Agent Q: Combining Search, Self-Critique and Reinforcement Learning for Autonomous Web Agents

Abstract

Large Language Models (LLMs) have shown remarkable capabilities in natural language tasks requiring complex reasoning. However, they struggle with generalizing to multi-step reasoning tasks in interactive environments like web navigation. This is primarily due to their pre-training on imitation learning datasets, which do not encompass the behaviors needed for interactive decision-making. Recent works have tried to overcome this challenge by supervised-fine tuning on curated expert demonstrations in such environments, however such behavior cloning objectives, suffer from compounding errors and yield sub-optimal policies due to limited exploration data.

To overcome these challenges, we propose a new methodology that combines guided MCTS search and AI self-critique with iterative fine-tuning on agent interactions with an off-policy variant of the Direct Preference Optimization (DPO) algorithm. Our method allows LLM agents to learn effectively from aggregate datasets of both successful and unsuccessful trajectories, improving their generalization in multi-step reasoning tasks. We validate our approach in the WebShop environment, where an agent navigates a simulated shopping website. Starting with an LLM pre-trained on agentic tasks, our iterative fine-tuning demonstrates enhanced performance and success rates compared to the behavior cloning and reinforced fine-tuning baseline, showcasing the potential for improved multi-step reasoning and decision making in interactive environments. In our real world booking experiments, we boost LLaMa-3 zero-shot performance from **18.6% to 81.7%** success rate after a single day of data collection

1 Introduction

The recent successes of Large Language Models (LLMs) represent a significant leap in artificial intelligence, particularly within the domain of natural language processing. Foremost models like ChatGPT John Schulman et al. (2022), Gemini Anil et al. (2023), Opus Anthropic (2024), and LLaMA-3 Touvron et al. (2023) demonstrate capabilities that match or even surpass human performance on a number of tasks. These breakthroughs have extended the utility of LLMs from traditional text-based applications to more dynamic, agentic roles, in which they do not just generate text but can take actions autonomously in a number of environments from code to web applications (Zhang & Zhang, 2023; Hong et al., 2023; Zhou et al., 2024; Deng et al., 2023; Gur et al., 2024) among others. However, despite these advancements, a significant challenge persists: LLMs struggle to generalize effectively in interactive, multi-step environments, since they are not native trained for such applications. This limitation is primarily due to traditional training approaches that rely heavily on static imitation learning datasets, mostly consisting of web text or human-written instructions, which do not adequately equip models to navigate the dynamic real-world interactions.

A growing literature on agentic formulation seeks to address these issues, however these works mostly focus on building frameworks around prompt-based learning on existing models or limited fine-tuning on static datasets, and are thus limited in their core reasoning and decision making capabilities. In this work, we seek to design an approach that allows a web agent to improve with autonomous experience and limited supervision.

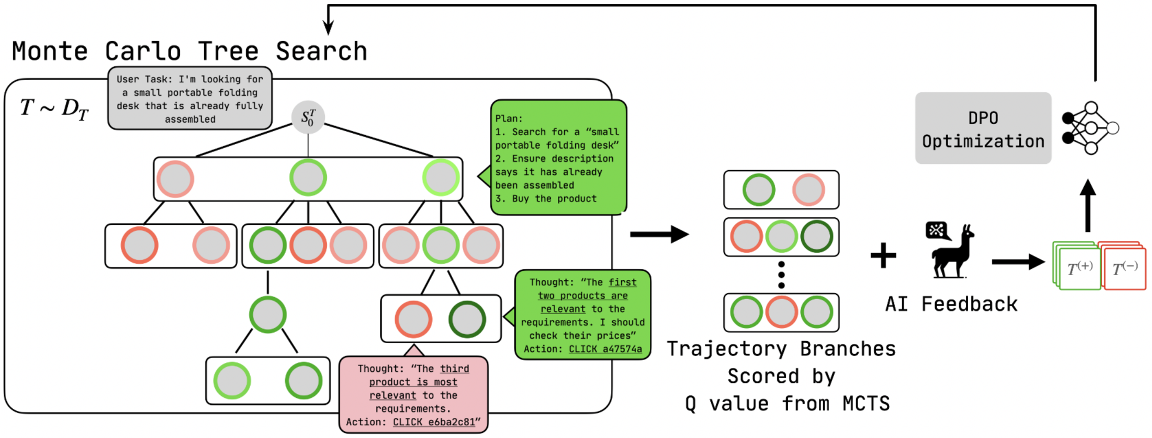

Monte Carlo Tree Search
$T \sim D_T$
User Task: I'm looking for a small portable folding desk that is already fully assembled
S_0^T
Plan:
1. Search for a "small portable folding desk"
2. Ensure description says it has already been assembled
3. Buy the product
Thought: "The first two products are relevant to the requirements. I should check their prices" Action: CLICK a47574a
Thought: "The third product is most relevant to the requirements. Action: CLICK e6ba2c81"
Trajectory Branches Scored by Q value from MCTS
AI Feedback
DPO Optimization
$T^{(+)}$
$T^{(-)}$

Figure 1: We use Monte Carlo Tree Search (MCTS) to guide trajectory collection and iteratively improve model performance using direct preference optimization (DPO). We begin on the left by sampling a user query from the list of tasks in the dataset. We iteratively expand the search tree using UCB1 as a heuristic to balance exploration and exploitation of different actions. We store the accumulated reward obtained for each node in the tree, where in this image darker green indicates higher reward and darker red indicates lower reward. To construct the preference dataset, we compute a weighted score of the MCTS average Q value and score generated by a feedback language model to construct contrastive pairs for DPO. The policy is optimized and can be iteratively improved.

Towards this goal, we introduce **Agent Q** - a novel approach that combines several key concepts - search, self-critique and reinforcement learning (RL) - to demonstrate SOTA results on web interaction tasks. We first evaluate and find that existing open-source LLMs struggle with search and navigation. To alleviate this issue we formulate an Monte Carlo Tree Search (MCTS) approach on web interfaces to autonomously generate data. In order to limit the need for human supervision we combine this data generation approach with AI-based self-critique at step level. This turns out be critical as in our experiments LLMs struggle to learn from sparse signal in long horizons. Finally we use this AI feedback pipeline to improve the model's zero-shot capabilities with Direct Preference Optimization training. This allows for a scalable self-improvement with AI feedback pipeline for autonomous web agents.

Our methodology is rigorously tested within the WebShop environment Yao et al. (2022), a simulated platform where an LLM agent is tasked with navigating a complex e-commerce site to locate and select products. Starting with a robustly pre-trained LLM on agentive tasks Zhang et al. (2024), we apply our novel training framework, which demonstrably surpasses traditional behavior cloning methods in efficacy with close to 50% relative improvement in success rates. We further evaluate our approach on a real world reservations booking website and improve the LLaMa 3-70B model zero-shot absolute success rate from **18.6%** to **81.7%** after a single day of autonomous data collection. We believe that our approach is an efficient pipeline for autonomous web agent improvement through it's search and self-critique capabilities, but a number of safety critical challenges remain before larger scale real web deployment.

2 Preliminaries

In this section we will outline the agent framework we use for our model, and our fine-tuning approaches.

2.1 Agentic LLM Formulations

Consider a general setup where an agent interacts with an environment to solve tasks over discrete time steps. At each time step t, the agent receives an observation $o_t \in O$ from the environment and selects an action $a_t \in \mathcal{A}$ based on a policy $\pi(a_t|s_t)$. Here, $s_t = (o_1, a_1, \ldots, o_{t-1}, a_{t-1}, o_t)$ is the current state of the agent, representing the context history. For example the actions of a web agent could be "CLICK [ELEMENT ID]", "SCROLL", "TYPE [CONTENT]" etcc.. However, directly mapping s_t to a_t can be difficult, particularly with complex tasks that require extensive reasoning. In this setting "zero-shot" LLM models that directly output actions from observations might struggle, similar to standard LLM reasoning applications. Following Wei et al. (2022) a number of agentic architectures have incorporated similar prompting approaches to improve planning and reasoning capabilities. The ReAct framework extends the agent's action space by incorporating "thoughts", resulting in an augmented action space (with some abuse of notation) $A = A \cup L$, where L denotes the space of language actions. A language action $a_t \in L$, also referred to as a thought or reasoning trace, does not affect the environment state, but allows the model additional compute time to produce the actual environment action. Instead of using reasoning traces at each step PlanAct Liu et al. (2023) creates a plan at the first action step and subsequently outputs direct actions, and PlanReAct is a combination, which produces both a general plan and thoughts at each step. For our simulated experiments on WebShop, we use ReAct prompting, following Zhang et al. (2024), while our real website experiments utilize a version of PlanReAact, which also includes a reactive explanation for the chosen action. For an example of a full agent trace with this format, consult appendix C.

2.2 Fine-Tuning Language Models From Feedback

Classical RLHF has used policy gradient type of algorithms, such as PPO Schulman et al. (2017), however, they tend to require online data and are quite complex to scale and unstable to train. While PPO has shown some success in prior web agent applications Nakano et al. (2021), the issues above largely make the approach not practical for general web tasks, beyond information retrieval. In this work we utilize some recent alternatives, outlined below.

2.2.1 Reinforced Fine-Tuning

Reinforced fine-tuning algorithms Zelikman et al. (2022); Gulcehre et al. (2023); Yuan et al. (2023); Singh et al. (2024) (we collectively refer to these methods as "RFT" approaches) have grown in popularity due to their simplicity and scalability. These methods iteratively sample a large number of responses form a model and filter out the sub-optimal data based on some reward model or a verifier to construct a growing dataset of high-quality examples, which is then used to train the model with standard supervised fine-tuning (SFT), which we can easily extend to the agentic multi-step setting.

2.3 Direct Preference Optimization

Direct Preference Optimization (DPO) Rafailov et al. (2023) is an alternative to the classical RLHF optimization pipeline. The original formulation considers feedback of pairwise comparisons (s, a^w, a^l), where s is a single prompt and s^w and s^l are two responses with $s^w \succ s^l$ indicating that s^w is preferred over s^l. While the algorithm was developed in a bandit setting Rafailov et al. (2024) has extended it to multi-turn settings over trajectories, using an objective of the form:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(\tau_w, \tau_l)\sim\mathcal{D}}\left[\log\sigma\left(\left(\sum_{t=0}^{i_{\tau^w}-1}\beta\log\frac{\pi_\theta(a_t^w|s_t^w)}{\pi_{\text{ref}}(a_t^w|s_t^w)}\right) - \left(\sum_{t=0}^{i_{\tau^l}-1}\beta\log\frac{\pi_\theta(a_t^l|s_t^l)}{\pi_{\text{ref}}(a_t^l|s_t^l)}\right)\right)\right] \tag{1}$$

where π_{ref} is a reference distribution (usually the base model being fine-tuned) and where a_t are again composite actions. The DPO algorithm has recently gained popularity, due to it's simplicity and performance Dubois et al. (2024); Tajwar et al. (2024). In addition, a key advantage is the ability to utilize offline and off-policy data, which is crucial for agentic applications where online learning and exploration can be potentially unsafe. In our experiments we contrast successful trajectories τ^w with unsuccessful ones τ^l for the same task (i.e. both trajectories start with the same user prompt). While this work did not explicit formulate the algorithm for agentic settings, it is easy to adapt the derivations there to the web navigation problem, since the environment is also deterministic.

3 Preliminary End-to-End Approach

In this section we first explore a preliminary approach to optimizing web agents based on end-to-end training with DPO. Our initial approach applied DPO in an online iterative fashion similar to Munos et al. (2023); Yuan et al. (2024); Pang et al. (2024), which yielded some improvement, but training was unstable and had variable performance. We attribute to a form of over-optimization Gao et al. (2023); Park et al. (2024), with concurrent works on DPO reasoning approaches making similar observations Pang et al. (2024); Hwang et al. (2024). Inspired by the Q-learning formulation of DPO Rafailov et al. (2024) we design a data accumulation with a replay buffer approach, which we found to be more stable, since the data training distribution is more stationary. Moreover, in this algorithm the reference probability π_{ref} is the data-generation distribution of the replay buffer $\mathcal{B}$ and by caching the log-likelihoods during the trajectory generation phase we can dispose of the separate reference model during training, which can be computationally quite untactful at larger scales.

For our initial experiments we use the standard WebShop environment Yao et al. (2022), where the agent needs to find particular products by browsing a simulated web shop. In this environment, we are provided with continuous rewards at the terminal state which use a combination of programmatic matching functions that consider the attributes, type, options, and price for a product. We use ReAct prompting Yao et al. (2023) with the AgentOhana xLAM-v0.1-r model Zhang et al. (2024), which is a fine-tune of a pre-trained Mixtral-8x7B-Instruct-v0.1 model Jiang et al. (2024) on a mix of agentic applications, including WebShop SFT data. We incorporate the same system prompt specified by the AgentLite Liu et al. (2024) work to ensure a fair comparison between our trained model and the xLAM base model performance.

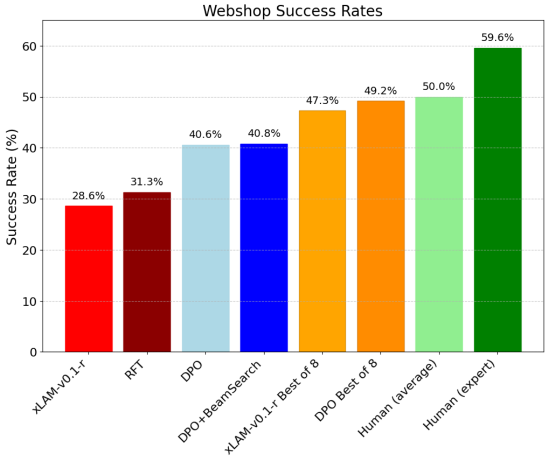


Figure 2: Succeess rate of different approaches on the WebShop Yao et al. (2022) task. All models are based on xLAM-v0.1-r Zhang et al. (2024).

The results of our experiments are shown in Fig. 2. The base model achieves a success rate of 28.6% and improves to 31.3% with STaR-based RFT training, as outlined in Section 2.2.1 and 40.6% with outcome-based trajectory DPO training as in Section 2.3. We hypothesize that the RFT approach yields limited improvement as the base xLAM model has already been trained on WebShop demonstrations data and similarly to the standard language generation problem, in the agent setting mode-covering objectives also under-perform DPO Tajwar et al. (2024). The best-of-n approach independently samples n trajectory rollouts for the same initial prompt at temperature of $t = 0.2$ (we found higher temperature sampling to under-perform) and reports the performance of the best rollout (in this case success). While end-to-end DPO yields significant improvement over the base model, it still meaningfully under-performs the best-of-8 baseline, which is close to the average human performance. We identify that one of the core failure modes of the policy at this stage is that it executes a greedy search when looking for matches to the product query. For example, for every search query, the WebShop environment yields a number of pages of results. However, we find that the model nearly always greedily searches for the best matching item in the first page of results rather than using the "[Next]" and "[Prev]" buttons to navigate between pages. Moreover using the DPO-tuned model in this setting does not yields only marginal improvement (0.2%) over the base model, despite significant difference in zero-shot performance of the two models. We can gain further inside from combining the DPO-trained model with beam-search during generation. As shown in Rafailov et al. (2024) this is equivalent to a best-first search using a DPO implicit value function when generating composite actions. This approach yields only marginal improvement over standard sampling-based approach from the DPO agent. These results reveal two related issues with the end-to-end approach for agent optimization - a simple exploration approach such as best-of-8 outperforming the DPO agent indicates that the agent is not able to learn an optimal exploration strategy. Moreover, adopting the Q-learning formulation of DPO Rafailov et al. (2024), we see that search-based optimization does improve meaningfully over the sub-optimal exploration strategy. Given that the environment is deterministic and tree-structured, **this indicates that the DPO agent is not able to learn full credit assignment from outcome supervision only**. This finding matches recent works on training LLM reasoning systems on math/code generation Xie et al. (2024); Hwang et al. (2024) with find meaningful improvement from step-level feedback and verification.

4 Combining Search and DPO

As we discovered in the previous section, while end-to-end training with DPO yields meaningful improvement, the model is not able to learn optimal search and credit assignment from sparse feedback. In this section, we aim to demonstrate further improvements to model performance by using a guided search strategy (MCTS) for data collection and AI feedback to generate **process supervision** at the step level.

4.1 Monte-Carlo Tree Search

The Monte Carlo Tree Search (MCTS) algorithm employed in this work leverages a mathematical framework to guide the iterative preference learning effectively. MCTS consists of four phases: selection, expansion, simulation, and backpropagation. Each phase plays a critical role in balancing exploration and exploitation while iteratively refining the policy.

4.1.1 Selection

The selection phase uses the Upper Confidence Bound (UCB1) formula to select nodes which aims to balance exploration and exploitation:

$$a_{t+1}^{*} = \arg\max_{a} \left[Q(s_t, a) + c_{\text{exp}} \cdot \sqrt{\frac{\log N(s_t)}{1 + N(s_{t+1})}} \right],$$

where $Q(s_t, a)$ represents the estimated value of taking action a in state s_t, $N(s_t)$ is the visitation frequency of state s_t, and c_{exp} is an exploration constant. For each rollout added to the tree, we start at the root node and follow the child states that maximize the UCB1 score until we reach a leaf node. This process is repeated for each tree/prompt in the batch.

4.1.2 Expansion

In the expansion phase, unlike traditional finite action spaces in games such as Chess or Go, web interactions have a free-form nature. To sample the space of actions, we generate K completions from the policy to expand the given state. To effectively learn from self-generated data, we require a diverse set of explored actions at each state. We encourage generation diversity using a high sampling temperature and instructions to generate creative actions in the system prompt. These modifications significantly improve the diversity of actions.

4.1.3 Simulation and Backpropagation

Once actions are expanded, the simulation phase begins; beginning from the selected state node's trace, we roll out the trajectory using the current policy π_θ until a terminal state is reached. The environment returns a reward at the end of the trajectory, R. We then backpropagate this reward multiplied by some discount factor, γ by updating the values of each node bottom up from the leaf node to the root as follows:

$$\begin{aligned} Q(s_t, a) &\leftarrow Q(s_t, a) + \gamma^{T-t} R \\ N(s_t, a) &\leftarrow N(s_t, a) + 1 \end{aligned}$$

where γ is the discount factor for future state values.

Each state node tracks two values: $Q(s_t, a)$, the total sum of rewards that were achieved by passing through this state and choosing action a and $N(s_t, a)$, the number of times it was visited during search. Hence, the expression $\frac{Q(s_t,a)}{N(s_t,a)}$ provides an approximation (that improves with more rollouts) of the value of taking this action.

4.2 Preference Pair Construction with AI Feedback

We incorporate AI-collected feedback to provide process supervision at the step level to enhance the quality of the preference pairs we will train on. We use the LLaMA-3-70B-Instruct model to produce a feedback score for each action by asking it to rank the generated actions by its perceived utility in helping the agent complete the user task. We use the following prompt:

"{System Prompt} Rather than generate an action for the instructions above, choose between the following actions. Each action has been given feedback, which is also provided. Actions and Feedback: {Generated Actions and Feedback} Select the better action. Give your answer as JSON, with the keys 'reasoning' and 'ranking'. The value of 'ranking' should be an integer corresponding to the best action. The value of 'reasoning' should be a string explaining your reasoning for the ranking. Reflect on the feedback provided for each action."

We query the feedback model for multiple iterations, each time removing the best action selected from the previous iteration from the list, until we have a full ranking of all actions. The resulting score for each action is recorded as $F(s_t)$. To construct the preference dataset for policy improvement, we evaluate each inner node in the MCTS and attempt to find contrastive pairs. Each child node receives a score based on two components: the Q value from the MCTS iteration and the LLM feedback score. The LLM feedback score is derived from the prompt mentioned above and provides an additional evaluation layer.

Using the ordering generated from the LLM prompt, we assign an $F(s_t)$ score to each child state. The total Q value at each state is then a weighted sum of the original Q value and the feedback score:

$$Q'(s_t, a) = \alpha Q(s_t, a) + (1 - \alpha) F(s_t).$$

We then compute the average Q value, normalized by visitation count for each node, and construct pairs where the positive sample branch has a reward at least greater than a threshold value $Q_{\min}$. By comparing each pair of nodes (if there are k children, we consider $\binom{k}{2}$ pairings), we identify higher reward trajectories and lower reward trajectories to form pairs. Once we have collected the preference dataset, we apply Direct Preference Optimization (DPO) over the pairings to fine-tune the final model. Our full algorithm is outlined in detail in Appendix A.

5 Extended Experiments

5.1 Tasks

For our full set of experiments, we test our method over the Webshop environment and a more complex real world environment, OpenTable.

5.1.1 OpenTable

In OpenTable, the agent is tasked with booking a restaurant reservation for a user. The agent must find a restaurant page on the OpenTable site, look for a reservation at a certain date and time, choose seating options that align with a user's preference and submit the user contact information to complete the task successfully. Since OpenTable is a live environment and is difficult to programatically measure metrics for, we use a language model, GPT-4-V to collect sub-rewards for each trajectory, based on the following metrics: Date and Time set correctly, Party Size set correctly, User information entered correctly, clicked complete reservation. Task completion is measured by the above properties being satisfied. To generate queries for the OpenTable benchmark dataset, we programatically generate a diverse set of user queries by combining the restaurant name, desired date and time, and user information.

Navigating on live websites pose a wide variety of challenges. Some examples include: (1) If the user specifies a restaurant in a different city than the location the browser is initialized in, the model will have to take extra steps to find the restaurant. (2) If the exact user requested date and time are not available, the model may have to choose the closest available reservation slot. (3) If there are preferences, such as indoor or outdoor seating options that the model is presented with, the desired behavior is to interact with the user to determine the best course of action. OpenTable presents a different much more complex set of challenges for web navigation agents; the number of steps required to complete the task is on average 13.9 steps, over double the average number of steps for Webshop, 6.8. We show that collecting data via MCTS and improving the policy via DPO we can significantly boost performance of the model.

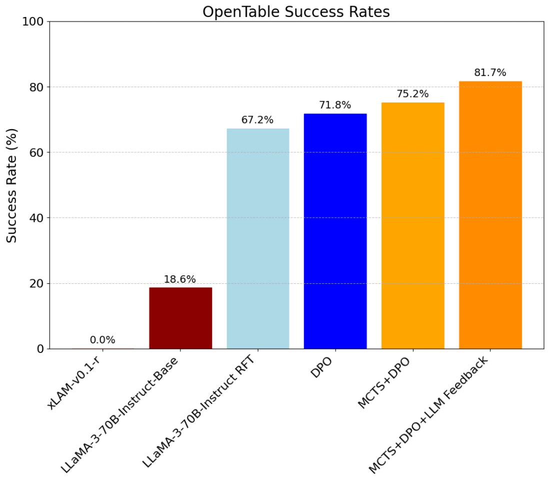


Figure 3: Success rates of different approaches on the OpenTable benchmark.

For the observation space for this environment, we design a condensed state representation that crawls the raw HTML content of a website to retrieve relevant visual components, and highlight interactive elements to the model. See appendix C for more details on the condensed representation. The agent is allowed the actions, "CLICK [ID]", "GOTO [URL]", "TYPE [ID] [TEXT]", "SUBMIT [ID]", "CLEAR [ID]", "SCROLL [UP/DOWN]", and "ASK USER HELP". For OpenTable experiments, we use the LLaMA-3-70B-Instruct model as the initial policy. We find that the superior reasoning abilities of this class of model is required for effective task completion, which is necessary to produce the diverse success and failure trajectories required to effectively improve the policy using DPO.

5.2 MCTS

In this set of experiments, we extend our iterative DPO method by collecting data via guided search (MCTS) as outlined in the prior section. Note that while our proposed algorithm is designed for iterative policy improvement, the results in this section are only from a single iteration due to time and compute constraints.

We conduct two different ablations for collecting data in the Webshop environment, where we change the number of sampled actions per node during the MCTS expansion phase. We found that MCTS guided sampling improved WebShop performance, but was not able to significantly improve success rates we achieve from best-of-n performance. Further, we found that larger branching factor rollouts slightly boosted performance. With the branching factor, k set to 3, we see a performance improvement to 42.03% success rate. When k is set to 5, we see performance up to 42.4% success rate. In Webshop, due to the model's tendency to select products in the first set of search results, we speculate that this improvement from higher branching comes from more effective search queries that the model learns to prioritize, as well as learning to inspect products in more detail before selecting them, as can be seen in Figure 1.

MCTS enables significant improvements in performance over the base policy in the OpenTable environments. Since the condensed DOM representations we designed for general websites are too large to fit multiple observations in a single context, we don't use ReaAct prompting in this setting. Rather, we provide the agent with the system prompt, condsed summary of action history, and the current observation. We conduct 3 experiments in the OpenTable environment, first using outcome supervision DPO as we did with Webshop, MCTS without LLM feedback during preference pair construction, and MCTS with LLM feedback. We find that due to the larger action space and more diverse observations, it was easy to incentivize the language model agent to produce diverse actions for every node expansion, and speculate that this enables a stronger learning signal for policy improvement. From the supervised fine-tuned model performance, MCTS and DPO without LLM feedback yields a gain from 67% success rate to 75.2%, and with LLM feedback, we gain 67% to 81.7% success rate. We speculate that the improvement from LLM feedback can be attributed from being able to precisely fix minute errors where value estimates might be sparse. For example, we found many instances where the agent intended to click a particular date/time but used the wrong element ID. The model would occasionally recover from these scenarios or obtain a partial reward from the environment, but these situations were easy for the LLM feedback agent to mark as negative samples during preference pair construction and provide a stronger learning signal. Please refer to Appendix C to see examples of the agent trajectories and failure modes.

6 Discussion

In this work we developed algorithms for autonomous improvement of web-agents with limited human supervision. While most prior works build frameworks around existing models without additional training, we specifically seek to fine-tune pre-trained models for web navigation tasks based on synthetic data. We extend the Direct Preference Optimization algorithm to multi-turn planning and reasoning interactive agents and show that we can meaningfully improve performance on a simulated web shopping environment. However with harder tasks and increased complexity of a real world reservation booking website we discover that the agent struggles with learning coherent search and credit assignment from sparse outcome feedback only, achieving only small overall improvement. To alleviate these issues we combine guided MCTS search over the web in combination with step-level AI self-critique for data generation. We then deploy a DPO feedback optimization at branch-level, which boots the performance of our real website agent by close to 15% total success rate. We believe our work opens up a number of avenues for further study. In particular, given the weekly supervised nature of our approach, we believe it is quite promising for autonomous self-improvement of general web-agents, beyond applications to a single task. While we did not explicitly pursue this direction in the current work, we believe that deploying smart search algorithms at *inference time* as alternative to zero-shot performance is another promising direction. However this approach also faces a number of limitations - doing tree-based search in dynamic and long-context environments can be complex and challenging, especially given the large scale DOM representation of web-pages and limited model context lengths. This can also be a challenge at inference time, given longer horizon tasks. Moreover, potentially deploying agents with free access to the web and personal details carries a number of security and privacy risks, which would require a robust safety infrastructure in place before deploying to safety-critical applications, such as personal communication, banking etc.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, Orhan Firat, James Molloy, Michael Isard, Paul R. Barham, Tom Hennigan, Benjamin Lee, Fabio Viola, Malcolm Reynolds, Yuanzhong Xu, Ryan Doherty, Eli Collins, Clemens Meyer, Eliza Rutherford, Erica Moreira, Kareem Ayoub, Megha Goel, George Tucker, Enrique Piqueras, Maxim Krikun, Iain Barr, Nikolay Savinov, Ivo Danihelka, Becca Roelofs, Anaïs White, Anders Andreassen, Tamara von Glehn, Lakshman Yagati, Mehran Kazemi, Lucas Gonzalez, Misha Khalman, Jakub Sygnowski, Alexandre Frechette, Charlotte Smith, Laura Culp, Lev Proleev, Yi Luan, Xi Chen, James Lottes, Nathan Schucher, Federico Lebron, Alban Rrustemi, Natalie Clay, Phil Crone, Tomas Kocisky, Jeffrey Zhao, Bartek Perz, Dian Yu, Heidi Howard, Adam Bloniarz, Jack W. Rae, Han Lu, Laurent Sifre, Marcello Maggioni, Fred Alcober, Dan Garrette, Megan Barnes, Shantanu Thakoor, Jacob Austin, Gabriel Barth-Maron, William Wong, Rishabh Joshi, Rahma Chaabouni, Deeni Fatiha, Arun Ahuja, Ruibo Liu, Yunxuan Li, Sarah Cogan, Jeremy Chen, Chao Jia, Chenjie Gu, Qiao Zhang, Jordan Grimstad, Ale Jakse Hartman, Martin Chadwick, Gaurav Singh Tomar, Xavier Garcia, Evan Senter, Emanuel Taropa, Thanumalayan Sankaranarayana Pillai, Jacob Devlin, Michael Laskin, Diego de Las Casas, Dasha Valter, Connie Tao, Lorenzo Blanco, Adrià Puigdomènech Badia, David Reitter, Mianna Chen, Jenny Brennan, Clara Rivera, Sergey Brin, Shariq Iqbal, Gabriela Surita, Jane Labanowski, Abhi Rao, Stephanie Winkler, Emilio Parisotto, Yiming Gu, Kate Olszewska, Yujing Zhang, Ravi Addanki, Antoine Miech, Annie Louis, Laurent El Shafey, Denis Teplyashin, Geoff Brown, Elliot Catt, Nithya Attaluri, Jan Balaguer, Jackie Xiang, Pidong Wang, Zoe Ashwood, Anton Briukhov, Albert Webson, Sanjay Ganapathy, Smit Sanghavi, Ajay Kannan, Ming-Wei Chang, Axel Stjerngren, Josip Djolonga, Yuting Sun, Ankur Bapna, Matthew Aitchison, Pedram Pejman, Henryk Michalewski, Tianhe Yu, Cindy Wang, Juliette Love, Junwhan Ahn, Dawn Bloxwich, Kehang Han, Peter Humphreys, Thibault Sellam, James Bradbury, Varun Godbole, Sina Samangooei, Bogdan Damoc, Alex Kaskasoli, Sébastien M. R. Arnold, Vijay Vasudevan, Shubham Agrawal, Jason Riesa, Dmitry Lepikhin, Richard Tanburn, Srivatsan Srinivasan, Hyeontaek Lim, Sarah Hodkinson, Pranav Shyam, Johan Ferret, Steven Hand, Ankush Garg, Tom Le Paine, Jian Li, Yujia Li, Minh Giang, Alexander Neitz, Zaheer Abbas, Sarah York, Machel Reid, Elizabeth Cole, Aakanksha Chowdhery, Dipanjan Das, Dominika Rogozińska, Vitaly Nikolaev, Pablo Sprechmann, Zachary Nado, Lukas Zilka, Flavien Prost, Luheng He, Marianne Monteiro, Gaurav Mishra, Chris Welty, Josh Newlan, Dawei Jia, Miltiadis Allamanis, Clara Huiyi Hu, Raoul de Liedekerke, Justin Gilmer, Carl Saroufim, Shruti Rijhwani, Shaobo Hou, Disha Shrivastava, Anirudh Baddepudi, Alex Goldin, Adnan Ozturel, Albin Cassirer, Yunhan Xu, Daniel Sohn, Devendra Sachan, Reinald Kim Amplayo, Craig Swanson, Dessie Petrova, Shashi Narayan, Arthur Guez, Siddhartha Brahma, Jessica Landon, Miteyan Patel, Ruizhe Zhao, Kevin Villela, Luyu Wang, Wenhao Jia, Matthew Rahtz, Mai Giménez, Legg Yeung, Hanzhao Lin, James Keeling, Petko Georgiev, Diana Mincu, Boxi Wu, Salem Haykal, Rachel Saputro, Kiran Vodrahalli, James Qin, Zeynep Cankara, Abhanshu Sharma, Nick Fernando, Will Hawkins, Behnam Neyshabur, Solomon Kim, Adrian Hutter, Priyanka Agrawal, Alex Castro-Ros, George van den Driessche, Tao Wang, Fan Yang, Shuo yiin Chang, Paul Komarek, Ross McIlroy, Mario Lučić, Guodong Zhang, Wael Farhan, Michael Sharman, Paul Natsev, Paul Michel, Yong Cheng, Yamini Bansal, Siyuan Qiao, Kris Cao, Siamak Shakeri, Christina Butterfield, Justin Chung, Paul Kishan Rubenstein, Shivani Agrawal, Arthur Mensch, Kedar Soparkar, Karel Lenc, Timothy Chung, Aedan Pope, Loren Maggiore, Jackie Kay, Priya Jhakra, Shibo Wang, Joshua Maynez, Mary Phuong, Taylor Tobin, Andrea Tacchetti, Maja Trebacz, Kevin Robinson, Yash Katariya, Sebastian Riedel, Paige Bailey, Kefan Xiao, Nimesh Ghelani, Lora Aroyo, Ambrose Slone, Neil Houlsby, Xuehan Xiong, Zhen Yang, Elena Gribovskaya, Jonas Adler, Mateo Wirth, Lisa Lee, Music Li, Thais Kagohara, Jay Pavagadhi, Sophie Bridgers, Anna Bortsova, Sanjay Ghemawat, Zafarali Ahmed, Tianqi Liu, Richard Powell, Vijay Bolina, Mariko Iinuma, Polina Zablotskaia, James Besley, Da-Woon Chung, Timothy Dozat, Ramona Comanescu, Xiance Si, Jeremy Greer,

Guolong Su, Martin Polacek, Raphaël Lopez Kaufman, Simon Tokumine, Hexiang Hu, Elena Buchatskaya, Yingjie Miao, Mohamed Elhawaty, Aditya Siddhant, Nenad Tomasev, Jinwei Xing, Christina Greer, Helen Miller, Shereen Ashraf, Aurko Roy, Zizhao Zhang, Ada Ma, Angelos Filos, Milos Besta, Rory Blevins, Ted Klimenko, Chih-Kuan Yeh, Soravit Changpinyo, Jiaqi Mu, Oscar Chang, Mantas Pajarskas, Carrie Muir, Vered Cohen, Charline Le Lan, Krishna Haridasan, Amit Marathe, Steven Hansen, Sholto Douglas, Rajkumar Samuel, Mingqiu Wang, Sophia Austin, Chang Lan, Jiepu Jiang, Justin Chiu, Jaime Alonso Lorenzo, Lars Lowe Sjösund, Sébastien Cevey, Zach Gleicher, Thi Avrahami, Anudhyan Boral, Hansa Srinivasan, Vittorio Selo, Rhys May, Konstantinos Aisopos, Léonard Hussenot, Livio Baldini Soares, Kate Baumli, Michael B. Chang, Adrià Recasens, Ben Caine, Alexander Pritzel, Filip Pavetic, Fabio Pardo, Anita Gergely, Justin Frye, Vinay Ramasesh, Dan Horgan, Kartikeya Badola, Nora Kassner, Subhrajit Roy, Ethan Dyer, Víctor Campos, Alex Tomala, Yunhao Tang, Dalia El Badawy, Elspeth White, Basil Mustafa, Oran Lang, Abhishek Jindal, Sharad Vikram, Zhitao Gong, Sergi Caelles, Ross Hemsley, Gregory Thornton, Fangxiaoyu Feng, Wojciech Stokowiec, Ce Zheng, Phoebe Thacker, Çağlar Ünlü, Zhishuai Zhang, Mohammad Saleh, James Svensson, Max Bileschi, Piyush Patil, Ankesh Anand, Roman Ring, Katerina Tsihlas, Arpi Vezer, Marco Selvi, Toby Shevlane, Mikel Rodriguez, Tom Kwiatkowski, Samira Daruki, Keran Rong, Allan Dafoe, Nicholas FitzGerald, Keren Gu-Lemberg, Mina Khan, Lisa Anne Hendricks, Marie Pellat, Vladimir Feinberg, James Cobon-Kerr, Tara Sainath, Maribeth Rauh, Sayed Hadi Hashemi, Richard Ives, Yana Hasson, YaGuang Li, Eric Noland, Yuan Cao, Nathan Byrd, Le Hou, Qingze Wang, Thibault Sottiaux, Michela Paganini, Jean-Baptiste Lespiau, Alexandre Moufarek, Samer Hassan, Kaushik Shivakumar, Joost van Amersfoort, Amol Mandhane, Pratik Joshi, Anirudh Goyal, Matthew Tung, Andrew Brock, Hannah Sheahan, Vedant Misra, Cheng Li, Nemanja Rakićević, Mostafa Dehghani, Fangyu Liu, Sid Mittal, Junhyuk Oh, Seb Noury, Eren Sezener, Fantine Huot, Matthew Lamm, Nicola De Cao, Charlie Chen, Gamaleldin Elsayed, Ed Chi, Mahdis Mahdieh, Ian Tenney, Nan Hua, Ivan Petrychenko, Patrick Kane, Dylan Scandinaro, Rishub Jain, Jonathan Uesato, Romina Datta, Adam Sadovsky, Oskar Bunyan, Dominik Rabiej, Shimu Wu, John Zhang, Gautam Vasudevan, Edouard Leurent, Mahmoud Alnahlawi, Ionut Georgescu, Nan Wei, Ivy Zheng, Betty Chan, Pam G Rabinovitch, Piotr Stanczyk, Ye Zhang, David Steiner, Subhajit Naskar, Michael Azzam, Matthew Johnson, Adam Paszke, Chung-Cheng Chiu, Jaume Sanchez Elias, Afroz Mohiuddin, Faizan Muhammad, Jin Miao, Andrew Lee, Nino Vieillard, Sahitya Potluri, Jane Park, Elnaz Davoodi, Jiageng Zhang, Jeff Stanway, Drew Garmon, Abhijit Karmarkar, Zhe Dong, Jong Lee, Aviral Kumar, Luowei Zhou, Jonathan Evens, William Isaac, Zhe Chen, Johnson Jia, Anselm Levskaya, Zhenkai Zhu, Chris Gorgolewski, Peter Grabowski, Yu Mao, Alberto Magni, Kaisheng Yao, Javier Snaider, Norman Casagrande, Paul Suganthan, Evan Palmer, Geoffrey Irving, Edward Loper, Manaal Faruqui, Isha Arkatkar, Nanxin Chen, Izhak Shafran, Michael Fink, Alfonso Castaño, Irene Giannoumis, Wooyeol Kim, Mikołaj Rybiński, Ashwin Sreevatsa, Jennifer Prendki, David Soergel, Adrian Goedeckemeyer, Willi Gierke, Mohsen Jafari, Meenu Gaba, Jeremy Wiesner, Diana Gage Wright, Yawen Wei, Harsha Vashisht, Yana Kulizhskaya, Jay Hoover, Maigo Le, Lu Li, Chimezie Iwuanyanwu, Lu Liu, Kevin Ramirez, Andrey Khorlin, Albert Cui, Tian LIN, Marin Georgiev, Marcus Wu, Ricardo Aguilar, Keith Pallo, Abhishek Chakladar, Alena Repina, Xihui Wu, Tom van der Weide, Priya Ponnapalli, Caroline Kaplan, Jiri Simsa, Shuangfeng Li, Olivier Dousse, Fan Yang, Jeff Piper, Nathan Ie, Minnie Lui, Rama Pasumarthi, Nathan Lintz, Anitha Vijayakumar, Lam Nguyen Thiet, Daniel Andor, Pedro Valenzuela, Cosmin Paduraru, Daiyi Peng, Katherine Lee, Shuyuan Zhang, Somer Greene, Duc Dung Nguyen, Paula Kurylowicz, Sarmishta Velury, Sebastian Krause, Cassidy Hardin, Lucas Dixon, Lili Janzer, Kiam Choo, Ziqiang Feng, Biao Zhang, Achintya Singhal, Tejasi Latkar, Mingyang Zhang, Quoc Le, Elena Allica Abellan, Dayou Du, Dan McKinnon, Natasha Antropova, Tolga Bolukbasi, Orgad Keller, David Reid, Daniel Finchelstein, Maria Abi Raad, Remi Crocker, Peter Hawkins, Robert Dadashi, Colin Gaffney, Sid Lall, Ken Franko, Egor Filonov, Anna Bulanova, Rémi Leblond, Vikas Yadav, Shirley Chung, Harry Askham, Luis C. Cobo, Kelvin Xu, Felix Fischer, Jun Xu, Christina Sorokin, Chris Alberti, Chu-Cheng Lin, Colin Evans, Hao Zhou, Alek Dimitriev, Hannah Forbes, Dylan Banarse, Zora Tung, Jeremiah Liu, Mark Omernick, Colton Bishop, Chintu Kumar, Rachel Sterneck, Ryan Foley, Rohan Jain, Swaroop Mishra, Jiawei Xia, Taylor Bos, Ge-

offrey Cideron, Ehsan Amid, Francesco Piccinno, Xingyu Wang, Praseem Banzal, Petru Gurita, Hila Noga, Premal Shah, Daniel J. Mankowitz, Alex Polozov, Nate Kushman, Victoria Krakovna, Sasha Brown, MohammadHossein Bateni, Dennis Duan, Vlad Firoiu, Meghana Thotakuri, Tom Natan, Anhad Mohananey, Matthieu Geist, Sidharth Mudgal, Sertan Girgin, Hui Li, Jiayu Ye, Ofir Roval, Reiko Tojo, Michael Kwong, James Lee-Thorp, Christopher Yew, Quan Yuan, Sumit Bagri, Danila Sinopalnikov, Sabela Ramos, John Mellor, Abhishek Sharma, Aliaksei Severyn, Jonathan Lai, Kathy Wu, Heng-Tze Cheng, David Miller, Nicolas Sonnerat, Denis Vnukov, Rory Greig, Jennifer Beattie, Emily Caveness, Libin Bai, Julian Eisenschlos, Alex Korchemniy, Tomy Tsai, Mimi Jasarevic, Weize Kong, Phuong Dao, Zeyu Zheng, Frederick Liu, Fan Yang, Rui Zhu, Mark Geller, Tian Huey Teh, Jason Sanmiya, Evgeny Gladchenko, Nejc Trdin, Andrei Sozanschi, Daniel Toyama, Evan Rosen, Sasan Tavakkol, Linting Xue, Chen Elkind, Oliver Woodman, John Carpenter, George Papamakarios, Rupert Kemp, Sushant Kafle, Tanya Grunina, Rishika Sinha, Alice Talbert, Abhimanyu Goyal, Diane Wu, Denese Owusu-Afriyie, Cosmo Du, Chloe Thornton, Jordi Pont-Tuset, Pradyumna Narayana, Jing Li, Sabaer Fatehi, John Wieting, Omar Ajmeri, Benigno Uria, Tao Zhu, Yeongil Ko, Laura Knight, Amélie Héliou, Ning Niu, Shane Gu, Chenxi Pang, Dustin Tran, Yeqing Li, Nir Levine, Ariel Stolovich, Norbert Kalb, Rebeca Santamaria-Fernandez, Sonam Goenka, Wenny Yustalim, Robin Strudel, Ali Elqursh, Balaji Lakshminarayanan, Charlie Deck, Shyam Upadhyay, Hyo Lee, Mike Dusenberry, Zonglin Li, Xuezhi Wang, Kyle Levin, Raphael Hoffmann, Dan Holtmann-Rice, Olivier Bachem, Summer Yue, Sho Arora, Eric Malmi, Daniil Mirylenka, Qijun Tan, Christy Koh, Soheil Hassas Yeganeh, Siim Põder, Steven Zheng, Francesco Pongetti, Mukarram Tariq, Yanhua Sun, Lucian Ionita, Mojtaba Seyedhosseini, Pouya Tafti, Ragha Kotikalapudi, Zhiyu Liu, Anmol Gulati, Jasmine Liu, Xinyu Ye, Bart Chrzaszcz, Lily Wang, Nikhil Sethi, Tianrun Li, Ben Brown, Shreya Singh, Wei Fan, Aaron Parisi, Joe Stanton, Chenkai Kuang, Vinod Koverkathu, Christopher A. Choquette-Choo, Yunjie Li, TJ Lu, Abe Ittycheriah, Prakash Shroff, Pei Sun, Mani Varadarajan, Sanaz Bahargam, Rob Willoughby, David Gaddy, Ishita Dasgupta, Guillaume Desjardins, Marco Cornero, Brona Robenek, Bhavishya Mittal, Ben Albrecht, Ashish Shenoy, Fedor Moiseev, Henrik Jacobsson, Alireza Ghaffarkhah, Morgane Rivière, Alanna Walton, Clément Crepy, Alicia Parrish, Yuan Liu, Zongwei Zhou, Clement Farabet, Carey Radebaugh, Praveen Srinivasan, Claudia van der Salm, Andreas Fidjeland, Salvatore Scellato, Eri Latorre-Chimoto, Hanna Klimczak-Plucińska, David Bridson, Dario de Cesare, Tom Hudson, Piermaria Mendolicchio, Lexi Walker, Alex Morris, Ivo Penchev, Matthew Mauger, Alexey Guseynov, Alison Reid, Seth Odoom, Lucia Loher, Victor Cotruta, Madhavi Yenugula, Dominik Grewe, Anastasia Petrushkina, Tom Duerig, Antonio Sanchez, Steve Yadlowsky, Amy Shen, Amir Globerson, Adam Kurzrok, Lynette Webb, Sahil Dua, Dong Li, Preethi Lahoti, Surya Bhupatiraju, Dan Hurt, Haroon Qureshi, Ananth Agarwal, Tomer Shani, Matan Eyal, Anuj Khare, Shreyas Rammohan Belle, Lei Wang, Chetan Tekur, Mihir Sanjay Kale, Jinliang Wei, Ruoxin Sang, Brennan Saeta, Tyler Liechty, Yi Sun, Yao Zhao, Stephan Lee, Pandu Nayak, Doug Fritz, Manish Reddy Vuyyuru, John Aslanides, Nidhi Vyas, Martin Wicke, Xiao Ma, Taylan Bilal, Evgenii Eltyshev, Daniel Balle, Nina Martin, Hardie Cate, James Manyika, Keyvan Amiri, Yelin Kim, Xi Xiong, Kai Kang, Florian Luisier, Nilesh Tripuraneni, David Madras, Mandy Guo, Austin Waters, Oliver Wang, Joshua Ainslie, Jason Baldridge, Han Zhang, Garima Pruthi, Jakob Bauer, Feng Yang, Riham Mansour, Jason Gelman, Yang Xu, George Polovets, Ji Liu, Honglong Cai, Warren Chen, XiangHai Sheng, Emily Xue, Sherjil Ozair, Adams Yu, Christof Angermueller, Xiaowei Li, Weiren Wang, Julia Wiesinger, Emmanouil Koukoumidis, Yuan Tian, Anand Iyer, Madhu Gurumurthy, Mark Goldenson, Parashar Shah, MK Blake, Hongkun Yu, Anthony Urbanowicz, Jennimaria Palomaki, Chrisantha Fernando, Kevin Brooks, Ken Durden, Harsh Mehta, Nikola Momchev, Elahe Rahimtoroghi, Maria Georgaki, Amit Raul, Sebastian Ruder, Morgan Redshaw, Jinhyuk Lee, Komal Jalan, Dinghua Li, Ginger Perng, Blake Hechtman, Parker Schuh, Milad Nasr, Mia Chen, Kieran Milan, Vladimir Mikulik, Trevor Strohman, Juliana Franco, Tim Green, Demis Hassabis, Koray Kavukcuoglu, Jeffrey Dean, and Oriol Vinyals. Gemini: A family of highly capable multimodal models, 2023.

Anthropic. Introducing the next generation of claude, 2024. URL `IntroducingthenextgenerationofClaude`.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. Mind2web: Towards a generalist agent for the web. In *NeurIPS Datasets and Benchmarks Track*, 2023. URL `https://openreview.net/forum?id=kiYqbO3wqw`.

Yann Dubois, Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Alpacafarm: A simulation framework for methods that learn from human feedback, 2024.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. *International Conference on machine Learning*, 2023.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. Reinforced self-training (rest) for language modeling, 2023.

Izzeddin Gur, Hiroki Furuta, Austin V Huang, Mustafa Safdari, Yutaka Matsuo, Douglas Eck, and Aleksandra Faust. A real-world webagent with planning, long context understanding, and program synthesis. In *ICLR*, 2024. URL `https://openreview.net/forum?id=9JQtrumvg8`.

Wenyi Hong, Weihan Wang, Qingsong Lv, Jiazheng Xu, Wenmeng Yu, Junhui Ji, Yan Wang, Zihan Wang, Yuxiao Dong, Ming Ding, and Jie Tang. Cogagent: A visual language model for gui agents. *ArXiv*, 2023.

Hyeonbin Hwang, Doyoung Kim, Seungone Kim, Seonghyeon Ye, and Minjoon Seo. Self-explore to avoid the pit: Improving the reasoning capabilities of language models with fine-grained rewards, 2024.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

Barret Zoph John Schulman, Christina Kim, Jacob Hilton, Jacob Menick, Jiayi Weng, Juan Felipe Ceron Uribe, Liam Fedus, Michael Pokorny Luke Metz, Rapha Gontijo Lopes, Shengjia Zhao, Arun Vijayvergiya, Eric Sigler, Adam Perelman, Chelsea Voss, Mike Heaton, Joel Parish, Dave Cummings, Rajeev Nayak, Valerie Balcom, David Schnurr, Tomer Kaftan, Chris Hallacy, Nicholas Turley, Noah Deutsch, Vik Goel, Jonathan Ward, Aris Konstantinidis, Wojciech Zaremba, Long Ouyang, Leonard Bogdonoff, Joshua Gross, David Medina, Sarah Yoo, Teddy Lee, Ryan Lowe, Dan Mossing, Joost Huizinga, Roger Jiang, Carroll Wainwright amd Diogo Almeida, Steph Lin, Marvin Zhang, Kai Xiao, Katarina Slama, Steven Bills, Alex Gray, Jan Leike, Jakub Pachocki, Phil Tillet, Shantanu Jain, Greg Brockman, Nick Ryder, Alex Paino, Qiming Yuan, Clemens Winter, Ben Wang, Mo Bavarian, Igor Babuschkin, Szymon Sidor, Ingmar Kanitscheider, Mikhail Pavlov, Matthias Plappert, Nik Tezak, Heewoo Jun, William Zhuk, Vitchyr Pong, Lukasz Kaiser, Jerry Tworek, Andrew Carr, Lilian Weng, Sandhini Agarwal, Karl Cobbe, Vineet Kosaraju, Alethea Power, Stanislas Polu, Jesse Han, Raul Puri, Shawn Jain, Benjamin Chess, Christian Gibson, Oleg Boiko, Emy Parparita, Amin Tootoonchian, Kyle Kosic, and Christopher Hesse. Introducing chatgpt, 2022. URL `https://openai.com/blog/chatgpt#OpenAI`.

Zhiwei Liu, Weiran Yao, Jianguo Zhang, Le Xue, Shelby Heinecke, Rithesh Murthy, Yihao Feng, Zeyuan Chen, Juan Carlos Niebles, Devansh Arpit, Ran Xu, Phil Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. Bolaa: Benchmarking and orchestrating llm-augmented autonomous agents, 2023.

Zhiwei Liu, Weiran Yao, Jianguo Zhang, Liangwei Yang, Zuxin Liu, Juntao Tan, Prafulla K. Choubey, Tian Lan, Jason Wu, Huan Wang, Shelby Heinecke, Caiming Xiong, and Silvio Savarese. Agentlite: A lightweight library for building and advancing task-oriented llm agent system, 2024.

Rémi Munos, Michal Valko, Daniele Calandriello, Mohammad Gheshlaghi Azar, Mark Rowland, Zhaohan Daniel Guo, Yunhao Tang, Matthieu Geist, Thomas Mesnard, Andrea Michi, Marco Selvi, Sertan Girgin, Nikola Momchev, Olivier Bachem, Daniel J. Mankowitz, Doina Precup, and Bilal Piot. Nash learning from human feedback, 2023.

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.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. Iterative reasoning preference optimization, 2024.

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. Disentangling length from quality in direct preference optimization, 2024.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.18290.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q^*: Your language model is secretly a q-function, 2024.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models, 2024.

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. 2022.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning, 2024.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents, 2022.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models, 2023.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models, 2024.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Jianguo Zhang, Tian Lan, Rithesh Murthy, Zhiwei Liu, Weiran Yao, Juntao Tan, Thai Hoang, Liangwei Yang, Yihao Feng, Zuxin Liu, et al. Agentohana: Design unified data and training pipeline for effective agent learning. *arXiv preprint arXiv:2402.15506*, 2024.

Zhuosheng Zhang and Aston Zhang. You only look at screens: Multimodal chain-of-action agents. *ArXiv*, 2023. URL https://api.semanticscholar.org/CorpusID:262053313.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In *ICLR*, 2024.

A Full MCTS Guided Direct Preference Optimization Algorithm

Algorithm 1 MCTS Guided Direct Preference Optimization

```
Input: $\pi_{\theta_0}$: initial LLM policy, $\mathcal{D}_T$: dataset of tasks the agent must complete in the environment, $N$: Number of iterations, $B$: Number of samples per iteration, $T$: MCTS tree depth, $\mathcal{B}$: replay buffer, $K$: Number of actions to sample for MCTS
Output: $\pi_{\theta_N}$, the trained LLM policy
for $i = 1$ to $N$ do
    $\pi_{\text{ref}} \leftarrow \pi_{\theta_i}$, $\pi_{\theta_i} \leftarrow \pi_{\theta_{i-1}}$
    Sample a batch of $B$ tasks from $\mathcal{D}_T$
    for each task in batch do
        Initialize the root node $s_0$
        for $t = 1$ to $T$ do
            Selection: Traverse the tree from the root node to a leaf node using tree policy (UCB1)
            Expansion: If the leaf node is not a terminal state, sample $K$ actions from policy
            Simulation: Simulate the rollout from the expanded node to obtain a value estimate
            Backpropagation: Backpropagate the value estimate bottom-up
        end for
        Collect trajectories from rollouts and store them in replay buffer $\mathcal{B}$
    end for
    Construct preference pairs $\mathcal{P} = \{(s_t, a_w^{(t)}, a_l^{(t)})\}_{t=1}^{T-1}$ where $s_t \sim \mathcal{D}_P$. For each node at step level $t$, compare each pair of child nodes, and construct the pair of generated actions $(a_w, a_l)$ if the values of taking the action, $|Q'(s_t, a_w) - Q'(s_t, a_l)| > \theta_{\text{threshold}}$
    Optimize LLM policy $\pi_{\theta_i}$ using DPO objective with $\mathcal{P}$ and $\pi_{\text{ref}}$
end for
```

B Webshop Environment Details

Here we show the Webshop system prompt which outlines the observation and action spaces of the environment.

C OpenTable Agent Interaction Examples

C.1 Example of Agent Input and Output

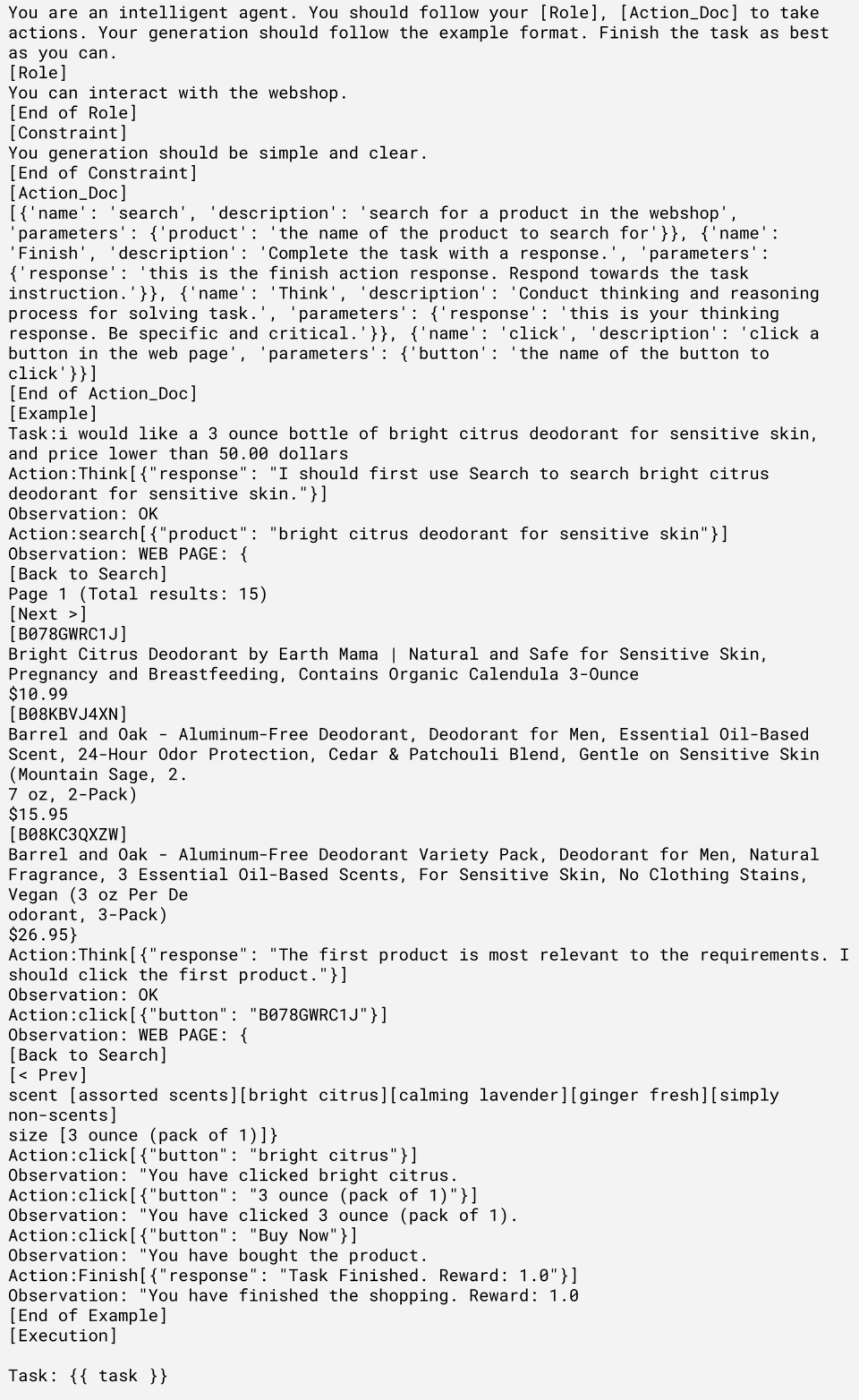
Here we demonstrate an example of the agent's input from the environment which includes a system prompt about the rules that the agent must follow, a execution history which is a condensed history actions that the agent has executed in prior steps and the current observation which is a condensed representation of the HTML DOM.

The agent output involves an optional plan, chain of thought, and the finally the list of commands that the agent must output.

C.2 Successful Task Completion

C.3 Failure Mode: Wrong Date

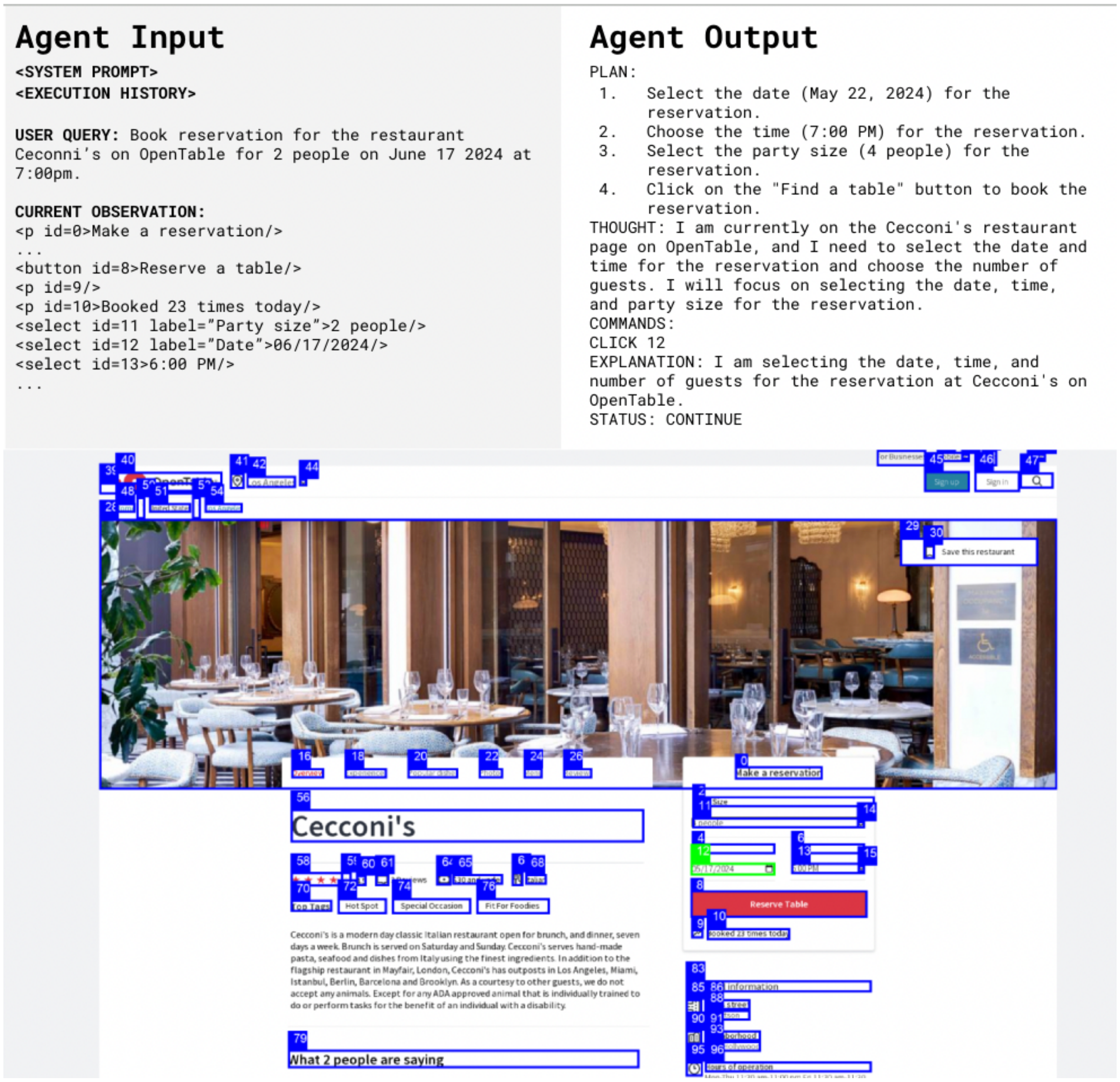
C.4 Failure Mode: Stuck in a loop



```
You are an intelligent agent. You should follow your [Role], [Action_Doc] to take
actions. Your generation should follow the example format. Finish the task as best
as you can.
[Role]
You can interact with the webshop.
[End of Role]
[Constraint]
You generation should be simple and clear.
[End of Constraint]
[Action_Doc]
[{'name': 'search', 'description': 'search for a product in the webshop',
'parameters': {'product': 'the name of the product to search for'}}, {'name':
'Finish', 'description': 'Complete the task with a response.', 'parameters':
{'response': 'this is the finish action response. Respond towards the task
instruction.'}}, {'name': 'Think', 'description': 'Conduct thinking and reasoning
process for solving task.', 'parameters': {'response': 'this is your thinking
response. Be specific and critical.'}}, {'name': 'click', 'description': 'click a
button in the web page', 'parameters': {'button': 'the name of the button to
click'}}]
[End of Action_Doc]
[Example]
Task:i would like a 3 ounce bottle of bright citrus deodorant for sensitive skin,
and price lower than 50.00 dollars
Action:Think[{"response": "I should first use Search to search bright citrus
deodorant for sensitive skin."}]
Observation: OK
Action:search[{"product": "bright citrus deodorant for sensitive skin"}]
Observation: WEB PAGE: {
[Back to Search]
Page 1 (Total results: 15)
[Next >]
[B078GWRC1J]
Bright Citrus 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
[B08KC3QXZW]
Barrel and Oak - Aluminum-Free Deodorant Variety Pack, Deodorant for Men, Natural
Fragrance, 3 Essential Oil-Based Scents, For Sensitive Skin, No Clothing Stains,
Vegan (3 oz Per De
odorant, 3-Pack)
$26.95}
Action:Think[{"response": "The first product is most relevant to the requirements. I
should click the first product."}]
Observation: OK
Action:click[{"button": "B078GWRC1J"}]
Observation: WEB PAGE: {
[Back to Search]
[< Prev]
scent [assorted scents][bright citrus][calming lavender][ginger fresh][simply
non-scents]
size [3 ounce (pack of 1)]}
Action:click[{"button": "bright citrus"}]
Observation: "You have clicked bright citrus.
Action:click[{"button": "3 ounce (pack of 1)"}]
Observation: "You have clicked 3 ounce (pack of 1).
Action:click[{"button": "Buy Now"}]
Observation: "You have bought the product.
Action:Finish[{"response": "Task Finished. Reward: 1.0"}]
Observation: "You have finished the shopping. Reward: 1.0
[End of Example]
[Execution]

Task: {{ task }}
```

Figure 4: Example of the agent's prompting and interaction with live websites.



Agent Input

```
<SYSTEM PROMPT>
<EXECUTION HISTORY>

USER QUERY: Book reservation for the restaurant
Ceconni's on OpenTable for 2 people on June 17 2024 at
7:00pm.

CURRENT OBSERVATION:
<p id=0>Make a reservation/>
...
<button id=8>Reserve a table/>
<p id=9/>
<p id=10>Booked 23 times today/>
<select id=11 label="Party size">2 people/>
<select id=12 label="Date">06/17/2024/>
<select id=13>6:00 PM/>
...
```

Agent Output

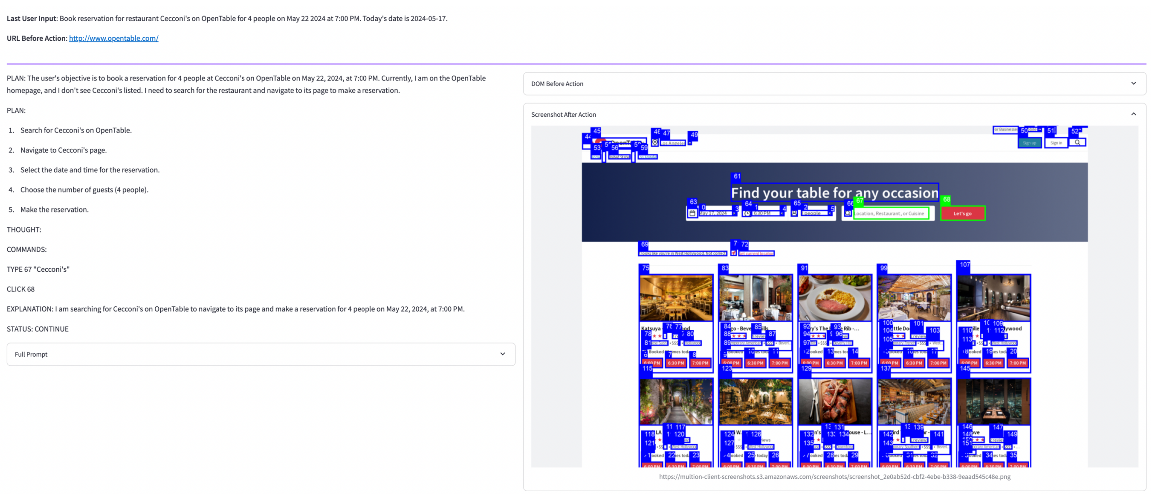
```
PLAN:
 1.   Select the date (May 22, 2024) for the
      reservation.
 2.   Choose the time (7:00 PM) for the reservation.
 3.   Select the party size (4 people) for the
      reservation.
 4.   Click on the "Find a table" button to book the
      reservation.
THOUGHT: I am currently on the Cecconi's restaurant
page on OpenTable, and I need to select the date and
time for the reservation and choose the number of
guests. I will focus on selecting the date, time,
and party size for the reservation.
COMMANDS:
CLICK 12
EXPLANATION: I am selecting the date, time, and
number of guests for the reservation at Cecconi's on
OpenTable.
STATUS: CONTINUE
```

Cecconi's

Cecconi's is a modern day classic Italian restaurant open for brunch, and dinner, seven days a week. Brunch is served on Saturday and Sunday. Cecconi's serves hand-made pasta, seafood and dishes from Italy using the finest ingredients. In addition to the flagship restaurant in Mayfair, London, Cecconi's has outposts in Los Angeles, Miami, Istanbul, Berlin, Barcelona and Brooklyn. As a courtesy to other guests, we do not accept any animals. Except for any ADA approved animal that is individually trained to do or perform tasks for the benefit of an individual with a disability.

What 2 people are saying

Figure 5: Examples of the agent's prompting and interaction with live websites.



Last User Input: Book reservation for restaurant Cecconi's on OpenTable for 4 people on May 22 2024 at 7:00 PM. Today's date is 2024-05-17.

URL Before Action: http://www.opentable.com/

PLAN: The user's objective is to book a reservation for 4 people at Cecconi's on OpenTable on May 22, 2024, at 7:00 PM. Currently, I am on the OpenTable homepage, and I don't see Cecconi's listed. I need to search for the restaurant and navigate to its page to make a reservation.

PLAN:

1. Search for Cecconi's on OpenTable.
2. Navigate to Cecconi's page.
3. Select the date and time for the reservation.
4. Choose the number of guests (4 people).
5. Make the reservation.

THOUGHT:

COMMANDS:

TYPE 67 "Cecconi's"

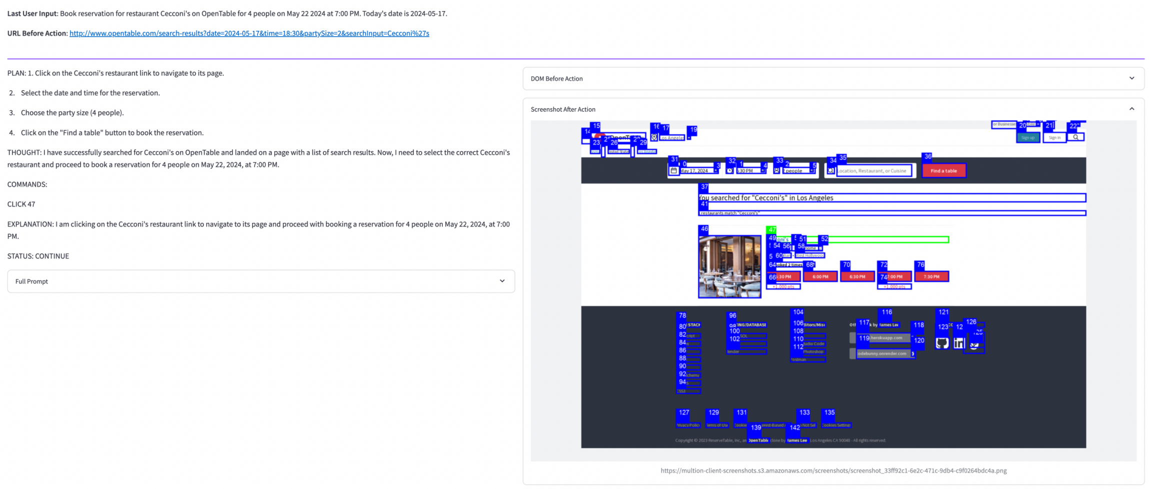
CLICK 68

EXPLANATION: I am searching for Cecconi's on OpenTable to navigate to its page and make a reservation for 4 people on May 22, 2024, at 7:00 PM.

STATUS: CONTINUE

https://multion-client-screenshots.s3.amazonaws.com/screenshots/screenshot_2e0ab52d-cbf2-4ebe-b338-9eaad545c48e.png

(a) Step 1: Agent generates a correct plan and successfully clicks search bar and searches for the user restaurant



Last User Input: Book reservation for restaurant Cecconi's on OpenTable for 4 people on May 22 2024 at 7:00 PM. Today's date is 2024-05-17.

URL Before Action: http://www.opentable.com/search-results?date=2024-05-17&time=18:30&partySize=2&searchInput=Cecconi%27s

PLAN: 1. Click on the Cecconi's restaurant link to navigate to its page.

2. Select the date and time for the reservation.
3. Choose the party size (4 people).
4. Click on the "Find a table" button to book the reservation.

THOUGHT: I have successfully searched for Cecconi's on OpenTable and landed on a page with a list of search results. Now, I need to select the correct Cecconi's restaurant and proceed to book a reservation for 4 people on May 22, 2024, at 7:00 PM.

COMMANDS:

CLICK 47

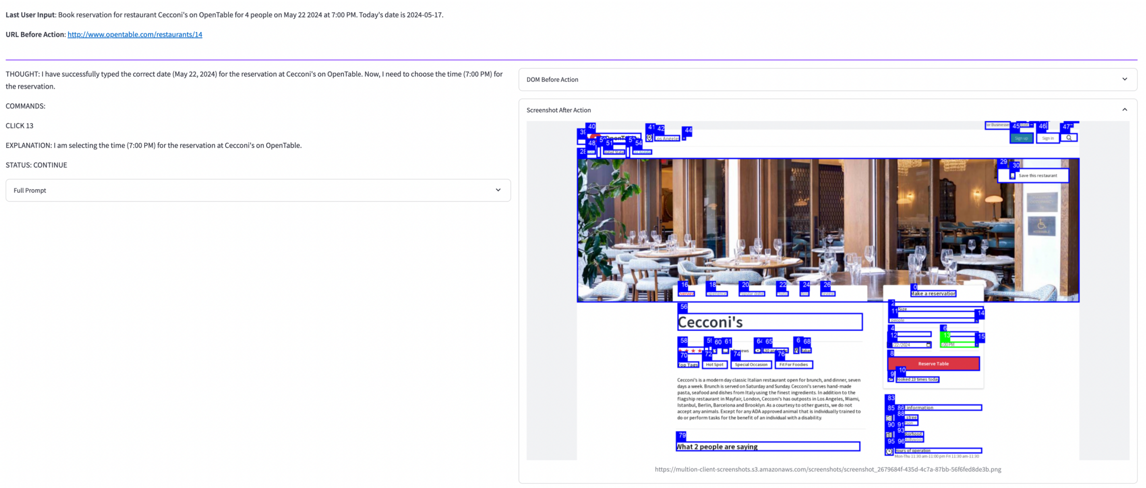
EXPLANATION: I am clicking on the Cecconi's restaurant link to navigate to its page and proceed with booking a reservation for 4 people on May 22, 2024, at 7:00 PM.

STATUS: CONTINUE

https://multion-client-screenshots.s3.amazonaws.com/screenshots/screenshot_33ff92c1-6e2c-471c-9db4-c9f0264bdc4a.png

Full Prompt
DOM Before Action
Screenshot After Action
Find a table
You searched for "Cecconi's" in Los Angeles
Copyright © 2023 ReserveTable, Inc., an OpenTable clone by James Lee Los Angeles CA 90048 - All rights reserved.

(b) Step 2: Agent clicks on correct restaurant.



Last User Input: Book reservation for restaurant Cecconi's on OpenTable for 4 people on May 22 2024 at 7:00 PM. Today's date is 2024-05-17.

URL Before Action: http://www.opentable.com/restaurants/14

THOUGHT: I have successfully typed the correct date (May 22, 2024) for the reservation at Cecconi's on OpenTable. Now, I need to choose the time (7:00 PM) for the reservation.

COMMANDS:

CLICK 13

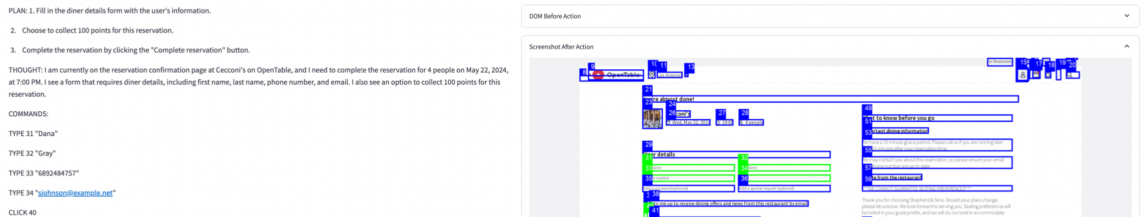
EXPLANATION: I am selecting the time (7:00 PM) for the reservation at Cecconi's on OpenTable.

STATUS: CONTINUE

Cecconi's is a modern day classic Italian restaurant open for brunch, and dinner, seven days a week. Brunch is served on Saturday and Sunday. Cecconi's serves hand-made pasta, seafood and dishes from Italy using the finest ingredients. In addition to the flagship restaurant in Mayfair, London, Cecconi's has outposts in Los Angeles, Miami, Istanbul, Berlin, Barcelona and Brooklyn. As a courtesy to other guests, we do not accept any animals. Except for any ADA approved animal that is individually trained to do or perform tasks for the benefit of an individual with a disability.

https://multion-client-screenshots.s3.amazonaws.com/screenshots/screenshot_2679684f-435d-4c7a-87bb-56f6fed8de3b.png

(c) Step 3-8: Agent modifies the party size, date, and time to match user request.



PLAN: 1. Fill in the diner details form with the user's information.

2. Choose to collect 100 points for this reservation.
3. Complete the reservation by clicking the "Complete reservation" button.

THOUGHT: I am currently on the reservation confirmation page at Cecconi's on OpenTable, and I need to complete the reservation for 4 people on May 22, 2024, at 7:00 PM. I see a form that requires diner details, including first name, last name, phone number, and email. I also see an option to collect 100 points for this reservation.

COMMANDS:

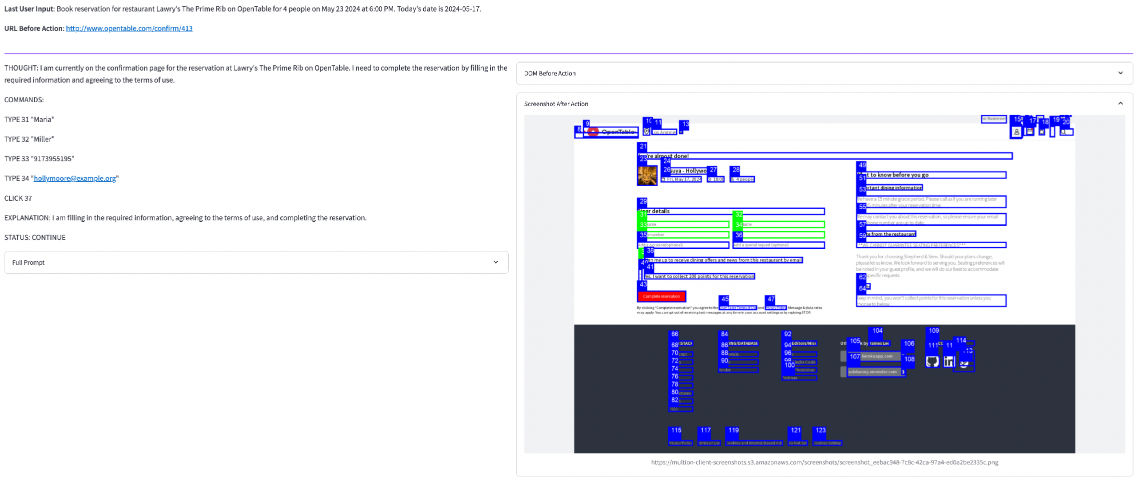
TYPE 31 "Dana"

TYPE 32 "Gray"

TYPE 33 "6892484757"

TYPE 34 "sjohnson@example.net"

CLICK 40



Last User Input: Book reservation for restaurant Lawry's The Prime Rib on OpenTable for 4 people on May 23 2024 at 6:00 PM. Today's date is 2024-05-17.

URL Before Action: http://www.opentable.com/confirm/413

THOUGHT: I am currently on the confirmation page for the reservation at Lawry's The Prime Rib on OpenTable. I need to complete the reservation by filling in the required information and agreeing to the terms of use.

COMMANDS:

TYPE 31 "Maria"

TYPE 32 "Miller"

TYPE 33 "9173955195"

TYPE 34 "hollymoore@example.org"

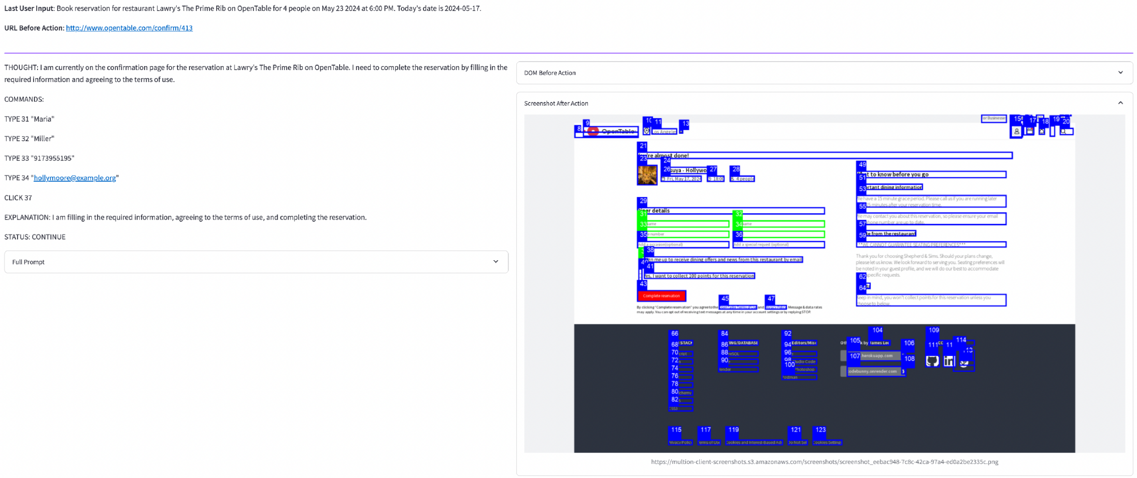
CLICK 37

EXPLANATION: I am filling in the required information, agreeing to the terms of use, and completing the reservation.

STATUS: CONTINUE

https://multion-client-screenshots.s3.amazonaws.com/screenshots/screenshot_eebac948-7c8c-42ca-97a4-ed0a2be2335c.png

(a) Agent thinks it has set the date and time, but has only set the search query date/time.



Last User Input: Book reservation for restaurant Lawry's The Prime Rib on OpenTable for 4 people on May 23 2024 at 6:00 PM. Today's date is 2024-05-17.

URL Before Action: http://www.opentable.com/confirm/413

THOUGHT: I am currently on the confirmation page for the reservation at Lawry's The Prime Rib on OpenTable. I need to complete the reservation by filling in the required information and agreeing to the terms of use.

COMMANDS:

TYPE 31 "Maria"

TYPE 32 "Miller"

TYPE 33 "9173955195"

TYPE 34 "hollymoore@example.org"

CLICK 37

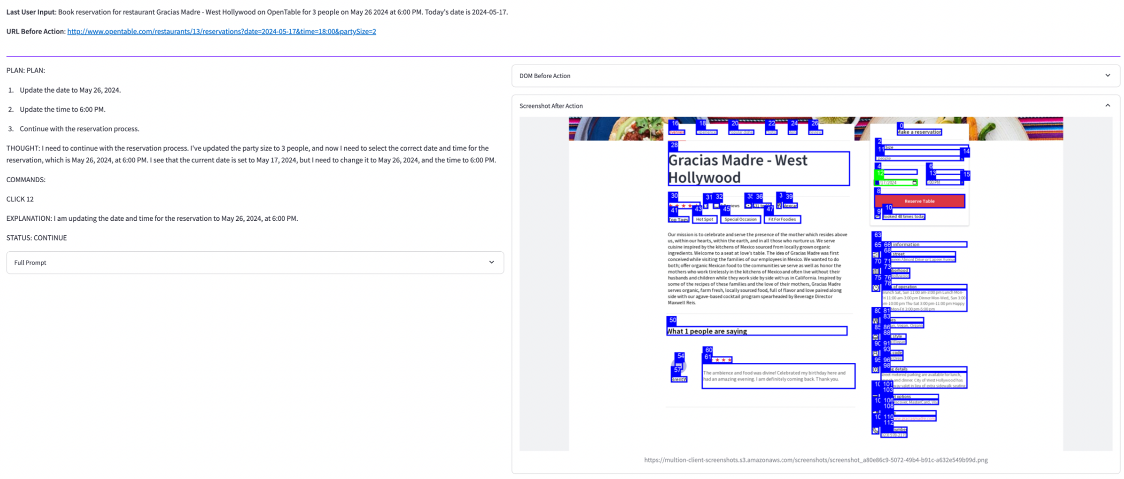
EXPLANATION: I am filling in the required information, agreeing to the terms of use, and completing the reservation.

STATUS: CONTINUE

https://multion-client-screenshots.s3.amazonaws.com/screenshots/screenshot_eebac948-7c8c-42ca-97a4-ed0a2be2335c.png

(b) Agent does not realize the search query date and time has not applied to the actual reservation.

Figure 7: Examples of agent not setting correct date time.



Last User Input: Book reservation for restaurant Gracias Madre - West Hollywood on OpenTable for 3 people on May 26 2024 at 6:00 PM. Today's date is 2024-05-17.

URL Before Action: http://www.opentable.com/restaurants/13/reservations?date=2024-05-17&time=18:00&partySize=2

PLAN: PLAN:

1. Update the date to May 26, 2024.
2. Update the time to 6:00 PM.
3. Continue with the reservation process.

THOUGHT: I need to continue with the reservation process. I've updated the party size to 3 people, and now I need to select the correct date and time for the reservation, which is May 26, 2024, at 6:00 PM. I see that the current date is set to May 17, 2024, but I need to change it to May 26, 2024, and the time to 6:00 PM.

COMMANDS:

CLICK 12

EXPLANATION: I am updating the date and time for the reservation to May 26, 2024, at 6:00 PM.

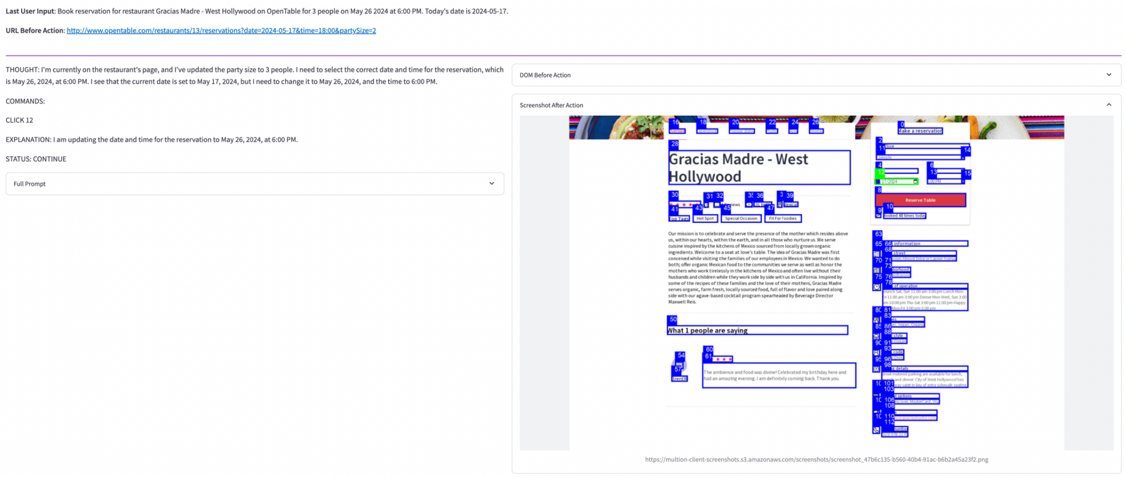
STATUS: CONTINUE

Our mission is to celebrate and serve the presence of the mother which resides above us, within our hearts, within the earth, and in all those who nurture us. We serve cuisine inspired by the kitchens of Mexico sourced from locally grown organic ingredients. Welcome to a seat at love's table. The idea of Gracias Madre was first conceived while visiting the families of our employees in Mexico. We wanted to do both; offer organic Mexican food to the communities we serve as well as honor the mothers who work tirelessly in the kitchens of Mexico and often live without their husbands and children while they work side by side with us in California. Inspired by some of the recipes of these families and the love of their mothers, Gracias Madre serves organic, farm fresh, locally sourced food, full of flavor and love paired along side with our agave-based cocktail program spearheaded by Beverage Director Maxwell Reis.

The ambience and food was divine! Celebrated my birthday here and had an amazing evening. I am definitely coming back. Thank you.

https://multion-client-screenshots.s3.amazonaws.com/screenshots/screenshot_a80e86c9-5072-49b4-b91c-a632e549b99d.png

(a) Agent continues clicking element ID 12, presumably expecting something to change in the environment, but is clicking on the textbox.



Last User Input: Book reservation for restaurant Gracias Madre - West Hollywood on OpenTable for 3 people on May 26 2024 at 6:00 PM. Today's date is 2024-05-17.

URL Before Action: http://www.opentable.com/restaurants/13/reservations?date=2024-05-17&time=18:00&partySize=2

THOUGHT: I'm currently on the restaurant's page, and I've updated the party size to 3 people. I need to select the correct date and time for the reservation, which is May 26, 2024, at 6:00 PM. I see that the current date is set to May 17, 2024, but I need to change it to May 26, 2024, and the time to 6:00 PM.

COMMANDS:

CLICK 12

EXPLANATION: I am updating the date and time for the reservation to May 26, 2024, at 6:00 PM.

STATUS: CONTINUE

Our mission is to celebrate and serve the presence of the mother which resides above us, within our hearts, within the earth, and in all those who nurture us. We serve cuisine inspired by the kitchens of Mexico sourced from locally grown organic ingredients. Welcome to a seat at love's table. The idea of Gracias Madre was first conceived while visiting the families of our employees in Mexico. We wanted to do both; offer organic Mexican food to the communities we serve as well as honor the mothers who work tirelessly in the kitchens of Mexico and often live without their husbands and children while they work side by side with us in California. Inspired by some of the recipes of these families and the love of their mothers, Gracias Madre serves organic, farm fresh, locally sourced food, full of flavor and love paired along side with our agave-based cocktail program spearheaded by Beverage Director Maxwell Reis.

The ambience and food was divine! Celebrated my birthday here and had an amazing evening. I am definitely coming back. Thank you.

https://multion-client-screenshots.s3.amazonaws.com/screenshots/screenshot_47b6c135-b560-40b4-91ac-b6b2a45a23f2.png

(b) Agent continues to click the same button in a loop.

Figure 8: Examples of agent not setting correct date time.