{i+1}$ over state $\mathcal{S}_i$ with respect to a policy; thus, we call it the *state advantage function*. This function can be either defined by heuristics or fitted by a neural network. In the latter case, one needs to sample trajectories from $\pi_{\theta_k}$, evaluate their rewards, and then use the (state, reward-to-go) pairs to train an estimator for the value function $V_{\pi_{\theta_k}}$.

Finally, we present the session-level optimization algorithm as Algorithm 1. In this algorithm, the state advantage function is the only component that concerns inter-session correlation. While the algorithm is iterative, we anticipate that in practice, the outer loop will require only a few iterations to converge.

## B &nbsp;&nbsp;Implementation details of AGILE

### B.1 &nbsp;&nbsp;ProductQA

**Implementation of** `[GetQuestion]` &nbsp;&nbsp;This function prompts the user for a new question and appends it to the LLM context. Every question is raised for a specific product, thus it has an associated product ID. Based on this ID, the function also appends the product information table's schema and the product metadata to the context.

**Implementation of** `[RetrieveMemory]` &nbsp;&nbsp;This function employs the provided question as a query to retrieve the most relevant historical QA pair and the most relevant knowledge entry from the agent's memory. To safeguard sensitive data from sellers, the agent is restricted to accessing QA records exclusively for the queried product from historical interactions. However, it is permitted to retrieve general knowledge from the whole trajectory since this information is not seller-specific. We utilize an embedding-based retrieval method, specifically employing the all-MiniLM-L6-v2 model [33] as the embedding model.

**Implementation of** `[SearchProdcut]` &nbsp;&nbsp;This function utilizes the LLM to predict a SQL query based on the context, and then invoke a MySQL execution engine. It appends the result to the LLM context. If there is an execution error, then the error is appended to the context too.

**Implementation of** `[SeekAdvice]` &nbsp;&nbsp;This requests for human expert advice and append it to the LLM context. In our implementation, the human expert simply returns the ground truth long answer from the ProductQA dataset.

17

**Implementation of** [PredictAnswer] This function passes control to the LLM to continue generating a long answer and a short answer.

**Implementation of** [Reflection] This function passes control to the LLM to continue generating a reflection result.

**Training Data Generation** We generate training data on a session-by-session basis, where each session consists of a QA pair. A session begins with an initial memory, consisting of historical QA pairs and knowledge entries accumulated from previous sessions. Recall that the [RetrieveMemory] function retrieves only the most relevant QA pair and knowledge entry per session. Thus, in constructing training memories, it suffices to put the retrieved QA pair and the retrieved knowledge entry into the memory. We select them in the following stochastic way: the retrieved QA pair can be the most relevant QA pair from the training set, or a random QA pair, or omitted entirely; similarly for the retrieved knowledge entry.

Based on the initial memory, we generate trajectories by following the agent workflow detailed in Section 4.1. Each trajectory begins with [GetUserQuestion] and [RetrieveMemory]. For QAs classified as Search-QA, a [SearchProduct] function is appended, followed by the corresponding SQL query and its execution result. For other QA types, if an associated knowledge entry exists and is successfully retrieved, the trajectory will extend with a [PredictAnswer] call with the ground truth answer as its result. If the knowledge entry is not retrieved or is absent, we use GPT-4 to evaluate whether the question can be answered with the available context. If affirmative, a [PredictAnswer] with the ground truth answer is appended. Otherwise, the trajectory extends with a [SeekAdvice] call with the ground truth answer as the advice, and a [Reflection] call, where the reflection result is the knowledge entry if it exists, or "no information" if not. Then the reflection result is appended to the memory via [UpdateMemory]. Finally, the trajectory is concluded by [SubmitAnswer].

In this way, we constructed 55,772 session-level trajectories in total, from 6 training tasks in ProductQA. This data is used for imitation learning. In PPO training, we reuse the initial memory data, while the session-level trajectories are generated by the model itself.

## B.2 MedMCQA

For MedMCQA, the memory is initialized with all QA pairs from the training set, simulating that the agent has processed the training set before reaching the test set. We also add a knowledge entry for each QA pair, obtained through GPT-4 reflection (see Figure 12 for the prompt).

**Training data generation** We sample a subset of training data from MedMCQA to construct session-level trajectories. Each trajectory begins with [GetUserQuestion] and [RetrieveMemory]. The [RetrieveMemory] function retrieves the five most relevant QA pairs and pieces of knowledge from the initial memory, using the same embedding similarity search method employed in ProductQA. Then, we prompt GPT-4 to predict an answer with chain-of-thought reasoning. If the GPT-4 answer is correct, we append a [PredictAnswer] call, the GPT-4 chain-of-thought, and the ground-truth answer to the trajectory. If the GPT-4 answer is wrong, which suggests that the question is hard, we append a [SeekAdvice] call with the ground-truth answer, followed by a [Reflection] call with the reflection result generated by GPT-4. Then the reflection result is appended to the memory via [UpdateMemory]. Finally, the trajectory is concluded by [SubmitAnswer]. In this way, we obtain 23,015 session-level trajectories in total.

## B.3 HotPotQA

In the HotPotQA task, the agent has the option to select either [Search], [SeekAdvice] or [PredictAnswer] in each round. Following ReAct [51], the agent first generates reasoning first and then selects an action.

**Implementation of** [Search] This function uses the LLM to generate a search query and invokes a search API. The first result not already present in the LLM context is selected and appended to the existing context.

18

**Training data generation**    We use the HotPotQA training set to construct session-level trajectories. Each trajectory begins with the [GetUserQuestion] prompt. We then repeatedly prompt GPT-4 to predict actions between [Search] and [PredictAnswer]. If GPT-4 predicts [Search], we prompt it to generate a search query and append the corresponding search results to the trajectory, continuing this cycle. This process continues until GPT-4 predicts [PredictAnswer]. If the answer is correct (as evaluated by the GPT-4 evaluator), we replace the predicted answer with the ground-truth answer; otherwise, the data is discarded. Additionally, if GPT-4 predicts [Search] five times in a session, we terminate and discard the data.

Next, for each trajectory, where there are $k$ rounds, we prompt GPT-3.5 using the first $k-1$ rounds as context to decide the final round's action: [PredictAnswer] or [SeekAdvice]. If GPT-3.5 selects [SeekAdvice], we replace the final step with [SeekAdvice] and the corresponding thoughts from GPT-3.5. Otherwise, the original trajectory remains unchanged.

This process results in 10,240 session-level trajectories for the imitation learning stage. For the reinforcement learning stage, we directly use the original HotPotQA training set, consisting of 90,447 samples.

### B.4    Defining proxy reward for RL

In the question-answering tasks, sessions are not independent. Actions taken in earlier sessions can influence memory, creating lasting effects on subsequent sessions. As illustrated in Equation (5), the term $A_i := V_{\pi_{\theta_k}}(\mathcal{S}_{i+1}) - V_{\pi_{\theta_k}}(\mathcal{S}_i)$ measures the advantage of state $\mathcal{S}_{i+1}$ over state $\mathcal{S}_i$ (note that $\mathcal{S}_i$ here represents the initial state of session $i$). In our experiment setting, if the agent predicts [SeekAdvice], it will receive expert advice, extract some knowledge by reflection, and write that knowledge to the memory. Intuitively, $A_i$ should increase if the new knowledge is useful in subsequent sessions, and it should decrease if there is already a lot of similar knowledge in the memory at the start of session $i$. Hence, we use the following heuristic definition,

$$A_i = \beta \frac{\mathbb{I}(N_{i+1:n}(q_i) > 0)}{M_{1:i-1}(q_i) + 1}, \tag{7}$$

where $q_i$ represents the user question in session $i$; $N_{i+1:n}(q_i)$ represents the number of user questions in session $i+1$ to session $n$ that are similar enough to $q_i$; $M_{0:i-1}(q_i)$ represents the number of user questions in session 1 to session $i-1$ that are both similar enough to $q_i$ and added to the memory; $\mathbb{I}(\cdot)$ is the indicator function. $\beta$ is a hyperparameter, we set $\beta = 0.1$ by default.

## C    Supplementary experimental results on RL training

In this section, we present detailed experimental results for RL training on ProductQA.

### C.1    Training curve

In Figure 5, we provide training curves, indicating that RL training converged after 500 steps.

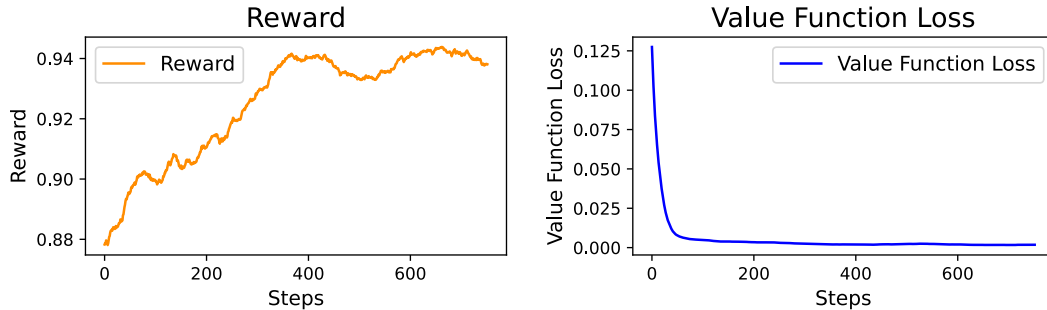



Figure 5: Reward and value function loss curves during the PPO training process on ProductQA.

19

## C.2 Training Robustness

We conduct multiple independent trials of PPO training to study the variation of the result, as shown in Table 10. On average, RL training improves the total score by 2.6%, with a standard deviation of 0.3%, demonstrating the significance of RL improvements.

Table 10: Robustness of RL training. Here, w/o RL represents the agent trained solely by imitation learning. `agile-vic13b-ppo-X` stands for the X-th RL experiment. The table presents the average and standard deviation across multiple RL training runs.

| Method | Advice Rate ↓ | Accuracy ↑ | Total Score ↑ | Relative Improvement to w/o RL |
|---|---|---|---|---|
| w/o RL | 0.256 | 0.843 | 0.766 | - |
| agile-vic13b-ppo-1 | 0.233 | 0.854 | 0.784 | 2.3% |
| agile-vic13b-ppo-2 | 0.226 | 0.855 | 0.787 | 2.7% |
| agile-vic13b-ppo-3 | 0.209 | 0.851 | 0.788 | 2.9% |
| average | 0.223 | 0.853 | 0.786 | 2.6% |
| standard deviation | 0.012 | 0.002 | 0.002 | 0.3% |

## C.3 Impact of PPO training

To further investigate the impact of PPO training in more general and varied scenarios, we conducted additional experiments in two distinct settings.

First, we re-generated SFT training data for `agile-vic13b-sft` such that the agent performs `[SeekAdvice]` randomly in 25% of cases. This initial policy is simpler but more general. In this setting, we name the SFT model `agile-vic13b-sft-random`, and the final model trained with RL on top of it `agile-vic13b-ppo-random`. As shown in Table 11, RL training brings a 7.1% improvement in this setting. Interestingly, the performance of `agile-vic13b-ppo-random` is better than that of `agile-vic13b-ppo`. We conjecture that random seeking-advice is a better initial policy because it enables exploration in all directions.

In the second experiment, we lowered the advice cost to 0.1. After PPO training, as shown in Table 11, the `agile-vic13b-ppo-random` agent quickly adapted to the new cost, performing `[SeekAdvice]` much more aggressively than the initial agent trained by SFT. In this scenario, RL training brings a 22.3% improvement.

Table 11: Improvement of PPO training. The training data for `agile-vic13b-sft` includes trajectories from GPT-4 agent. The training data for `agile-vic13b-random` is constructed by randomly assigning `[SeekAdvice]` to 25% of the data. `agile-vic13b-ppo` and `agile-vic13b-ppo-random` are initialized from `agile-vic13b-sft` and `agile-vic13b-sft-random`, respectively, and both are trained with PPO.

| Method | seeking advice cost | Advice Rate ↓ | Accuracy ↑ | Total Score ↑ |
|---|---|---|---|---|
| agile-vic13b-sft | 0.3 | 0.256 | 0.843 | 0.766 |
| agile-vic13b-ppo | 0.3 | 0.233 | 0.854 | 0.784 (+2.3%) |
| agile-vic13b-sft-random | 0.3 | 0.014 | 0.749 | 0.745 |
| agile-vic13b-ppo-random | 0.3 | 0.306 | 0.89 | 0.798 (+7.1%) |
| agile-vic13b-sft-random | 0.1 | 0.014 | 0.749 | 0.748 |
| agile-vic13b-ppo-random | 0.1 | 0.671 | 0.981 | 0.914 (+22.3%) |

20

# D Tables

Table 12: Statistics of the ProductQA dataset. # Products indicates the number of products within each group. # Fact-QA, # Search-QA and # Reasoning-QA display the respective numbers of QA pairs categorized as Fact-QA, Search-QA, and Reasoning-QA.

| | Groups | # Products | # Fact-QA | # Search-QA | # Reasoning-QA | Total |
|---|---|---|---|---|---|---|
| Train | Blades | 20 | 2,147 | 769 | 631 | 3,547 |
| | Headlight Bulbs | 20 | 1,767 | 644 | 463 | 2,874 |
| | Cell Phones | 20 | 1,636 | 761 | 374 | 2,771 |
| | Portable Power Banks | 20 | 3,344 | 673 | 500 | 4,517 |
| | Dresses | 20 | 2,287 | 738 | 263 | 3,288 |
| | Everyday Bras | 20 | 1,942 | 684 | 336 | 2,962 |
| | Wrist Watches | 20 | 2,169 | 757 | 389 | 3,315 |
| | Blu-ray Players | 20 | 1,630 | 688 | 572 | 2,890 |
| | Camera Lenses | 20 | 1,859 | 769 | 1,025 | 3,653 |
| | Headphones | 20 | 5,432 | 766 | 583 | 6,781 |
| | Mice | 20 | 5,653 | 490 | 294 | 6,437 |
| | Point & Shoot Digital Cameras | 20 | 1,696 | 722 | 565 | 2,983 |
| | Coffee Machines | 20 | 4,184 | 681 | 638 | 5,503 |
| | Digital Scales | 20 | 2,724 | 391 | 682 | 3,797 |
| | Space Heaters | 20 | 2,283 | 674 | 498 | 3,455 |
| | Printers | 20 | 1,431 | 760 | 489 | 2,680 |
| | Litter | 20 | 1,860 | 753 | 507 | 3,120 |
| | Grips | 20 | 1,771 | 713 | 413 | 2,897 |
| | Gun Holsters | 20 | 1,679 | 94 | 1,362 | 3,135 |
| | Handheld Flashlights | 20 | 2,009 | 768 | 482 | 3,259 |
| | Total | 400 | 49,503 | 13,295 | 11,066 | 73,864 |
| Test | Leggings | 20 | 969 | 743 | 527 | 2,239 |
| | Camera Cases | 20 | 975 | 706 | 898 | 2,579 |
| | Motherboards | 20 | 989 | 736 | 826 | 2,551 |
| | All Pans | 20 | 973 | 747 | 275 | 1,995 |
| | Rollerball Pens | 20 | 967 | 760 | 603 | 2,330 |
| | Rifle Scopes | 17 | 979 | 714 | 978 | 2,671 |
| | Total | 117 | 5,852 | 4,406 | 4,107 | 14,365 |

Table 13: Training statistics for each experiment.

| Task | Number of H800 GPU | SFT Training Time | RL Training Time |
|---|---|---|---|
| ProductQA | 8 | 3.6 hours | 5.5 hours |
| MedMCQA | 8 | 0.9 hours | 2.0 hours |
| HotPotQA | 8 | 7.9 hours | 27.5 hours |

21

Table 14: Detail performance of our methods and other baselines on six test product groups of ProductQA. X–prompt represents directly prompting the model X; agile-X–Y represents incorporating the model X within the AGILE framework, while Y represents prompting or PPO training. The Short and Long stand for the results evaluated on short answers and long answers, respectively. The seeking advice cost is $c = 0.3$. The best total scores are highlighted in bold.

| Group | | gpt3.5-prompt Short | gpt3.5-prompt Long | gpt4-prompt Short | gpt4-prompt Long | agile-vicuna-13b-prompt Short | agile-vicuna-13b-prompt Long | agile-gpt3.5-prompt Short | agile-gpt3.5-prompt Long | agile-gpt4-prompt Short | agile-gpt4-prompt Long | agile-vic7b-ppo(ours) Short | agile-vic7b-ppo(ours) Long | agile-vic13b-ppo(ours) Short | agile-vic13b-ppo(ours) Long |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Camera Cases | Advice Rate↓ | - | - | - | - | 0.182 | 0.182 | 0.313 | 0.313 | 0.175 | 0.175 | 0.199 | 0.199 | 0.263 | 0.263 |
| | Accuracy↑ | 0.200 | 0.320 | 0.385 | 0.495 | 0.182 | 0.330 | 0.537 | 0.644 | 0.775 | 0.791 | 0.818 | 0.776 | 0.860 | 0.841 |
| | Total Score↑ | - | - | - | - | 0.127 | 0.275 | 0.443 | 0.550 | 0.722 | 0.738 | 0.758 | 0.716 | **0.781** | **0.762** |
| Leggings | Advice Rate↓ | - | - | - | - | 0.154 | 0.154 | 0.359 | 0.359 | 0.200 | 0.200 | 0.201 | 0.201 | 0.251 | 0.251 |
| | Accuracy↑ | 0.181 | 0.306 | 0.503 | 0.594 | 0.154 | 0.267 | 0.497 | 0.646 | 0.766 | 0.790 | 0.837 | 0.834 | 0.876 | 0.885 |
| | Total Score↑ | - | - | - | - | 0.108 | 0.221 | 0.389 | 0.538 | 0.706 | 0.730 | 0.777 | 0.774 | **0.801** | **0.810** |
| All Pans | Advice Rate↓ | - | - | - | - | 0.167 | 0.167 | 0.336 | 0.336 | 0.220 | 0.220 | 0.184 | 0.184 | 0.220 | 0.220 |
| | Accuracy↑ | 0.201 | 0.297 | 0.470 | 0.538 | 0.167 | 0.272 | 0.506 | 0.605 | 0.784 | 0.804 | 0.843 | 0.831 | 0.866 | 0.869 |
| | Total Score↑ | - | - | - | - | 0.117 | 0.222 | 0.405 | 0.504 | 0.718 | 0.738 | 0.788 | 0.776 | **0.800** | **0.803** |
| Rollerball Pens | Advice Rate↓ | - | - | - | - | 0.130 | 0.130 | 0.333 | 0.333 | 0.231 | 0.231 | 0.162 | 0.162 | 0.212 | 0.212 |
| | Accuracy↑ | 0.193 | 0.271 | 0.449 | 0.573 | 0.130 | 0.242 | 0.482 | 0.627 | 0.767 | 0.808 | 0.776 | 0.769 | 0.816 | 0.824 |
| | Total Score↑ | - | - | - | - | 0.091 | 0.203 | 0.382 | 0.527 | 0.698 | 0.739 | 0.727 | 0.720 | **0.752** | **0.760** |
| Mother-boards | Advice Rate↓ | - | - | - | - | 0.214 | 0.214 | 0.303 | 0.303 | 0.225 | 0.225 | 0.162 | 0.162 | 0.235 | 0.235 |
| | Accuracy↑ | 0.253 | 0.431 | 0.511 | 0.637 | 0.215 | 0.337 | 0.525 | 0.686 | 0.815 | 0.855 | 0.835 | 0.831 | 0.877 | 0.882 |
| | Total Score↑ | - | - | - | - | 0.151 | 0.273 | 0.434 | 0.595 | 0.747 | 0.788 | 0.786 | 0.782 | **0.806** | **0.812** |
| Rifle Scopes | Advice Rate↓ | - | - | - | - | 0.197 | 0.197 | 0.293 | 0.293 | 0.198 | 0.198 | 0.167 | 0.167 | 0.216 | 0.216 |
| | Accuracy↑ | 0.187 | 0.306 | 0.463 | 0.587 | 0.197 | 0.313 | 0.502 | 0.657 | 0.770 | 0.806 | 0.802 | 0.760 | 0.828 | 0.822 |
| | Total Score↑ | - | - | - | - | 0.138 | 0.254 | 0.414 | 0.569 | 0.711 | 0.747 | 0.752 | 0.710 | **0.763** | **0.757** |
| Average | Advice Rate↓ | - | - | - | - | 0.174 | 0.174 | 0.323 | 0.323 | 0.208 | 0.208 | 0.179 | 0.179 | 0.233 | 0.233 |
| | Accuracy↑ | 0.202 | 0.322 | 0.464 | 0.571 | 0.174 | 0.294 | 0.508 | 0.644 | 0.780 | 0.809 | 0.818 | 0.800 | 0.854 | 0.854 |
| | Total Score↑ | - | - | - | - | 0.122 | 0.242 | 0.411 | 0.547 | 0.718 | 0.747 | 0.764 | 0.746 | **0.784** | **0.784** |

22

Table 15: Ablation study on ProductQA test tasks. **w/o Reflection** represents removing the reflection function. **w/o Memory** represents prohibiting memory component. **w/o Advice** represents removing the seeking advice function. **Non-adapt advice** represents seeking advice in the same number with `agile-vic13b-ppo` at the beginning of trajectory. **w/o Tool-Use** represents removing the search product function. **w/o RL** represents the `agile-vic13b-sft`. The best scores are highlighted in bold.

| Group | | w/o Reflection | w/o Memory | w/o Advice | Non-adapt Advice | w/o Tool-Use | w/o RL | agile-vic-13b-ppo |
|---|---|---|---|---|---|---|---|---|
| Camera Cases | Advice Rate ↓ | 0.335 | 0.459 | 0.000 | 0.263 | 0.452 | 0.295 | 0.263 |
| | Accuracy ↑ | 0.851 | 0.869 | 0.735 | 0.827 | 0.870 | 0.849 | 0.860 |
| | Total Score ↑ | 0.750(-4.1%) | 0.731(-6.8%) | 0.735(-6.3%) | 0.748(-4.4%) | 0.734(-6.4%) | 0.760(-2.8%) | **0.781** |
| Leggings | Advice Rate ↓ | 0.276 | 0.437 | 0.000 | 0.251 | 0.529 | 0.290 | 0.251 |
| | Accuracy ↑ | 0.874 | 0.902 | 0.762 | 0.828 | 0.880 | 0.867 | 0.876 |
| | Total Score ↑ | 0.791(-1.3%) | 0.771(-3.9%) | 0.762(-5.1%) | 0.753(-6.4%) | 0.721(-11.1%) | 0.780(-2.7%) | **0.801** |
| All Pans | Advice Rate ↓ | 0.263 | 0.413 | 0.000 | 0.220 | 0.550 | 0.225 | 0.220 |
| | Accuracy ↑ | 0.867 | 0.900 | 0.759 | 0.818 | 0.877 | 0.855 | 0.866 |
| | Total Score ↑ | 0.788(-1.5%) | 0.776(-3.1%) | 0.759(-5.4%) | 0.752(-6.4%) | 0.712(-12.4%) | 0.788(-1.5%) | **0.800** |
| Rollerball Pens | Advice Rate ↓ | 0.237 | 0.378 | 0.000 | 0.212 | 0.501 | 0.220 | 0.212 |
| | Accuracy ↑ | 0.818 | 0.843 | 0.727 | 0.785 | 0.868 | 0.812 | 0.816 |
| | Total Score ↑ | 0.747(-0.7%) | 0.730(-3.0%) | 0.727(-3.4%) | 0.721(-4.3%) | 0.718(-4.7%) | 0.746(-0.8%) | **0.752** |
| Motherboards | Advice Rate ↓ | 0.270 | 0.368 | 0.000 | 0.235 | 0.483 | 0.285 | 0.235 |
| | Accuracy ↑ | 0.878 | 0.886 | 0.766 | 0.829 | 0.873 | 0.871 | 0.877 |
| | Total Score ↑ | 0.797(-1.1%) | 0.776(-3.9%) | 0.766(-5.2%) | 0.758(-6.3%) | 0.728(-10.7%) | 0.786(-2.5%) | **0.806** |
| Rifle Scopes | Advice Rate ↓ | 0.237 | 0.385 | 0.000 | 0.216 | 0.440 | 0.221 | 0.216 |
| | Accuracy ↑ | 0.824 | 0.858 | 0.733 | 0.783 | 0.824 | 0.805 | 0.828 |
| | Total Score ↑ | 0.753(-1.3%) | 0.742(-2.8%) | 0.733(-4.1%) | 0.718(-6.3%) | 0.692(-10.3%) | 0.739(-3.2%) | **0.763** |
| Average | Advice Rate ↓ | 0.270 | 0.407 | 0.000 | 0.233 | 0.492 | 0.256 | 0.233 |
| | Accuracy ↑ | 0.852 | 0.876 | 0.747 | 0.812 | 0.865 | 0.843 | 0.854 |
| | Total Score ↑ | 0.771(-1.7%) | 0.754(-4.0%) | 0.747(-5.0%) | 0.742(-5.7%) | 0.717(-9.3%) | 0.766(-2.3%) | **0.784** |

Table 16: Performance of the model (`agile-vic13b-ppo`) trained on different seeking advice cost settings.

| Group | | Seeking Advice Cost | | | | |
|---|---|---|---|---|---|---|
| | | 0.5 | 0.4 | 0.3 | 0.2 | 0.1 |
| Camera Cases | Advice Rate | 0.108 | 0.189 | 0.263 | 0.339 | 0.458 |
| | Accuracy | 0.806 | 0.829 | 0.860 | 0.885 | 0.929 |
| Leggings | Advice Rate | 0.098 | 0.188 | 0.251 | 0.317 | 0.464 |
| | Accuracy | 0.824 | 0.844 | 0.876 | 0.877 | 0.921 |
| All Pans | Advice Rate | 0.094 | 0.163 | 0.220 | 0.262 | 0.384 |
| | Accuracy | 0.813 | 0.845 | 0.866 | 0.889 | 0.926 |
| Rollerball Pens | Advice Rate | 0.100 | 0.163 | 0.212 | 0.264 | 0.406 |
| | Accuracy | 0.780 | 0.799 | 0.816 | 0.829 | 0.891 |
| Motherboards | Advice Rate | 0.103 | 0.162 | 0.235 | 0.307 | 0.443 |
| | Accuracy | 0.825 | 0.839 | 0.877 | 0.901 | 0.941 |
| Rifle Scopes | Advice Rate | 0.087 | 0.144 | 0.216 | 0.257 | 0.385 |
| | Accuracy | 0.780 | 0.797 | 0.828 | 0.845 | 0.897 |
| Average | Advice Rate | 0.098 | 0.168 | 0.233 | 0.291 | 0.423 |
| | Accuracy | 0.805 | 0.825 | 0.854 | 0.871 | 0.918 |

23

# E  Case study

Case #1, illustrated in Table 17, provides a specific example demonstrating how `agile-vic13b-ppo` proactively seeks advice from a human expert for questions it cannot answer. Furthermore, it leverages reflection to extract general knowledge from the expert's responses, which can then be applied in future QA sessions.

Case #2, shown in Table 18, demonstrates how `agile-vic13b-ppo` utilizes tools to address product recommendation questions.

24

Table 17: Case #1 from `agile-vic13b-ppo` on the test task, camera cases group, in ProductQA. ✓ represents correct responses. Our agent proactively seeks advice from human experts at session 313 and summarizes a piece of general knowledge by reflection. In the following sessions, the agent leverages this knowledge to successfully answer three questions for different products at sessions 721, 1265, and 1364, respectively. It demonstrates the effectiveness of seeking advice and reflection.

| Session | Context | Action | Reflection / Answer |
|---|---|---|---|
| 313 | Answer users' [Question] about product B00PV0IEDY based on the following three types of information:<br>1. [Product Metadata] ... 'Carrying Options': 'backpack straps' ...<br>2. [Memory] Existing knowledge in your memory about camera_cases. camera cases with no tripod mount are not suitable for scenarios where a stable camera setup is crucial, such as long-exposure photography or macro work where camera shake can significantly impact image quality. 3.[History] ...<br>...<br>**Question**: Is the camera case designed for prolonged travel and capable of withstanding heavy loads? | Seek Advice | **Human Expert:** Certainly, the backpack straps on the camera case are designed for comfort even when carrying heavier loads. These straps enable ergonomic weight distribution, making the case ideal for prolonged travel. ✓<br>**Reflection:** *Camera cases with backpack straps feature ergonomic designs that help distribute the weight of the camera equipment across the user's back, reducing strain and increasing comfort during transport.* |
| 721 | Answer users' [Question] about product B000WH86BQ ...<br>1. [Product Metadata] ... 'Carrying Options': 'backpack straps' ...<br>2. [Memory] Existing knowledge in your memory about camera_cases. *Camera cases with backpack straps feature ergonomic designs that help distribute the weight of the camera equipment across the user's back ...* 3.[History] ...<br>**Question**: Is the camera case suitable for extended journeys and potentially heavy loads? | Predict Answer | **Answer:** The case is designed to be adjustable and has padded straps which makes it suitable for long-term use with heavy equipment. ✓ |
| 1265 | Answer users' [Question] about product B005J09OXU ...<br>1. [Product Metadata] ... 'Carrying Options': 'backpack straps' ...<br>2. [Memory] Existing knowledge in your memory about camera_cases. *Camera cases with backpack straps feature ergonomic designs that help distribute the weight of the camera equipment across the user's back ...* 3.[History] ...<br>**Question**: Will this camera case enhance the comfort of transporting my gear on treks? | Predict Answer | **Answer:** Certainly, a camera case with backpack straps is engineered to evenly distribute the weight across your shoulders, improving ergonomics and comfort. This is particularly beneficial for reducing fatigue and strain during prolonged activities like treks. ✓ |
| 1364 | Answer users' [Question] about product B000YA33DC ...<br>1. [Product Metadata] ... 'Carrying Options': 'backpack straps' ...<br>2. [Memory] Existing knowledge in your memory about camera_cases. *Camera cases with backpack straps feature ergonomic designs that help distribute the weight of the camera equipment across the user's back ...* 3.[History] ...<br>**Question**: Does this camera case facilitate easier transportation when I frequently travel? | Predict Answer | **Answer:** Certainly, the backpack straps on this camera case not only simplify the process of carrying but also evenly distribute the weight, making it an ideal option for those who are frequently on the move and prefer a more comfortable and balanced carrying solution. ✓ |

25

Table 18: Case #2 from `agile-vic13b-ppo` on the ProductQA test set. ✓ represents correct responses. By leveraging SQL searching tools, agent successfully find the correct product to recommend to users.

| Session | Context | Action | Output |
|---|---|---|---|
| 918 | Answer users' [Question] about product B00FUHWE5M based on the following three types of information:<br>1. [**Product Metadata**] provides metadata about product B00FUHWE5M from the leggings database. The schema of the leggings database is outlined below (in the format field[unit](value1, value2, ...)).<br><br>price[dollar]<br>*brand*(Jescakoo, QZUnique, ...)<br>warmth(hot,moderate,cool) ...<br>2. [**Memory**] ...<br>3. [**History**] ...<br><br>...<br>**Question:** I've heard good things about <u>QZUnique</u>. Spit out the first leggings you have from them. ... | Search Product | **SQL:** SELECT product_id FROM "leggings" WHERE *brand* = "<u>QZUnique</u>" LIMIT 1;<br><br>**Answer:** The product that meets your requirements is as follows. 'QZUnique Women's Classic Retro Printed Casual Pattern Ankle Length Elastic Tights Leggings' with asin: B012IKGF62 ✓ |
| 1298 | Answer users' [Question] about product B012AQGKXC based on the following three types of information:<br>1. [**Product Metadata**] provides metadata about product B012AQGKXC from the motherboards database. The schema of the motherboards database is outlined below (in the format field[unit](value1, value2, ...)).<br><br>price[dollar]<br>*brand*(ASRock, Gigabyte, ...)<br>*memory_support*[GB] ...<br>2. [**Memory**] ...<br>3. [**History**] ...<br><br>...<br>**Question:** I'm looking for an <u>ASRock</u> motherboard with at least <u>32 GB</u> of memory support. Don't keep me waiting, hustle up and find it for me. ... | Search Product | **SQL:** SELECT product_id FROM "motherboards" WHERE <u>brand</u> = "<u>ASRock</u>" AND *memory_support* >= <u>32</u> LIMIT 1;<br><br>**Answer:** The product that would suit your preferences is: 'AS Rock LGA1155 DDR3 SATA3 USB3.0 Quad CrossFireX and Quad SLI A E ATX Motherboard Z77 EXTREME4' with asin: B007KTY4A6 ✓ |

26

# F Development of the ProductQA dataset

## F.1 Product collection

The product groups and the corresponding products are collected by the following steps.

1. Filter the Amazon Review Data to retain only products with at least 100 reviews, then cluster them by category tags.

2. Sort the clusters by size, from largest to smallest. Manually review each cluster in order: we keep product clusters that involve diverse technical details and long-tail domain knowledge, such as electronics, from which we can potentially construct a diverse set of user questions. The manual review ends when we have collected 26 clusters. Each cluster is referred to as a *product group*.

3. For each product group, we remove the top 10% of products with the highest number of reviews. We exclude these most popular products from the datasets to prevent data leakage, as information about them is likely included in the pre-training set of LLMs. From the remaining items, we randomly select up to 20 products to form the final product set.

## F.2 Annotation guidelines

There are two annotation tasks, product table creation and QA collection. We provide the annotation guidelines in this Section.

**Task 1: Product table creation**    For each product group, we provide a series of features and their corresponding values for each product in the group. This information is obtained by prompting GPT-4 to extract data from the reviews of each product. The task of annotators is to construct a product table containing only the metadata. Please follow these steps:

1. Select up to 15 common features relevant to the product group. These features must include product ID, product title, brand, and price. Choose additional features based on their commonality and necessity within the product group.

2. For each product in the product group, verify the feature values for each selected feature.

Finally, the product tables are reviewed and refined by the authors.

**Task 2: QA collection**    Annotators are required to fill out a table as shown in Table 19. Each row contains a triplet consisting of a *question*, a *long answer*, and a *short answer*, all generated by GPT-4. Annotators should fill the following columns: *Is question reasonable*, *Is long answer correct*, *Refined long answer*, *Is short answer correct* and *Refined short answer*. Please follow these steps:

1. **Evaluate the question**: Verify if the *question* resembles a typical query found in real-world product conversations in online shopping. Select 'yes' or 'no' in the *Is question reasonable* column. Any question containing harmful information is considered unreasonable and should be labeled as 'no'. If 'no' is selected, the row will be dropped, and you do not need to proceed with the subsequent steps for that row.

2. **Assess the long answer**: Check if the *long answer* correctly responds to the *question*. Select 'yes', 'no' or 'I do not know' in the *Is long answer correct* column. Consider the following special cases:

   • If the long answer is ambiguous (e.g., 'The product is designed to be waterproof, while some users do not think so.'), mark it as incorrect.

   • For numerical questions, an answer is considered correct if it fits the real-world scenario and the conclusion is clear. Specific values or ranges (e.g., 5cm, 5cm-10cm, several months) are acceptable if they correspond to the real-world scenario.

   • If the long answer contains a specific piece of knowledge, verify its accuracy.

   • If the long answer is incorrect or does not address the question, and you do not know the correct answer (even after checking the product information table, looking up the product URL, and searching online), select 'I do not know'.

27

- Any long answer containing harmful information should be labeled as 'I do not know'.

If you select 'I do not know', the row will be dropped, and you do not need to perform the subsequent steps for that row.

3. **Refine the long answer**: If you select 'no' in step 2, provide a correct long answer in the *Refined long answer* column.

4. **Assess the short answer**: Determine whether the *short answer* is correct. A short answer must be 'yes', 'no', or an entity. Choose 'yes' or 'no' in the *Is short answer correct* column. Consider the following special cases:

   - If the question is a choice and the short answer is 'yes' or 'no', it is incorrect.
   - If the question pertains to degrees (e.g. 'How durable ... ?') and the short answer is 'yes' or 'no', it is incorrect.
   - If the short answer does not align with the long answer, it is incorrect.

5. **Refine the short answer**: If you select 'no' in step 4, provide a correct short answer in the *Refined short answer* column.

The authors will review the annotation in batches. Specifically, 5% of each batch will be checked. If the accuracy rate of the checked annotation is below 98%, the entire batch will be relabeled.

Table 19: An example of the ProductQA annotation table.

| Question | Is question reasonable | Long answer | Is long answer correct | Refined long answer | Short answer | Is short answer correct | Refined short answer |
|---|---|---|---|---|---|---|---|
| What is the size of the neodymium driver used in the JVC HA-EB75 headphones? | [To fill] | The JVC HA-EB75 headphones contain a 13.5 mm neodymium driver in each earpiece, which contributes to the enhanced sound quality. | [To fill] | [To fill] | 13.5 mm | [To fill] | [To fill] |

# G   Broader impact

## G.1   Positive broader impact

(1) We created ProductQA, a dataset of 88,229 QA pairs across 26 product groups. This dataset provides a comprehensive evaluation environment for LLM agents, addressing real-world challenges such as managing historical information and accumulated knowledge, using tools, interacting with humans, performing self-evaluation, conducting reflection, and adapting to new tasks. We believe that ProductQA can advance the research in LLM agents.

(2) AGILE serves as a general framework that supports a wide range of extensions. Agents within the framework can use more tools, perform complex reasoning using LLMs alone or in combination with other tools, and self-improve by accumulating experiences and knowledge. AGILE provides an architecture for creating powerful agents with the potential to achieve human-level intelligence.

(3) AGILE supports proactive seeking advice from human experts, ensuring a high level of accuracy for applications, even when dealing with challenging questions. Within this framework, we can manage the trade-off between accuracy and human cost. These features enable AGILE agents to be applied in real-world scenarios.

## G.2   Negative broader impact

In practical applications, LLM agents exhibit superior capabilities compared to standalone LLMs. Our research validates that the AGILE framework is a highly effective approach for optimizing LLM agents. However, this improvement also increases the potential risks of harmful applications. Therefore, it is crucial to intensify research on the safety and responsible use of LLM agents.

# H   Limitations

(1) Due to resource constraints, our experiments primarily utilize LLMs with 7B or 13B parameters within AGILE. We expect that applying AGILE framework to larger models will result in more

28

powerful agents, especially in planning and reasoning. Expanding AGILE to larger LLMs is our future work.

(2) Our ProductQA dataset includes QA pairs from 20 product groups in the training set. Due to resource constraints, we randomly selected 6 of the 20 groups for training our AGILE agent. Despite using a subset of training data, our `agile-vic13b-ppo` shows significant improvements over GPT-4 agent in accuracy and total score. Future work could enhance the agent's capabilities by training on a larger and more diverse dataset, potentially further improving performance and effectiveness.

## I    Ethical considerations

ProductQA is constructed based on the Amazon Review Dataset. We only use the review data for each product without any user personal information, such as the identity of the reviewers.

All data in ProductQA are annotated by human annotators, as described in Appendix F.2. Any data containing harmful information is removed during the annotation process.

The annotation team has 20 annotators, each holding at least a college degree, and employed by a commercial data annotation company. We have contracted this company and paid them for the annotation work at a market price.

## J    Prompt templates

**Prompt templates for ProductQA**    Figure 6 shows the prompt template for `gpt3.5-prompt`, `gpt4-prompt`.    Figure 7 provides the prompt template for `agile-vic13b-prompt`, `agile-gpt3.5-prompt`, and `agile-gpt4-prompt`. We leave the "{knowledge} and "{history}" empty when evaluate `gpt3.5-prompt` and `gpt4-prompt`. The prompt template for reflection is shown in Figure 8.

The prompt template for long answer evaluation is shown in Figure 9.

**Prompt templates for MedMCQA**    Figure 10 provides the prompt template for `Meerkat-7b-prompt`. Figure 11 illustrates the prompt template for `agile-gpt3.5-prompt`, `agile-gpt4-prompt`. We leave the "{related_question} and "{related_knowledge}" empty when evaluate `gpt3.5-prompt` and `gpt4-prompt`. The prompt template for reflection is shown in Figure 12.

**Prompt templates for HotPotQA**    Figure 13 provides the prompt template for `ReAct-gpt4-prompt`.    Figure 14 illustrates the prompt template for `agile-gpt4-prompt`. The prompt template for answer evaluation is shown in Figure 15.

29

```
[prompt]

Answer users' [Question] about product {asin} based on the following three types
of information:

1. [Product Metadata] provides metadata about product {asin} from the
{product_category} database. The schema of the {product_category} database is
outlined below (in the format field[unit](value1, value2, ...)).
{schema}
{metadata}
In addition to the provided metadata about product {asin}, you have the option to
access the full {product_category} database by executing SQL commands.

2. [Memory] Existing knowledge in your memory about {product_category}.
{knowledge}

3. [History] Previous question-answer pairs related to product {asin}.
{history}

Evaluate whether the question can be answered based solely on the information
available.
[Question]: {question}

- If the question can be answered directly, output `[PredictAnswer]\n[Answer]: `

- If an SQL search is required, output `[SearchProduct]\n[SQL]: SELECT product_id
FROM {product_category} `
```

Figure 6: The prompt for gpt3.5-prompt and gpt4-prompt on ProductQA.

30

```
[prompt]

Answer users' [Question] about product {asin} based on the following three types
of information:

1. [Product Metadata] provides metadata about product {asin} from the
{product_category} database. The schema of the {product_category} database is
outlined below (in the format field[unit](value1, value2, ...)).
{schema}
{metadata}
In addition to the provided metadata about product {asin}, you have the option to
access the full {product_category} database by executing SQL commands.

2. [Memory] Existing knowledge in your memory about {product_category}.
{knowledge}

3. [History] Previous question–answer pairs related to product {asin}.
{history}

If `[SeekAdvice]` is selected, the correct answer will be provided, and the
current problem and summarized knowledge will be added to the [History] and
[Memory] for subsequent round questions. Current status:
Similar Memory: {similar_past_knowledge_num}
Similar History: {similar_past_question_num}
Round: {round}

Evaluate whether the question can be answered based solely on the information
available.
[Question]: {question}

– If the question can be answered directly, output `[PredictAnswer]\n[Answer]: `

– If an SQL search is required, output `[SearchProduct]\n[SQL]: SELECT product_id
FROM {product_category} `

– If the information available is insufficient, necessitating seek advice from a
human, output `[SeekAdvice]`

(If you choose `[PredictAnswer]` or `[SearchProduct]`, you will receive 1 point
for doing right and no point for doing wrong. If you choose `[SeekAdvice]`, you
will directly receive 0.7 points)
```

Figure 7: The prompt for `agile-vic13b-prompt`, `agile-gpt3.5-prompt`, and
`agile-gpt4-prompt` on ProductQA.

```
[prompt]

Analyze the question and answer that pertain to an online shopping scenario
involving products in the "{product_category}" category. Your task is to extract
any information that are generally applicable to the entire {product_category}
category. Focus should be on the broader category characteristics rather than on
details specific to a specific product.

If the question–answer pair offers no relevant general insights about the
{product_category} category, simply respond with '[no information]'. Otherwise,
directly summarize the applicable general knowledge or insights about the
{product_category} category based on the provided question–answer pair.

**Question**: {question}
**Answer**: {answer}
Output:
```

Figure 8: The prompt for reflection on ProductQA.

31

```
[Prompt for long_answer evaluation]

Based on the provided question and reference answer, please determine if the
response is correct or incorrect. Begin by articulating your rationale, and
conclude with a single word judgment: 'Yes' for correct or 'No' for incorrect.

question: {question}
reference answer: {reference}
response: {response}
```

Figure 9: The prompt for long answer evaluation on ProductQA.

```
[prompt]

Answer the multiple-choice question about medical knowledge.

[question]{question}
(A){Option A}
(B){Option B}
(C){Option C}
(D){Option D}
[Answer]
```

Figure 10: The prompt for `Meerkat-7b-prompt` on MedMCQA.

```
[prompt]

The following is a multiple-choice question about medical knowledge and some
related questions and knowledge references.

[Related Question] {related_question}
[Releted Knowledge] {related_knowledge}

[Question] {question}

Solve this in a step-by-step fashion, starting by summarizing the available
information.
- If the question can be answered directly, output `[PredictAnswer]` and
conclude your response with the phrase `the answer is ([option_id])`
- If the information available is insufficient, necessitating seek advice from a
human, output `[SeekAdvice]`
(If you choose `[PredictAnswer]`, you will receive 1 point for doing right and no
point for doing wrong. If you choose `[SeekAdvice]`, you will directly receive 0.6
points.)
```

Figure 11: The prompt for `agile-gpt3.5-prompt` and `agile-gpt4-prompt` on MedMCQA.

```
[prompt]

Analyze the following {medical} [Question] and [Answer]. Your task is to extract
any [Information] that is typically applicable to issues related to the medical
field. Summarize and generalize the extracted knowledge into one sentence. If the
question-answer pair offers no relevant medical insights, simply respond with '[no
information]'.

[Question]: {question}
[Answer]: {answer}
[Information]:
```

Figure 12: The prompt for reflection on MedMCQA.

32

```
[prompt]

You are an intelligent agent with the ability to search knowledge. Please answer
the following questions.

You can analyze the solution steps based on the problem and known information.
For missing information, you can use search tools by output `[Search] ([entity])`.
Please note that the [entity] should be a noun word/phrase rather than a sentence.
If there is enough information, you can answer directly by output `[PredictAnser]
([answer])`. Please note that the answer must be the span in the observation
sentences.

[Question]: What U.S Highway gives access to Zilpo Road, and is also known as
Midland Trail?

Thought1: I need to search Zilpo Road, and find the U.S Highway gives access to it.
Action: [Search] (Zilpo Road)
Observation1: Search Result — Zilpo Road
The nine mile byway starts south of Morehead, Kentucky and can be accessed by U.S.
Highway 60.

Thought2: Zilpo Road is located in the eastern Kentucky, United States. So I need
to search Kentucky and find the Highway gives access to Zilpo Road.
Action: [Search] (Kentucky)
Observation2: Search Result — Morehead, Kentucky
Morehead is a home rule-class city located along US 60 (the historic Midland Trail)
and Interstate 64 in Rowan County, Kentucky, in the United States.

Thought3: US 60 also named as Midland Trail, so the answer is US 60.
Action: [PredictAnswer] (US 60)

[Question]: {question}

Thought1:
```

Figure 13: The prompt for ReAct-gpt4-prompt on HotPotQA.

33

```
[prompt]

You are an intelligent agent with the ability to search knowledge. Please answer
the following questions.

You can analyze the solution steps based on the problem and known information.
For missing information, you can use search tools by output `[Search] ([entity])`.
If there is enough information, you can output `[PredictAnser] ([answer])` to
answer the question directly or output `[Seekadvice] ()` if you are not sure and
need to seek advice. Please note that the answer must be the span in the
observation sentences.

[Question]: What U.S Highway gives access to Zilpo Road, and is also known as
Midland Trail?

Thought1: I need to search Zilpo Road, and find the U.S Highway gives access to it.
Action: [Search] (Zilpo Road)
Observation1: Search Result — Zilpo Road (Summary version)
The nine mile byway starts south of Morehead, Kentucky and can be accessed by U.S.
Highway 60.

Thought2: Zilpo Road is located in the eastern Kentucky, United States. So I need
to search Kentucky and find the Highway gives access to Zilpo Road.
Action: [Search] (Kentucky)
Observation2: Search Result — Morehead, Kentucky (Summary version)
Morehead is a home rule—class city located along US 60 (the historic Midland Trail)
and Interstate 64 in Rowan County, Kentucky, in the United States.

Thought3: US 60 also named as Midland Trail, so the answer is US 60.
Action: [PredictAnswer] (US 60)

[Question]: {question}

Thought1:
```

Figure 14: The prompt for `agile-gpt4-prompt` on HotPotQA.

```
[prompt]

Based on the provided question and reference answer, please determine if the
response is correct or incorrect. Begin by articulating your rationale, and
conclude with a single word judgment: 'Yes' for correct or 'No' for incorrect.

question: {question}
reference answer: {reference}
response: {response}
```

Figure 15: The prompt for answer evaluation on HotPotQA.

34

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Our main claims in the abstract and introduction accurately reflect the paper's contributions and scope by clearly stating the claims, highlighting the contributions, and aligning with the experimental results.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations in Appendix H of our paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

35

Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have provided the experimental settings and implementation details in Section 4, Appendix B, and Appendix J of the paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

36

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have provided our data and code in supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We have provided the necessary experimental details in Section 4, Appendix B, and Appendix J in our paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The reason for not reporting error bars is due to the high computational cost associated with corresponding experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

37

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We have provided the resource details in Section 4 of our paper.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

Answer: [Yes]

Justification: Our research conforms to the NeurIPS Code of Ethics guidelines, including considerations for fair wages, data privacy, societal impact, potential harm, bias and fairness, and impact mitigation measures.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We have discussed both the potential positive societal impacts and negative societal impacts in Appendix G of our paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

38

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: We have described potential security risks and safeguards in Appendix G and Appendix I of our paper.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have credited the existing assets used in our paper, and mentioned the necessary information in our paper.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

39

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: We have provided the details in Section 3, Appendix D, and Appendix F of our paper.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We have provided the details in Section 3, Appendix F, Appendix I, and Appendix J of our paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [Yes]

    Justification: We have included the related details in Appendix F and Appendix I of our paper.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

40

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

41