Appendix for

"Retroformer: Retrospective Large Language Agents with Policy Gradient Optimization"

A CHALLENGES

Although LLMs are not designed to handle tool use or take actions, it has been observed (Gravitas, 2023; Nakajima, 2023; Chase, 2023) that empirically for text-rich environment, especially when the actions and states are accurately described using natural languages, LLMs work surprisingly well. However there are still plenty of challenges applying LLM-based agents. Here we list several below.

Spurious Actions LLMs are not pre-trained or designed with an action-agent application in mind. Even some restrictions are explicitly specified in the prompt, the LLM model may still generate spurious actions that are not in the action space $\mathcal{A}$.

Limited Prompt Length LLM itself is stateless. However, in applications it is preferred to empower agents with states or memories for better performance. It has been observed that LLM based agents are easy to run into infinite loops if the states are not handled nicely. Many LLM agents concatenate all the previous state descriptions and actions into the prompt so that LLM as a way to bestow "state" to the LLM. Inevitably this methodology runs into the prompt length issues. As the trajectory grows longer, the prompt runs out of spaces.

Heuristic Prompt Engineering Even though a lot of paradigms have been proposed to improve LLM agents' performance (Yao et al., 2023; Ahn et al., 2022), there is a lack of systematic methodologies for consistent model refinement. In fact, manual prompt tuning is still widely used in a lot of the application scenarios.

Prohibitive Training Most of the well-performing LLMs are too large to be fit in just one or two GPUs. It is technically challenging to optimize the LLMs directly as is done in the the classical reinforcement learning setting. In particular, OpenAI has not provided any solution for RL based finetuning. Most of the issues are caused by the fact that LLMs are not pre-trained or designed with an action-agent application in mind.

B INTUITION

Compared to the LLM-based action agents, classical RL agents, though not able to handle text-based environments as nicely in the zero shot setting, are able to keep improving based on the feedback and rewards provided by the environment. Popular RL algorithms include Policy Gradient (Sutton et al., 2000), Proximal Policy Optimization Algorithm (PPO) (Schulman et al., 2017), Trust Region Policy Optimization (TRPO) (Schulman et al., 2015), and Advantage Actor Critic methods (Mnih et al., 2016).

In this draft we are proposing a simple but powerful novel framework to tackle the challenges mentioned above. On one hand, we would like to leverage the classical RL based optimization algorithms such as policy gradient to improve the model performance. On the other hand, our framework avoids finetuning on the LLM directly. The key is, instead of training the LLM directly, we train a retrospective LM. The retrospective LM takes users' prompt, rewards and feedback from the environment as input. Its output will be prompt for the actual LLM to be consumed. RL algorithms are employed to optimize the weights in the retrospective LM model instead of directly on the LLM. In our framework the weights in the actual LLM is assumed to be fixed (untrainable), which aligns well with the application scenario when the LLM is either too large to tune or prohibited from any tuning.

Another perspective viewing our framework is, we train a retrospective LM to apply automatic prompt tuning for the LLM agents. In this case, the RL algorithms such as policy gradients are employed to optimize the prompts. Ideally the retrospective LM can help summarize the past "experience", the users' prompt, the environments' feedback into a condensed text with length limit so that it is easier for the LLM to digest. To some extent, in our setting the original LLM can be considered as part of the environment since its parameters are all fixed.

C Implementation Details

C.1 Retroformer

Model We use GPT-3 (model: text-davinci-003) as the frozen actor model. For the retrospective model, we instantiate it from LongChat (model: longchat-7b-16k), which is a LM with 16k context length by fine-tuning llama-7b on instruction-following samples from ShareGPT. In all experiments, we set the temperature of actor LM as zero, i.e., T=0 and top p =1 to isolate the randomness of LM from the effects of reflections. We acknowledge that setting a higher temperature value can encourage exploration but it can obscure the impact of the proposed approaches, making it difficult to compare against existing baselines with T=0 (Yao et al., 2023; Shinn et al., 2023).

Setup Our proposed learning framework is developed by using multiple open-source tools as follows. We use the OpenAI connectors from *langchain* to build our actor models M_a. During inference of the retrospective model, we host an API server using *FastChat* and integrates it with *langchain* agents. The tool can host longchat-7b-16k with concurrent requests to speed up RL policy rollouts. For fine-tuning the retrospective model, we develop our training pipeline with *trl*, which supports transformer reinforcement learning with PPO trainer.

We present the details of the specific prompts we used and the full agent demonstrations and examples for each environment in Appendix E.

Data Collection For HotPotQA environment, We collected 3,383 reflection samples by running the base rollout policy for 3 trials ($N = 3$) for 3,000 tasks in the training set, in which 1,084 instruction-response pairs have positive ratings. For AlfWorld, we collected 523 reflection samples and for WebShop, we collected 267 reflection samples.

Training We fine-tune the retrospective model M_r with 4-bit quantized LoRA adapters (r=1 or r=4) on the offline RL datasets with epochs=4; batch size=8; lr=1.4e-5. The number of trainable parameters is 0.53M (0.015% of llama-7b) or 2.25M. Since longchat-16k is based on Llama, we used the default llama recipes for finetuning. Specifically, we first run supervised fine-tuning trainer on the samples with positive ratings for 2 epochs and then the RLHF pipeline, including reward modeling, and RL fine-tuning with PPO, on the whole offline rating dataset using the default settings for llama-7b model. We list the key hyperparameters here:

- **Supervised Finetuning**: learning rate=1e-5, batch size=32, max steps=5,000
- **Reward Modeling**: learning rate=2.5e-5, batch size=32, max steps=20,000
- **Policy Gradient Finetuning**: learning rate=1.4e-5, max steps=20,000, output max length=128, batch size=64, gradient accumulation steps=8, ppo epochs=4

Reproducibility All experiments are done in Google Cloud Platform (GCP) GKE environment with A100 40GB GPUs. The code can be found in `https://anonymous.4open.science/r/Retroformer-F107`. We plan to open source the code repository after the review period.

Algorithm The offline PPO algorithm we used for finetuning the Retrospective component in this paper is presented below in Algorithm 1. It contains three steps: offline data collection, reward model learning, and policy gradient finetuning. We use the offline ratings data to train a reward model first, and plug in the reward model for PPO finetuning.

Algorithm 1 `Retroformer` with Policy Gradient Optimization

Initialize TEXT-DAVINCI-003 as the Retrospective model with LONGCHAT-16K. Set the maximum trials for rollouts as $N = 3$. The temperature used for sampling $t_s = 0.9$.
Step 1: Offline Data Collection. Collect multiple rollouts for each environments k ($k = 1, \cdots, K$) for the tasks in the training sets and save as D_{RL}.
for episode t = 1, . . . , N **do**
  for source domain k = 1, . . . , K **do**
    Receive trajectory $[s_{k,i,\tau}, a_{k,i,\tau}, r_{k,i,\tau}]_{\tau=1}^{T}$ and episodic returns $G_{k,i}$ for task i.
    for unsuccessful tasks j **do**
      Randomly sample a pair of reflection responses $(y_{k,j}^{(1)}, y_{k,j}^{(2)})$ with Retrospective LM temperature set to t_s, with the same instruction prompt defined in Eq. (4).
      Roll out the next episode with $y_{k,j}$, and receive the episodic returns $(G_{k,i+1}^{(1)}, G_{k,i+1}^{(2)})$.
      Compute reflection response rating by $r(x_{k,i}, y_{k,i}) \triangleq G_{k,i+1} - G_{k,i}$ in Eq. (5).
      Label the response with higher ratings as the accepted response while the lower response is labeled as the rejected response.
    end for
  end for
end for
Step 2. Reward Model Learning. Use the REWARDTRAINER in TRL to train a model for classifying accepted and rejected responses given instructions.
Step 3: Policy Gradient Finetuning. Plug-in the trained reward model and use the PPOTRAINER in TRL to finetune the Retrospective model for generating reflection responses with higher ratings.

C.2 BASELINE: SOFT-ACTOR CRITIC AGENT

Traditional reinforcement learning methods have been recognized to perform well within the same framework of interaction-feedback-learning. We include one online RL algorithm, i.e., Soft Actor-Critic (Haarnoja et al., 2018), or SAC as baseline model for comparison. Given that the three environments are text-based games, inspired by (Yuan et al., 2018), we do mean-pooling for the embeddings of the generated text outputs, such as "Search[It Takes a Family]" as the agent actions. Therefore, the action space is continuous and is of 768 dimension. We apply LoRA adapters with $r = 4$ on the agent Action model instantiated from longchat-16k, and use SAC to do the online updates, with discount factor gamma=0.99, interpolation factor polyak=0.995, learning rate=0.01, entropy regularzation alpha=0.2, and batch size=8.

C.3 REWARD FUNCTION

HotPotQA F1 reward is used in the HotPotQA environment for comparing the matching of a generated answer to a question against the ground truth answer. After removing the stopwords in both answers, we calculate the number of common tokens in two answers. Then Precision is # of common tokens divided by # of generated answer tokens and the Recall is # common tokens divided by # ground truth answer tokens. We can then compute f1 from precision and recall.

AlfWorld The binary success (1) and failure of the tasks at the end of episode is used as the reward.

WebShop In each episode, the agent receives a reward $r = \mathcal{R}(s_T, a)$ in the end at timestep T, where $a = \texttt{choose[buy]}$, y is the product chosen by the agent in the final state s_T, and Y_{att} and Y_{opt} are its corresponding attributes and options. The reward is defined as:

$$r = r_{\text{type}} \cdot \frac{|U_{\text{att}} \cap Y_{\text{att}}| + |U_{\text{opt}} \cap Y_{\text{opt}}| + \mathbf{1}[y_{\text{price}} \leq u_{\text{price}}]}{|U_{\text{att}}| + |U_{\text{opt}}| + 1} \tag{7}$$

where the type reward $r_{\text{type}} = \texttt{TextMatch}(\bar{y}, \bar{y}^*)$ is based on text matching heuristics to assign low reward when y and y^* have similar attributes and options but are obviously different types of products. For example, "butter" and "plant-based meat" differ in types but may both contain attributes "cruelty-free", "non-GMO", and an option "size: pack of 2".

D Additional Experiments

Frozen model **response**	Reinforced model **response**
I got stuck in a loop where I kept searching for the same person, Henry Miller, instead of Beatrice Sylvas Wickens. It's possible that I may have misheard or misunderstood the question. To avoid this, I should have written down the question and made sure I understood it correctly before beginning the search. Additionally, I should have tried searching for "James Henry Miller's wife" instead of just "James Henry Miller".	I think the issue might have been with the search. I was not able to find the name "Miller" in the paragraph, which is why I was not able to find James Henry Miller's wife. I should have searched for "Miller's wife" instead of "James Henry Miller's wife" to avoid this confusion. New Plan: To avoid confusion, I will search for "Miller's wife" instead of "James Henry Miller's wife" and directly look for the nationality of the wife.

Figure 7: Response refinement from the reinforced retrospective model with structured format.

Structured Reflections and Action Plans. We discover one emergent behavior of the reinforced model that it can automatically paraphrase the original responses into two separate structured sections, namely *Reflection* section and *New plan:* section, although not being explicitly trained or prompted for. One such example is shown in Fig. 7. The paraphrased response retrospects in the first paragraph and provides actionable insights next, while the response from the frozen LM interleaved both parts in one paragraph, making it hard to comprehend. We can also observer from Fig. 5 that the reinforced response removes the messy, irrelevant “Next trial:” content in the end for cleaner format, which may very likely result from LLM hallucination.

E Full Examples

E.1 Actor Prompt Engineering

An example of the HotPotQA actor language model prompt is shown below.

```
Solve a question answering task with interleaving Thought, Action, Observation steps. Thought can
reason about the current situation, and Action can be three types:

(1) Search[entity], which searches the exact entity on Wikipedia and returns the first paragraph
if it exists. If not, it will return some similar entities to search.
(2) Lookup[keyword], which returns the next sentence containing the keyword in the last passage
successfully found by Search.
(3) Finish[answer], which returns the answer and finishes the task.
You may take as many steps as necessary.
You have attempted to answer the following question before and failed. The following
reflection(s) give a plan to avoid failing to answer the question in the same way you did
previously. Use them to improve your strategy of correctly answering the given question.

Reflections:
- I should have searched for the title of the book instead of the author. I also didn't check for
the proverb which was not the question.
- I should have searched for "It Takes a Family by Rick Santorum" instead of just "It Takes a
Family" to get the correct answer.
- My search was not concise and I did not specify what I was looking for. I also did not check
similar results. In this case, I should have searched for 'It Takes a Family by Rick Santorum' to
find the correct answer.

Question: It Takes a Family is a response to this 1996 book that was published by who
Thought 1: I need to search for the title of the book, "It Takes a Family", and find who
published it.
Action 1: Search["It Takes a Family"]
Observation 1: Richard John Santorum ( san-TOR-əm; born May 10, 1958) is an American politician,
attorney, author, and political commentator who represented Pennsylvania in the United States
Senate from 1995 to 2007. He was the Senate's third-ranking Republican during the final six years
of his tenure. He also ran unsuccessfully for President of the United States in the 2012
Republican primaries, finishing second to Mitt Romney.Santorum was elected to the United States
Senate from Pennsylvania in 1994. He served two terms until losing his 2006 reelection bid. A
Roman Catholic, Santorum is a social conservative who opposes abortion and same-sex marriage and
embraced a cultural warrior image during his Senate tenure. While serving as a senator, Santorum
authored the Santorum Amendment, which would have promoted the teaching of intelligent design. He
was a leading sponsor of the 2003 federal law known as the Partial-Birth Abortion Ban Act.In the
years following his departure from the Senate, Santorum has worked as a consultant, private
practice lawyer, and news contributor. He ran for the Republican nomination in the 2012 U.S.
presidential election. Before suspending his campaign on April 10, 2012, Santorum exceeded
expectations by winning 11 primaries and caucuses and receiving nearly four million votes, making
him the runner-up to eventual nominee Mitt Romney. Santorum ran for president again in 2016, but
ended his campaign in February 2016 after a poor showing in the Iowa caucuses. In January 2017,
he became a CNN senior political commentator. However, he was terminated from his contract with
CNN in May 2021 due to comments he made about Native Americans a few weeks prior which were
deemed "dismissive".
Thought 2: The paragraph mentions Rick Santorum, so I need to search "It Takes a Family by Rick
Santorum".
Action 2: Search["It Takes a Family by Rick Santorum"]
Observation 2: It Takes a Family is a 2005 book by then Pennsylvania Senator Rick Santorum. The
title is a response to the 1996 book It Takes a Village by then-First Lady Hillary Clinton. In
the book, Santorum states that the family structure is necessary. He argues that liberal social
policies have devastated the family structure and that prevailing socially liberal attitudes have
systematically devalued families across the board. He argues that the government should take a
proactive role in promoting strong families.
Thought 3: It Takes a Family was published by Rick Santorum, so the answer is Rick Santorum.
Action 3: Finish[Rick Santorum]
Observation 3: Answer is INCORRECT
```

An example of the AlfWorld actor language model prompt is shown below.

```
You are in the middle of a room. Looking quickly around you, you see a cabinet 10, a cabinet 9, a
cabinet 8, a cabinet 7, a cabinet 6, a cabinet 5, a cabinet 4, a cabinet 3, a cabinet 2, a
cabinet 1, a coffeemachine 1, a countertop 2, a countertop 1, a diningtable 2, a diningtable 1, a
drawer 3, a drawer 2, a drawer 1, a fridge 1, a garbagecan 1, a microwave 1, a sinkbasin 1, a
stoveburner 4, a stoveburner 3, a stoveburner 2, a stoveburner 1, and a toaster 1.

[Few-Shot Demonstrations]

Your memory for the task below:
Trial 0: I will try to find a different task to complete. I will look for other objects in the
room and try to complete tasks related to those objects. For example, I could try to find a book
to read or a glass to drink from. I will also try to use the objects I have found in the room in
different ways to see if that helps me complete a task.


Your task is to: heat some mug and put it in coffeemachine.
> think: To solve the task, I need to find and take a mug, then heat it with stoveburner, then
put it in coffeemachine.
Ok.
> look
You are in the middle of a room. Looking quickly around you, you see nothing.
> think: First I need to find a mug. A mug is more likely to appear in countertop (1-2),
diningtable (1-2), drawer (1-3), fridge (1), sinkbasin (1), stoveburner (1-4), toaster (1). I can
check one by one, starting with countertop 1.
Ok.
> look
You are in the middle of a room. Looking quickly around you, you see nothing.
> go to countertop 1
On the countertop 1, you see a butterknife 1, a cup 1, a fork 2, and a mug 1.
> think: Now I find a mug (1). Next, I need to take it.
Ok.
> take mug 1 from countertop 1
You pick up the mug 1 from the countertop 1.
> think: Now I take a mug (1). Next, I need to heat it with stoveburner (1-4).
Ok.
> go to stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.

STATUS: FAIL

New plan: I was stuck in a loop in which I continually examined stoveburner 1 instead of heating
mug 1 with stoveburner 1. I should have taken mug 1 from countertop 1, then heated it with
stoveburner 1, then put it in coffeemachine 1. It did not help to execute two identical actions
in a row. I will try to execute a different action if I am stuck in a loop again.
```

An example of the WebShop actor language model prompt is shown below.

```
Webshop
[Few-Shot Demonstrations]

Instruction:
i would like a 3 ounce bottle of bright citrus deodorant for sensitive skin, and price lower than
50.00 dollars
[Search]

Your memory for the task below:
Trial 0: In this attempt, I was unsuccessful. I accidentally bought a product that was more than
my budget of $50. The initial search results were not good.

Action: search[3 ounce bright citrus deodorant sensitive skin]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B078GWRC1J]
Bright Citrus Deodorant by Earth Mama | Natural and Safe for Sensitive Skin, Pregnancy and
Breastfeeding, Contains Organic Calendula 3-Ounce
$10.99
[B078GTKVXY]
Ginger Fresh Deodorant by Earth Mama | Natural and Safe for Sensitive Skin, Pregnancy and
Breastfeeding, Contains Organic Calendula 3-Ounce
$10.99
[B08KBVJ4XN]
Barrel and Oak - Aluminum-Free Deodorant, Deodorant for Men, Essential Oil-Based Scent, 24-Hour
Odor Protection, Cedar & Patchouli Blend, Gentle on Sensitive Skin (Mountain Sage, 2.7 oz,
2-Pack)
$15.95

Action: think[B078GWRC1J and B078GTKVXY are bright citrus deodorant less then 50 dollars. I can
check B078GWRC1J first.]
Observation: OK.

Action: click[B078GWRC1J]
Observation:
[Back to Search]
[< Prev]
scent [assorted scents][bright citrus][calming lavender][ginger fresh][simply non-scents]
size [travel set (4-pack)][3 ounce (pack of 1)][3-ounce (2-pack)]
Bright Citrus Deodorant by Earth Mama | Natural and Safe for Sensitive Skin, Pregnancy and
Breastfeeding, Contains Organic Calendula 3-Ounce
Price: $10.99
Rating: N.A.
[Description]
[Features]
[Reviews]
[Buy Now]

Action: think[For 3 ounce bottle of bright citrus deodorant for sensitive skin, the item has
options 'bright citrus' and '3 ounce (pack of 1)' and seems good to buy.]
Observation: OK.

Action: click[bright citrus]
Observation: You have clicked bright citrus.

Action: click[3 ounce (pack of 1)]
Observation: You have clicked 3 ounce (pack of 1).

Action: click[Buy Now]
```

E.2 RETROSPECTIVE PROMPT ENGINEERING

An example of the HotPotQA retrospective instruction prompt is shown below.

You are an advanced reasoning agent that can improve based on self reflection. You will be given a previous reasoning trial in which you were given access to an Docstore API environment and a question to answer. You were unsuccessful in answering the question either because you guessed the wrong answer with Finish[<answer>], or you used up your set number of reasoning steps. In a few sentences, Diagnose a possible reason for failure and devise a new, concise, high level plan that aims to mitigate the same failure. Use complete sentences.

Here are some examples.
[Few-Shot Demonstrations]

Previous trial:
Question: The 2016 Washington State Cougars were led by the coach who previously helmed which other team?

Thought: I need to search Washington State Cougars and find the coach of the 2016 team, then find which other team the coach previously helmed.
Action: Search[Washington State Cougars]
Observation: The Washington State Cougars (known informally as the Cougs) are the athletic teams that represent Washington State University. Located in Pullman, Washington, WSU is a member of the Pac-12 Conference in NCAA Division I. The athletic program comprises ten women's sports and six men's intercollegiate sports, and also offers various intramural sports.

Thought: The paragraph does not mention the coach of the 2016 team. I need to look up "coach" to find the coach of the 2016 team.
Action: Lookup[coach]
Observation: (Result 2/5) During the 2001–03 seasons, the Cougar football teams were distinguished by three ten-win seasons, three top ten poll rankings, and appearances in the Sun, Rose, and Holiday Bowls. The Cougars shared the Pac-10 title in 2002. Alumnus Paul Wulff, WSU's 31st head coach, was fired in late November 2011, after compiling a 9–40 (.184) record in four seasons. Mike Leach coached the team from 2012 to 2019, guiding them to six bowl games. Leach left to coach Mississippi State in 2020 and was replaced by Hawaii head coach Nick Rolovich. Rolovitch was fired in October 2021 for not complying with the state's COVID-19 vaccine mandate. Defensive coordinator Jake Dickert was named the interim head coach.

Thought: The 2016 Washington State Cougars were led by Mike Leach. I need to search Mike Leach and find which other team he previously helmed.
Action: Search[Mike Leach]
Observation: Michael Charles Leach (March 9, 1961 – December 12, 2022) was an American college football coach who primarily coached at the NCAA Division I FBS level. He was a two-time national coach of the year, three-time conference coach of the year and the mastermind behind the NCAA record-setting air raid offense. He was the head coach at Texas Tech University from 2000 to 2009, where he became the winningest coach in school history. After Texas Tech, he coached at Washington State University from 2012 to 2019, where he recorded the third-most wins of any coach in school history. He then coached at Mississippi State from 2020 until his death in 2022.Leach was known for directing offenses using lots of passing to several receivers, in a spread system known as the air raid, which Leach developed with Hal Mumme when Mumme was head coach and Leach was offensive coordinator at Iowa Wesleyan, Valdosta State, and Kentucky in the 1990s. Leach's offenses with Mumme, and later as a head coach himself, have broken numerous school and NCAA records. On eighteen occasions, Leach led his unranked team to victory over a team ranked in the AP poll, which is the most in the AP era.

Thought: Mike Leach previously helmed Texas Tech University and Kentucky. So the answer is Texas Tech University and Kentucky.
Action: Finish[Texas Tech University and Kentucky]
Observation: Answer is INCORRECT

Reflection:

An example of the AlfWorld retrospective instruction prompt is shown below.

```
You will be given the history of a past experience in which you were placed in an environment and
given a task to complete. You were unsuccessful in completing the task. Do not summarize your
environment, but rather think about the strategy and path you took to attempt to complete the
task. Devise a concise, new plan of action that accounts for your mistake with reference to
specific actions that you should have taken. For example, if you tried A and B but forgot C, then
devise a plan to achieve C with environment-specific actions. You will need this later when you
are solving the same task. Give your plan after "Plan". Here are two examples:

[Few-Shot Demonstrations]

Your task is to: heat some mug and put it in coffeemachine.
> think: To solve the task, I need to find and take a mug, then heat it with stoveburner, then
put it in coffeemachine.
Ok.
> look
You are in the middle of a room. Looking quickly around you, you see nothing.
> think: First I need to find a mug. A mug is more likely to appear in countertop (1-2),
diningtable (1-2), drawer (1-3), fridge (1), sinkbasin (1), stoveburner (1-4), toaster (1). I can
check one by one, starting with countertop 1.
Ok.
> look
You are in the middle of a room. Looking quickly around you, you see nothing.
> go to countertop 1
On the countertop 1, you see a butterknife 1, a cup 1, a fork 2, and a mug 1.
> think: Now I find a mug (1). Next, I need to take it.
Ok.
> take mug 1 from countertop 1
You pick up the mug 1 from the countertop 1.
> think: Now I take a mug (1). Next, I need to heat it with stoveburner (1-4).
Ok.
> go to stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.
> examine stoveburner 1
On the stoveburner 1, you see a pan 2.

STATUS: FAIL

New plan: I was stuck in a loop in which I continually examined stoveburner 1 instead of heating
mug 1 with stoveburner 1. I should have taken mug 1 from countertop 1, then heated it with
stoveburner 1, then put it in coffeemachine 1. It did not help to execute two identical actions
in a row. I will try to execute a different action if I am stuck in a loop again.
```

An example of the WebShop retrospective instruction prompt is shown below.

```
You will be given the history of a past experience in which you were placed in an environment and given a task
to complete. You were unsuccessful in completing the task. Do not summarize your environment, but rather think
about the strategy and path you took to attempt to complete the task. Devise a concise, new plan of action that
accounts for your mistake with reference to specific actions that you should have taken. There are two examples
below.

[Few-Shot Demonstrations]

Instruction:
i am looking for dairy free and apple variety pack of chips, and price lower than 30.00 dollars
[Search]

Action: search[dairy free and apple variety pack of chips]
Observation:
[Back to Search]
Page 1 (Total results: 50)
[Next >]
[B07HRFJWP8]
Enjoy Life Foods Soft Baked Ovals, Breakfast Bars, Nut Free Bars, Soy Free, Dairy Free, Non GMO, Gluten Free,
Vegan, Variety Pack, 4 Boxes (20 Bars Total)
$100.0
[B01KMHY5PG]
Enjoy Life Soft Baked Chewy Bars, Variety Pack, Nut Free Bars, Soy Free, Dairy Free, Gluten Free, 6 Boxes (30
Total Bars)
$21.49
[B008D2X8C4]
Enjoy Life Lentil Chips Variety Pack, Dairy Free Chips, Soy Free, Nut Free, Non GMO, Vegan, Gluten Free, 24
Bags (0.8 oz)
$100.0

Action: think[The first two items are not what I want. The third item is the dairy free and apple variety pack
of chips I want.]
Observation: OK.

Action: click[B008D2X8C4]
Observation:
[Back to Search]
[< Prev]
flavor name [dill and sour cream][garlic & parmesan][light sea salt][margherita pizza][thai chili lime][variety
pack]
size [0.8 ounce (pack of 24)][4 ounce (pack of 12)]
Price: $100.0
Rating: N.A.
[Description]
[Features]
[Reviews]
[Buy Now]

Action: think[The item has options 'variety pack' and '0.8 ounce (pack of 24)' and seems good to buy.]
Observation: OK.

Action: click[variety pack]
Observation: You have clicked variety pack.

Action: click[0.8 ounce (pack of 24)]
Observation: You have clicked 0.8 ounce (pack of 24).

Action: click[Buy Now]

STATUS: FAIL

Next plan: In this attempt, I was unsuccessful. I accidentally bought a product that was $100, which is more
than my budget of $30. Either way, the initial search results were not good. Next time, I will do
search["variety pack of chips"] and then check if the results meet the dairy free and the $30 budget
constraints. I will continue to refine my searches so that I can find more products.
```