

BENCHMARKING IN-CONTEXT EXPERIENTIAL LEARNING THROUGH REPEATED PRODUCT RECOMMENDATIONS

Anonymous authors

Paper under double-blind review

ABSTRACT

To reliably navigate ever-shifting real-world environments, agents must grapple with incomplete knowledge and adapt their behavior through *experience*. However, current evaluations largely focus on tasks that leave no ambiguity, and do not measure agents' ability to adaptively learn and improve as they accrue experience. We exemplify the need for in-context experiential learning in a product recommendation context, where agents must navigate shifting customer preferences and product landscapes through natural language dialogue. We curate **BIEL**: a benchmark that combines i) rich real-world products from Amazon, ii) a diverse collection of user personas to represent heterogeneous yet latent preferences, and iii) a LLM user simulator powered by the persona to create realistic and interactive trajectories. We observe that current frontier models struggle to meaningfully improve across episodes, underscoring the need for agentic systems with strong in-context experiential learning capabilities.

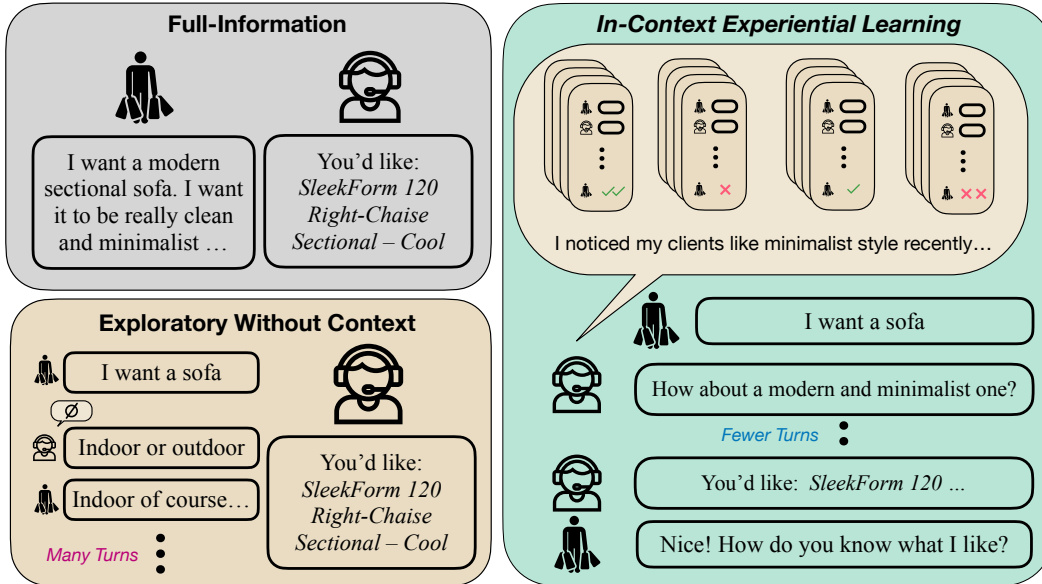


Figure 1: **Top Left.** Typical agentic benchmarks (e.g. Yao et al. (2024)) focus on settings where all information is provided initially, and the model is tasked with producing the correct answer in a zero-shot fashion. **Bottom Left.** Recent benchmarks (e.g. Li et al. (2024)) for LLM agents increasingly focus on multi-turn settings. **Right.** Our In-Context Experiential Learning setting.

Examples of Recommendation Dialogues for In-Context Experiential Learning		
Ep 1 (Hair Styling Gels):	Ep 2 (Packaged Rice Dishes):	Ep 3 (Almonds):
AGT : What is your hair type?	AGT : What's your budget range for packaged rice dishes?	AGT : Do you prefer raw or roasted almonds, or flavored varieties?
CUST 1: Curly.	CUST 1: I usually aim to keep it under \$10 per package.	CUST 2: Raw almonds.
AGT : What level of hold?	AGT : Do you have any dietary restrictions?	AGT : What are your preferences for this product category?
CUST 1: Medium.	CUST 1: I prefer a plant-based diet.	CUST 2: I appreciate products that are sustainable, environmentally friendly, and ethically sourced.
AGT : What is your budget range?	AGT : Do you prefer specific rice varieties?	
CUST 1: Around \$10 to \$15.	CUST 1: I prefer wild rice.	
AGT : Do you prefer a gel that provides moisture?	Selected Product:	Selected Product:
CUST 1: I prefer a gel that [...]	ID: 34444	ID: 72479
	<i>Frizz Free Styling Smoother</i>	<i>Roasted Sugared Almonds</i>
CUST 1 Feedback:	Selected Product:	CUST 2 Feedback:
I appreciate the suggestion, but I was actually looking for something more focused on care, like a conditioning balm or oil to keep it healthy and styled.	ID: 72329	Oh dear, I was really hoping for something more in line with sustainable and locally sourced goods. This recommendation seems a bit off.
	<i>Dr. McDougall's Vegan Pilaf</i>	
	CUST 1 Feedback:	
	Oh, this looks great! I really appreciate the suggestion, but I usually prefer something with more whole grains.	

Figure 2: **Benchmark for In-context Experiential Learning (BIEL):** An exemplar recommendation dialogues for in-context experiential learning across 2 customer personas and 3 choice sets.

1 INTRODUCTION

The ability to learn and improve from experience is a hallmark of intelligence. Real-world environments involve uncertainty arising from unobserved information, and intelligent agents must deliberately act to minimize mistakes and quickly learn from experience. However, the prevailing pre- and post-training paradigms primarily focus on knowledge distillation (Brown et al., 2020; Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Guo et al., 2025); while incredibly effective at tasks with little uncertainty (e.g., instruction following, math exams), resulting models often lack the ability to grapple with uncertainty, let alone to improve through repeated interactions with the environment (Liu et al., 2024; Zhou et al., 2024).

Following Silver & Sutton (2025), we refer the ability to adapt and improve from heterogeneous past interactions as in-context *experiential learning*. Without this ability, agents are confined to solving only familiar, fully-observable problems, leaving them brittle and ill-equipped to handle real-world tasks shaped by ever-changing environments and new uncertainties. Resilient and reliable agentic systems must be capable of long-horizon planning involving actively gathering costly yet informative feedback to reduce future uncertainty, reassessing uncertainty based on the feedback (“posterior updates”), and refining strategies over time.

We propose and construct a benchmark that measures the agent’s ability to reason through uncertainty, and make discoveries over time by leveraging past interactions / episodes. We exemplify in-context experiential learning capabilities using recommendation tasks (Figure 1), which offer a naturally dynamic environment characterized by a constant stream of new customers and products. An effective agent must actively discover users’ latent preferences through exploratory questions and iteratively refine recommendation based on ambiguous, text-based feedback. Notably, departing from the common formulation of partially observable Markov decision process (POMDP), we consider the “rewards” primarily encoded in *free-form natural language responses* (Yuksekgonul et al., 2025). The difference highlights a key requisite for a capable LLM agent: the ability to interpret and learn from natural language feedback.

Aspect	MediQ	Streambench	LMRL Gym	Science World	BIEL (ours)
Multi-turn	✓	✗	✓	✓	✓
Exploratory	✓	✗	✗	✓	✓
Scalability	✗	✗	✗	✗	✓
Experiential	✗	✓	✗	✓	✓

Table 1: Key axes evaluated across benchmarks. ✓ = satisfies, ✗ = does not, ✗ = mixed.

Recommendation tasks provide a fertile testbed for in-context experiential learning capabilities since each new customer and product introduces fresh *uncertainties*: the customers’ preferences are initially unknown, and new sets of products form unseen landscapes of available choices. An ideal recommender agent must actively plan its interactions, strategically select questions to elicit responses over multiple turns, and eventually provide a final recommendation. Consider two representative scenarios. In the first scenario (“personalization”), the agent must discover and attend to a particular user’s preference as it recommends different products and receives feedback over time. In the second scenario (“choice set”), the agent repeatedly sells a fixed choice set to a rotating pool of new customers and the focus shifts to learning how the products compare to each other across a diverse customer distribution.

In both of these scenarios, we envision agents that can actively discover user preferences through experience by leveraging multiple episodes. See Figure 2 for an example of our setting. Compared to works on pluralistic alignment or uncertainty quantification that focus on a single interaction/episode (Castricato et al., 2025; Zollo et al., 2025; Li et al., 2024), we primarily focus on the ability to learn across multiple interactions (*experiential learning*). Even when restricted to the personalization setting, we are interested in an agent that continuously interacts with the same customer, where the task is to gradually uncover and tailor the agent’s recommendation to the customer’s preferences.

We curate a large dataset for language-based recommendations by first pulling Amazon products from Hou et al. (2024) and categorizing them into a predefined list of categories (ASINSpotlight, 2023). This categorization enables us to form sets of interchangeable products that a customer might consider during a shopping session. Next, we draw on persona descriptions from Li et al. (2025) and use an LLM to simulate user preferences over these products. Simultaneously, the user-simulating LLM form the backbone of an interactive question-answering module, enabling realistic and dynamic interactions between agents and simulated users.

Beyond serving as a benchmark for evaluating experiential learning capabilities, our dataset offers a versatile and realistic framework for studying recommendation systems. By combining real-world product data, diverse user personas, and interactive dynamics powered by LLM simulator, it enables researchers to explore a wide range of questions including user modeling, preference elicitation, and cold-start recommendation, to name a few. We believe our dataset can be of separate and significant interest to the broader recommendation systems community.

Our main contributions are as follows:

- We formulate in-context experiential learning as a centerpiece to intelligence. Agents must be able to implicitly reason through uncertainty and refine their strategies by leveraging past experience.
- Going beyond sequential tool-use capabilities, we focus on multi-episodic settings where the agent necessarily have to mistakes initially due to ambiguity. Key differences from prior benchmarks are summarized in Table 1.
- We develop a Benchmark for In-context Experiential Learning (BIEL), a dataset of diverse and scalable product categories, including 71K products and 2K choice sets. Coupled with 1M scalably generated personas from Li et al. (2025), we support up to 2B multi-turn environments on which one can build a wealth of experiential learning settings.
- We observe even state-of-the-art models generally fail to exhibit meaningful learning across episodes, highlighting their inability to navigate ever-shifting real-world environments.

2 RELATED WORKS

RL for Language Model Training. The primary application of reinforcement learning (RL) in large language model (LLM) training has been Reinforcement Learning with Human Feedback (RLHF) (Christiano et al., 2017; Stiennon et al., 2020; Ziegler et al., 2020). RLHF has proven highly effective for aligning models with complex, difficult-to-quantify objectives that lack well-defined, differentiable reward functions: for example, RLHF has enabled models to improve on dimensions such as translation quality (Ramos et al., 2024; Kreutzer et al., 2018), helpfulness (Ouyang et al., 2022; OpenAI et al., 2024), and factual accuracy (Bai et al., 2022; Glaese et al., 2022; Touvron et al., 2023; Sun et al., 2024), among other desiderata. However, these common settings are fundamentally limited: they correspond to environments that consist of only a single-step interaction, where the model takes one action (i.e. generating a response), and immediately receives a scalar reward. While this paradigm has been remarkably successful for training high-performing zero-shot models, it fails to capture the interactive, dynamic nature of real-world decision-making, where actions and feedback over multiple turns or episodes can aid the models’ decisions. In contrast, our work focuses on multi-turn and multi-episode settings, and highlights the insufficiency of current state-of-the-art models to adapt and improve through interactions.

Multi-turn and Multi-episode RL. Recent years have witnessed a surge of interest in agentic models powered by LLMs (Jimenez et al., 2024; Yao et al., 2024; Karten et al., 2025). These agents are expected to autonomously plan, act and adapt through iterative interactions with their environment, requiring models to engage in *multi-turn interactions* within each episode, and to learn from experiences accumulated *across many episodes*. As a result, prior works have explored multi-turn settings in text-based games (Abdulhai et al., 2025; Tajwar et al., 2025), medical question-answering (Li et al., 2024), and numerous other tasks (Liu et al., 2024). A few other works have explored multi-episode settings (Wu et al., 2024; Zheng et al., 2025), but they primarily focus on environments devoid of uncertainties. Our experiential learning setting is closest to ScienceWorld (Wang et al., 2022), in which learning casual abstractions across episodes (Majumder et al., 2023) is the key to solving the tasks therein. In contrast, our setup focuses on assessing the ability of the models to *reason through uncertainties* of the latent preference of the customers in context.

Recommendation System and LLM Personalization. There has been a long line of work on recommendation systems (Resnick et al., 1994; Koren et al., 2009). In the era of foundation models, LLMs have demonstrated remarkable zero-shot performance in recommendation tasks (Geng et al., 2022; He et al., 2023; Lyu et al., 2024). However, the classical yet crucial setting of sequential recommendation (Hidasi et al., 2016; Tan et al., 2016) remains underexplored in this era, and the even more realistic problem of modeling interactions with customers has received little attention. Our dataset addresses this gap by enabling the study of these settings through an interactive user simulator powered by LLMs. Separate but relatedly, the rise of LLMs has sparked growing interest in personalized LLMs (Castricato et al., 2025; Jang et al., 2024; Zollo et al., 2025). Compared to these post-hoc heuristic approaches, we propose a more principled way to instill agents with the ability to learn to personalize. Our dataset is designed to directly evaluate and help advance this capability.

3 RECOMMENDATION SYSTEMS AS A FERTILE TESTBED FOR EXPERIENTIAL LEARNING

Recommendation systems provide a rich foundation for testing the experiential learning capabilities of agentic systems. Consider a common shopping scenario: a *customer* (c) enters a store in seek of a product to satisfy a specific need. Many products fulfill the same functional purpose, but they differ in style, aesthetics, or price, leading to a preference unique to the customer. We refer to this group of functionally equivalent products as a *choice set* (S). The recommender agent must discover customer’s preferences over S by engaging in *multiple turns* of queries: asking targeted questions, receiving, and interpreting the customer’s response. This entire interaction, from initial inquiry to final recommendation, constitutes an *episode*.

Following each recommendation, the agent typically receives *feedback*, such as a purchase decision or free-form text opinions about the suggested product. This feedback, combined with the intermediate responses, encodes rich information about the *latent factors* (θ) underlying a given episode.

This latent factor may include the customer’s preferences, the dynamics of the question-answering process, how the customer perceives the structure of the choice set, and so forth. Compared to a typical sequential decision-making setup (e.g., POMDPs), the “reward” based on which the agent should optimize its strategy is encoded primarily in text as part of the observations (Yuksekgonul et al., 2025). A capable agent must interpret these text-based signals to sharpen its belief of the latent θ to improve its recommendations in subsequent episodes.

A performant recommendation system must learn across users and products and improve based on experience. Thus, we are not merely interested in the recommender agent’s performance within a single episode, but rather their ability to improve as its experience accrues. For instance, consider a personalized recommendation setting where the same customer interacts with the agent across multiple episodes. In this case, the agent should focus on uncovering the customer’s underlying preferences to improve future recommendations. Conversely, imagine a scenario where the agent repeatedly sells a fixed choice set of products to a stream of new customers. Here, the goal shifts to identifying how these products compare relative to one another across the diverse distribution of customers. In the most difficult setting, both customers and choice sets can evolve over time.

Formally, the agent encounters a *stream of episodes* over its lifetime, where each e -th episode is characterized by a tuple $(c_e, \mathcal{S}_e)$ that induces a corresponding latent factor θ_e . Each episode, indexed by $e = 1, \dots, E$, represents a single shopping session defined by a tuple $(c_e, \mathcal{S}_e)$, representing a customer and a choice set of products. In an episode (shopping session), customer c_e is interested in buying one product from $\mathcal{S}_e$ according to their preference. Each episode contains a sequence of turns where each turn consists of the recommender’s question or recommendation $a_{e,t}$, and the customer’s response $o_{e,t}$. If the action $a_{e,t}$ is a question, then the response $o_{e,t}$ is the customer’s answer; if $a_{e,t}$ is a recommendation, then $o_{e,t}$ is a feedback provided by the customer, encoding the customer’s preference over the recommended product. Naturally, the e -th episode concludes after a recommendation is made, and the task moves on to the next episode. We denote this final feedback as f_e for convenience.

We denote the the sequence of latent factors by $\Theta = \{\theta_e \mid e = 1, 2, \dots\}$; some components of these latent factors may remain stable over time, while others may shift. An intelligent agent must learn to adapt to the dynamics of Θ in an online fashion. At each turn, the response $o_{e,t} = g(\theta_e, a_{e,t})$ is a function of both the action and the latent factor of the episode. At episode e and turn t , the entire history available to the recommender agent is

$$\mathcal{H}_{e,t} = \underbrace{\{(a_{e',t'}, o_{e',t'}) \mid \forall e' \in [e-1], \forall t' \in [T_{e'}]\}}_{\text{Past Episodes}} \cup \underbrace{\{(a_{e,t'}, o_{e,t'}) \mid \forall t' \in [t-1]\}}_{\text{Current Episode}},$$

with which the agent $\pi(\cdot)$ takes an action $a_{e,t} = \pi(\mathcal{H}_{e,t})$.

To measure the quality of the recommendations, we score the products for each unique pair of $(\mathcal{S}, c)$ that defines an episode. Given one such pair, for each product $p_a \in \mathcal{S}$, we require a score $y_a = h(p_a, c)$. The main metric of merit for the recommendation is the *regret*: $y^* - y_r$, where y^* is a highest score, $y^* = \max(\{y_a \mid y_a = h(p_a, c, \mathcal{S}) \mid p_a \in \mathcal{S}\})$, and y_r is the score of the recommended product.

4 BENCHMARK FOR IN-CONTEXT EXPERIENTIAL LEARNING (BIEL)

To instantiate the above experiential learning framework, our benchmark comprises of two main modules: (1) a set of $\mathcal{C} = \{c_i\}$ of hypothetical customers that can respond *interactively* to the recommender, and (2) a set $\mathcal{S} = \{\mathcal{S}_j\}$ of choice sets. For both modules, we want the underlying sets to be scalable and diverse to support nuanced scoring and a wide variety of Θ dynamics. We address this challenge by designing novel filtering schemes that allow us to construct a large-scale language-based product recommendation dataset, consisting of 71K products and 2K choice sets. (Since some categories share products, the dataset contains in total 100K product-category pairs.)

Persona-Based Customer Simulation. To power the customer module, we use the 1M persona specifications in Li et al. (2025) to provide diversity in $\mathcal{C}$. We then utilize LLMs to simulate the customer’s response corresponding to a persona file to enable interactive question-answering. The persona therein are generated by LLMs, and hence are as scalable and diverse as what modern LLMs encode.

An Example Persona and Their Preferences		
Meet Karen Thompson, a 59-year-old woman living in Minneapolis, Minnesota. She has short, curly brown hair and bright blue eyes, often accentuated by her warm and welcoming smile. [...]		
Simplified Preference Scores (Hair Styling Gels)		
ID	Product	Score
40255	Tigi Bed Head Foxy Curls Contour Creme, 6.76 oz	95.0
34764	Shea Moisture Sacha Inchi Oil Curl Defining Smoothie, 12 oz	85.0
...	... more products ...	...
35903	NKICAW IVILA Hair Straightening Cream (3PCS)	15.0

Figure 3: An exemplar persona taken from Li et al. (2025) and their preferences over a category of products from Hou et al. (2024). Scoring is done by GPT-4o and Gemini-1.5-Pro. Consistent with the persona’s curly hair, curl-enhancing products are rated highly, whereas straightening products receive low scores.

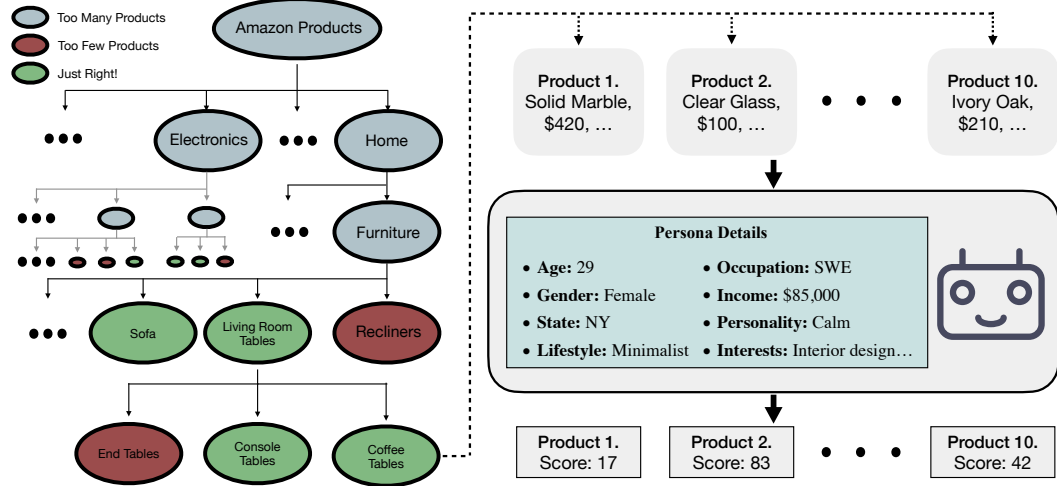


Figure 4: We use a predefined tree of categories (ASINSpotlight, 2023) and filtered the ones unsuitable as choice sets. We then score the products within each choice set with a persona-simulating LLM.

Products and Choice Sets Curation. To form the choice sets module, we need a pool of products, correctly categorized into reasonable choice sets. For the pool of products, we sampled 100K products from the Amazon Reviews dataset (Hou et al., 2024), which consists of 34 primary product categories. We applied filters to remove products missing essential fields such as product descriptions. We match each product to a predefined category from ASINSpotlight (2023). Through a rigorous matching process, we were able to assign 70% of the sampled products to the categories of ASINSpotlight (2023); see Appendix C for additional details.

Then, we extract $\mathcal{S}$ as a subset of predefined categories, consisting of categories that could reasonably be treated as choice sets. We first truncate the top two levels of this category hierarchy as these levels contained overly broad classifications ill-suited as choice sets (e.g., Home & Kitchen). We also filter categories that contain too few products to get nontrivial recommendation settings. See Figure 4 for an illustration of the process.

On scalability of available choice sets, one can easily imagine expanding the number of valid choice sets by sampling additional products from Hou et al. (2024) or other large-scale datasets. to increase the number of valid choice sets using our pipeline. Moreover, our framework is designed to be broadly compatible, allowing it to incorporate any choice set that contains a sufficient number of products with descriptions. More sophisticated or specialized data collection pipelines could be developed to gather richer and more diverse choice sets, further extending the scope of our benchmark.

Preference Generation. For preference generation ($y_a = h(p_a, c, \mathcal{S})$), the scoring function $h(\cdot)$ was an average of the scores generated using two models: GPT-4o and Gemini-1.5-Pro. Each model received the full description of the current persona encoding c , along with details of the products in the set $\mathcal{S}$ under evaluation. The models were instructed to assign a score (0–100) reflecting how much the persona would enjoy or appreciate each product within $\mathcal{S}$. The consistency of the scoring process is crucial to the integrity of the benchmark. To evaluate the consistency, we randomly selected five categories and measure the average variation in the scores across three seeds. We found that the averaged score variation is 4.1, much smaller than the standard deviation in scores within each category (25.4), which suggests that the scoring process is consistent. For instance, in Figure 3, the persona is described as having *curly brown hair*. Consistent with this trait, products intended for curl enhancement are assigned high scores, while products designed for hair straightening receive lower scores.

Customer Simulation and Feedback. Users are simulated by a LLM (GPT-4o) prompted to act as the persona when shopping within a given product category. To prevent unrealistic scenarios (e.g., an individual with no musical interests shopping for pianos), we exclude categories in which the highest product score is below 60. Using this filtering procedure, an average of 3.75 categories were skipped across the first 20 seeds.

Performance measures. At the end of each e -th episode, the simulated user provides a feedback f_e . We support three forms of feedback: (1) **Regret**, (2) **Stars**, and (3) **Free-form Text**. Regret feedback directly provides the regret of the recommendation. While informative, this metric is unrealistic in practice, as it is typically infeasible to obtain exact utility scores from real users. To approximate more practical feedback, we provide star ratings as $5 \times \frac{\text{score of chosen product}}{\text{score of best product}}$, rounded to the nearest integer, mirroring the star systems commonly found on e-commerce platforms. Finally, free-form text feedback involves prompting the customer LLM with the recommendation, its regret value, and the top three scored products, and instructing it to generate natural language feedback—expressed in the persona’s voice on the extent to which the chosen product meets their preferences. See Figure 2 for examples. The generated feedback reflects the persona’s curly hair type and consistently maintains the persona’s character throughout the interaction.

Robustness check. To rule out the possibility that poor agent performance stems from an impossible task, we conducted a manual, human-curated questioning run as seen in Appendix A.5. The questions were deliberately *reasonable* in that they (i) targeted concrete, product-determining attributes, (ii) prompted the persona to reveal personality details, and (iii) were answerable without specialized knowledge. Under this regime, the agent identified the best scoring product (regret = 0) when it had otherwise consistently chosen a product with regret 37.5. This demonstrates that, given a well-planned query policy, the information required to reach the best item is accessible, and the environment is *solvable*.

5 EXPERIMENTS

We study three sequential recommendation settings that exemplify experiential learning: (1) same customer across episodes with different categories each episode, (2) same category across episodes with different customers each episode, and (3) different categories and customers each episode. Of course, the rich set of personas and categories available can support many more settings than the ones considered in this section.

To evaluate performance, aside from the aforementioned regret metrics, we monitor the number of questions asked. In our experiments, recommender agents are not explicitly prompted to opt for fewer questions. Therefore, they should ask as many questions as needed to resolve uncertainties for recommendation. We consider three baselines throughout. RANDOM recommends products uniformly at random, and POPULARITY selects the product with the highest original rating, breaking

ties by choosing the cheaper option. Finally, ORACLE is a Claude-Sonnet-4 with access to the full persona description and prompted to directly select a product from the category. Since Claude is not used during the initial scoring of products, this setup ensures a clean separation and serves as a lower bound on regret in cases where the recommender has full access to persona information.

5.1 COMPARING BASE MODELS

We first evaluate five models—GPT-4o, Gemini-2.5-Pro, Gemini-2.5-Flash, Claude-Opus-4, and Claude-Sonnet-4—over 10 episodes across 40 random seeds. Across these experiments, we use the Free-form Text feedback.

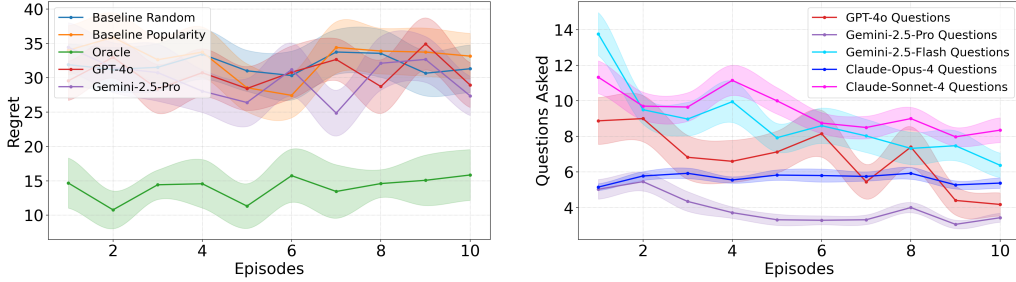


Figure 5: **LEFT:** The models are not learning from previous experiences. **RIGHT:** The models tend to ask fewer questions in later episodes, despite failing to learn from the interactions.

As shown in the left panel of Figure 5, none of the models achieve meaningful improvements over simple baselines, and all significantly underperform the oracle baseline. This highlights the difficulty of leveraging prior episodes for better recommendations for current SOTA models. Full ablations can be found in Appendix B.

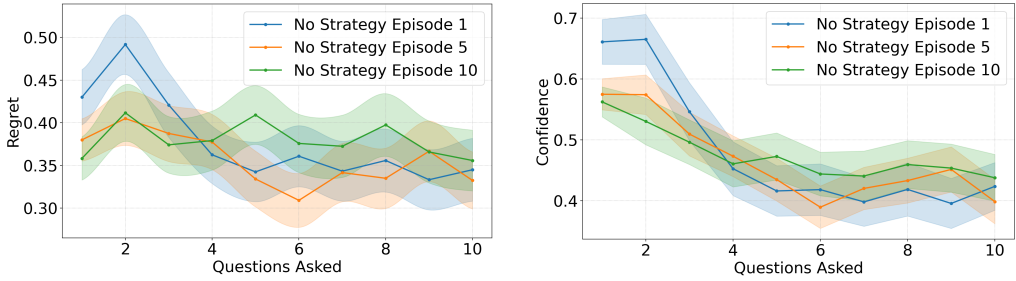


Figure 6: **LEFT:** GPT-4o performs no better across episodes nor over questions asked. **RIGHT:** GPT-4o grows less confident with more questions asked. Results are shown for confidence on regret within 10.

If agents perform poorly, we expect them to ask more questions to improve their recommendations. However, on the right of Figure 5, we observe that the number of questions asked generally declines across episodes. We highlight this as a major deficiency in the behavior patterns of the SOTA models.

To understand how models implicitly quantify its own uncertainty, we prompt the GPT-4o to output six kinds of confidences on: (1-2) whether the chosen item would fall into top-1 and top-5 favorite categories of the customer, and (3-6) whether the realized regret would fall into the expected regret within 5, 10, 20, and 30 points. See Figure 7. Across all settings, we found that the model is largely poorly calibrated. See Appendix B.5 for complete statistics.

An occasional issue with GPT-4o was an endless loop of repetitive questions. At times, the model would repeatedly ask the same queries despite having access to the full conversation history, continuing until the 20-question limit was reached. Increasing the temperature somewhat reduced the

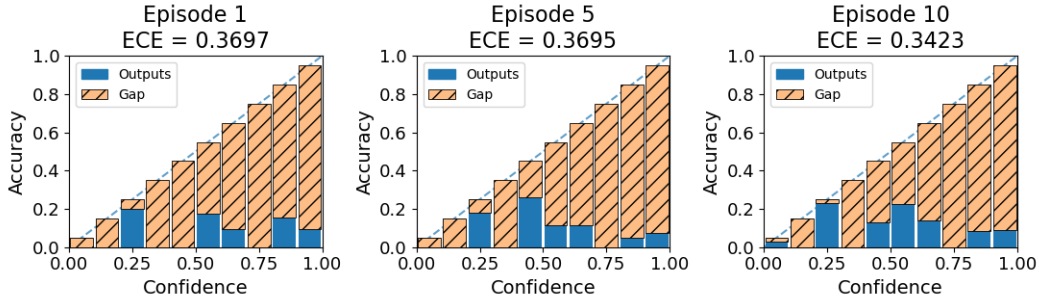


Figure 7: ECE plots for GPT-4o with No Strategy. The model is clearly poorly calibrated. Results are shown for confidence on regret within 10 across all turns.

frequency of this behavior, but did not eliminate it entirely. We observed similar conversational looping with other models as well; both Claude-Sonnet-4 and Gemini-2.5-Flash occasionally fell into these repetitive patterns.

5.2 PLANNING CAPABILITIES VIA PROMPTING

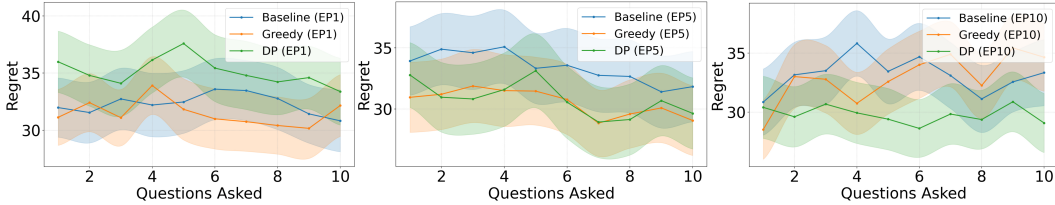


Figure 8: Performance of different prompting strategies across episodes (from left to right) with GPT-4o. Observe that only DP prompts seem to induce learning across episodes, whereas other prompts fail to improve.

We investigate whether explicit prompting strategies could encourage more deliberate planning in recommendations. We consider three variants were evaluated using GPT-4o over 60 random seeds: (a) simply prompts the agents to ask questions for T turns (**No Strategy**), (b) explicitly instructs to act greedily (**Greedy**), and (c) instructs to act as a DP-optimal planner in a POMDP (**DP**). In all planning experiments, the agent was run for 10 episodes under standard conditions, with additional evaluations conducted in the 1st, 5th, and 10th episodes. In these episodes, the agent was additionally prompted to make a recommendation after each question, allowing us to measure regret at every turn. Figure 8 reports a recommendation as a function of the number of questions asked. Greedy prompting occasionally yields some reductions in regret relative to the baseline, but does not improve across episodes. DP-style prompting exhibits greater improvements over episodes; by episode 10, it shows a reduction in regret compared to both the baseline and greedy prompting. This shows that prompting may encourage agents to learn, but the performance is nonetheless far worse than ORACLE.

6 CONCLUSION

We argue for a shift in focus toward measuring agents’ ability to engage in adaptive, multi-episode interactions. To support this goal, we introduced a sequential recommendation dataset designed to evaluate in-context experiential learning capabilities. Beyond its use for benchmarking agents, our dataset may also be of independent interest to the broader recommender systems community.

Our experiments reveal that SOTA models do not learn from experience. They struggle to calibrate their uncertainties and, as a result, fail to proactively ask informative questions. By highlighting these limitations, we aim to motivate future research toward improving these critical abilities, which are essential for building reliable, real-world agents.

7 REPRODUCIBILITY STATEMENT

We provide the code used for all the experiments in this paper in BIEL.zip of <https://github.com/17my15/BIEL/releases>.

REFERENCES

- Marwa Abdulhai, Isadora White, Charlie Victor Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. LMRL gym: Benchmarks for multi-turn reinforcement learning with language models. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=hmGhP5D02W>.
- ASINSpotlight. Amazon Categories List CSV, 2023. URL <https://www.asinspotlight.com/amz-categories-list-csv>. Accessed: 7 September 2025.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Ols-son, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. URL <https://arxiv.org/abs/2212.08073>.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.
- Louis Castricato, Nathan Lile, Rafael Rafailov, Jan-Philipp Fränken, and Chelsea Finn. PER-SONA: A reproducible testbed for pluralistic alignment. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 11348–11368, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL <https://aclanthology.org/2025.coling-main.752/>.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/d5e2c0adad503c91f91df240d0cd4e49-Paper.pdf.
- Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM Conference on Recommender Systems*, RecSys ’22, pp. 299–315, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392785. doi: 10.1145/3523227.3546767. URL <https://doi.org/10.1145/3523227.3546767>.
- Amelia Glaese, Nat McAleese, Maja Trebacz, John Aslanides, Vlad Firoiu, Timo Ewalds, Mari-beth Rauh, Laura Weidinger, Martin Chadwick, Phoebe Thacker, Lucy Campbell-Gillingham, Jonathan Uesato, Po-Sen Huang, Ramona Comanescu, Fan Yang, Abigail See, Sumanth Dathathri, Rory Greig, Charlie Chen, Doug Fritz, Jaume Sanchez Elias, Richard Green, Soňa

- Mokrá, Nicholas Fernando, Boxi Wu, Rachel Foley, Susannah Young, Iason Gabriel, William Isaac, John Mellor, Demis Hassabis, Koray Kavukcuoglu, Lisa Anne Hendricks, and Geoffrey Irving. Improving alignment of dialogue agents via targeted human judgements, 2022. URL <https://arxiv.org/abs/2209.14375>.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shiron Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, Sep 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL <https://doi.org/10.1038/s41586-025-09422-z>.
- Zhankui He, Zhouhang Xie, Rahul Jha, Harald Steck, Dawen Liang, Yesu Feng, Bodhisattwa Prasad Majumder, Nathan Kallus, and Julian McAuley. Large language models as zero-shot conversational recommenders. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM ’23, pp. 720–730, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3614949. URL <https://doi.org/10.1145/3583780.3614949>.
- Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, 2016.
- Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.
- Joel Jang, Seungone Kim, Bill Yuchen Lin, Yizhong Wang, Jack Hessel, Luke Zettlemoyer, Hananeh Hajishirzi, Yejin Choi, and Prithviraj Ammanabrolu. Personalized soups: Personalized large language model alignment via post-hoc parameter merging. In *Adaptive Foundation Models: Evolving AI for Personalized and Efficient Learning*, 2024. URL <https://openreview.net/forum?id=EMrnoPRvxex>.
- Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R Narasimhan. SWE-bench: Can language models resolve real-world github issues? In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=VTF8yNQM66>.

- Seth Karten, Jake Grigsby, Stephanie Milani, Kiran Vodrahalli, Amy Zhang, Fei Fang, Yuke Zhu, and Chi Jin. The pokeagent challenge: Competitive and long-context learning at scale. In *NeurIPS Competition Track*, April 2025.
- Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009. doi: 10.1109/MC.2009.263.
- Julia Kreutzer, Joshua Uyheng, and Stefan Riezler. Reliability and learnability of human bandit feedback for sequence-to-sequence reinforcement learning. In Iryna Gurevych and Yusuke Miyao (eds.), *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1777–1788, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1165. URL <https://aclanthology.org/P18-1165/>.
- Ang Li, Haozhe Chen, Hongseok Namkoong, and Tianyi Peng. Llm generated persona is a promise with a catch, 2025. URL <https://arxiv.org/abs/2503.16527>.
- Shuyue Stella Li, Vidhisha Balachandran, Shangbin Feng, Jonathan S. Ilgen, Emma Pierson, Pang Wei Koh, and Yulia Tsvetkov. Mediq: Question-asking LLMs and a benchmark for reliable interactive clinical reasoning. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=W4pIBQ7bAI>.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating LLMs as agents. In *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=zAdUB0aCTQ>.
- Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. LLM-rec: Personalized recommendation via prompting large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 583–612, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.39. URL <https://aclanthology.org/2024.findings-naacl.39/>.
- Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. CLIN: A continually learning language agent for rapid task adaptation and generalization. In *Second Agent Learning in Open-Endedness Workshop*, 2023. URL <https://openreview.net/forum?id=8wgNZ7Kado>.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen

- Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O’Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report, 2024. URL <https://arxiv.org/abs/2303.08774>.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/blfede53be364a73914f58805a001731-Paper-Conference.pdf.
- Miguel Ramos, Patrick Fernandes, António Farinhas, and Andre Martins. Aligning neural machine translation models: Human feedback in training and inference. In Carolina Scarton, Charlotte Prescott, Chris Bayliss, Chris Oakley, Joanna Wright, Stuart Wrigley, Xingyi Song, Edward Gow-Smith, Rachel Bawden, Víctor M Sánchez-Cartagena, Patrick Cadwell, Ekaterina Lapshinova-Koltunski, Vera Cabarrão, Konstantinos Chatzitheodorou, Mary Nurminen, Diptesh Kanojia, and Helena Moniz (eds.), *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pp. 258–274, Sheffield, UK, June 2024. European Association for Machine Translation (EAMT). URL <https://aclanthology.org/2024.eamt-1.22/>.
- Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. Grouplens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work, CSCW ’94*, pp. 175–186, New York, NY, USA, 1994. Association for Computing Machinery. ISBN 0897916891. doi: 10.1145/192844.192905. URL <https://doi.org/10.1145/192844.192905>.
- David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 1, 2025.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.

- Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liangyan Gui, Yu-Xiong Wang, Yiming Yang, Kurt Keutzer, and Trevor Darrell. Aligning large multimodal models with factually augmented RLHF. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13088–13110, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.775. URL <https://aclanthology.org/2024.findings-acl.775/>.
- Fahim Tajwar, Yiding Jiang, Abitha Thankaraj, Sumaita Sadia Rahman, J Zico Kolter, Jeff Schneider, and Russ Salakhutdinov. Training a generally curious agent. In *Forty-second International Conference on Machine Learning*, 2025. URL <https://openreview.net/forum?id=UeB3Hdrrhda>.
- Yong Kiam Tan, Xinxing Xu, and Yong Liu. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*, DLRS 2016, pp. 17–22, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450347952. doi: 10.1145/2988450.2988452. URL <https://doi.org/10.1145/2988450.2988452>.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL <https://arxiv.org/abs/2307.09288>.
- Ruoyao Wang, Peter Jansen, Marc-Alexandre Côté, and Prithviraj Ammanabrolu. ScienceWorld: Is your agent smarter than a 5th grader? In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang (eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11279–11298, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.775. URL <https://aclanthology.org/2022.emnlp-main.775/>.
- Cheng-Kuang Wu, Zhi Rui Tam, Chieh-Yen Lin, Yun-Nung Chen, and Hung yi Lee. Streambench: Towards benchmarking continuous improvement of language agents. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. URL <https://openreview.net/forum?id=8hUUy3hoS8>.
- Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ -bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL <https://arxiv.org/abs/2406.12045>.
- Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639:609–616, 2025.
- Junhao Zheng, Xidi Cai, Qiuke Li, Duzhen Zhang, ZhongZhi Li, Yingying Zhang, Le Song, and Qianli Ma. LifelongAgentBench: Evaluating llm agents as lifelong learners, 2025. URL <https://arxiv.org/abs/2505.11942>.
- 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 *The Twelfth International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=oKn9c6ytLx>.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2020. URL <https://arxiv.org/abs/1909.08593>.

Thomas P Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. PersonalLLM: Tailoring LLMs to individual preferences. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=2R7498e2Tx>.

A PROMPTS

A.1 ORACLE AGENT

We present the prompt used to define the oracle baseline. The oracle agent is given the full persona description and the complete set of candidate products, and is instructed to return only the index of the single best product. This setting establishes a lower bound on regret under full-information conditions.

Prompt for Oracle Agent Baseline

System Message: You are an oracle recommendation agent with perfect knowledge of a customer’s preferences.

Customer Persona:

{self.persona.description}

Product Category: {category}

Available Products:

{self._format_products(products)}

Task: Given the customer’s complete persona description, choose the single best product that would most satisfy their preferences and needs. You have perfect knowledge of what this customer would want.

Output format (MUST be exactly one line, no extra text):

RECOMMEND: <array_index_0_to_{num_products-1}>

Rules:

- Choose the product that best matches the customer’s persona.
- Consider all aspects of their preferences, lifestyle, and needs.
- Return the array index (0-based), not the product ID.
- No explanations, just the recommendation index.

Figure 9: Prompt used to instantiate the oracle baseline.

A.2 PERSONA AGENTS

We introduce the prompts used for the persona agents. These agents are employed in three settings: (1) scoring personas, (2) generating dialogue, and (3) producing persona-based feedback.

Prompts for Customer Simulation

(a) Preference Generation

“persona_des” : “...”

“category” : “...”

“products” : “[...]”

“instructions”: “You ARE the persona described. Rate each product with a score from 0 to 100 (integers only) based on how much YOU would like it. Return a JSON object with key ‘results’ as an array of objects: {id, score}. Do not include any other keys or text.”

(b) Response Simulation

“persona_des” : “...”

“question” : “...”

“instructions”: “You simulate a user with the given persona description. Answer **strictly** as this persona would: – Only answer the question asked. – Do not re-state persona or add rationale. – If a choice is requested, give one choice only. Return the answer as plain text.”

Figure 10: Prompts for simulating customer behavior with persona agents: (a) generating product preference scores, (b) producing persona-consistent responses to agent queries.

Prompt for User Feedback Response

Reminder: You are a user with this persona:
`{self._persona_text}`

A recommendation agent just suggested a product to you.

Context:

- `{chosen_info}`
- `{conversation_context}`

`{tone_instruction}`

Task: Respond naturally as this persona would—like you’re talking to a helpful salesperson or friend. Be conversational and specific about your preferences. Keep it to 1–2 sentences and sound like a real person, not a formal review. Make it a statement about your preferences, not a question. Never mention specific scores, regret values, or reveal which product would be better.

Your response:

Figure 11: Prompt for eliciting naturalistic persona feedback following a recommendation. The tone is adjusted based on the quality of the recommendation, estimated through regret.

A.3 RECOMMENDER AGENT

We describe the prompts used to instantiate the recommender agent. At its core, the agent is queried at each turn with a baseline prompt, shown in Figure 12, which specifies the available context and requires the model to either ask one clarifying question or make a recommendation.

To evaluate whether agents can exploit past interactions, we additionally replace the raw conversation history with a summary of prior episodes, producing the variant shown in Figure 13.

To test whether explicit reasoning instructions improve performance, we append a chain-of-thought style enhancement to the baseline prompt, shown in Figure 14. These prompt designs define the controlled conditions under which we ablate model behavior, with results reported in Appendix B.

Prompt for Recommender Agent Action

System Message: You are a product recommendation agent. Your goal is to find the best product for this user, while asking the fewest number of questions before being confident in the best product for the user.

Context:

`{context}`
`{feedback_context}`

Task: First, analyze what you already know from the conversation. Then, either:

- Ask one short, consumer-friendly question to clarify user preferences, or
- If sufficiently confident, recommend one product by index.

CRITICAL OUTPUT FORMAT (MUST FOLLOW EXACTLY):

- To ask a question: **QUESTION:** [your question here]
- To recommend: **RECOMMEND:** [number between 0 and `{num_products-1}`]

STRICT RULES:

- Your response must start with either “**QUESTION:**” or “**RECOMMEND:**”.
- Do NOT include any explanations, reasoning, or additional text.
- Do NOT use bullets, multiple lines, or formatting.

Figure 12: LLM prompt for the recommender agent to decide its next action (ask or recommend).

Prompts for Episode Generation Summary

System Message: You just completed Episode {episode_num} in the {category} category for Persona {persona}.

Episode Details:

{dialog_text}

Selected Product: {selected_product_id}

Feedback: {feedback}

Your task: Provide the context from this episode that you would want a future agent to know. Focus on:

- What worked or didn't work in your approach.
- Key insights about user preferences or product selection.
- Any patterns you noticed that could help in similar situations.

Instruction: Write only the summary, no additional commentary.

Figure 13: LLM prompt variant for generating episode summaries.

Prompts Enhancement for Chain-of-Thought (CoT)

Let me think through this systematically:

- Customer preferences: [analyze what I know]
- Available products: [analyze the options]
- Best match: [reason about the best choice]
- Decision: [decide whether to ask or recommend]

Let's reason step by step:

1. What do I know about the customer so far?
2. What information am I still missing?
3. Based on this reasoning, what should I do next?

Before making your decision, think again: What are you unsure about regarding this customer? What questions should you ask next? Consider what additional information would help you make a better recommendation.

Think through each step carefully before responding.

Figure 14: Chain-of-Thought (CoT) enhancement appended to agent prompts to test the reasoning process.

A.4 PLANNING AGENTS

In addition to the baseline recommender prompts described above, we introduce prompts designed to explicitly encourage planning behavior. These variants aim to test whether models can adopt more deliberate strategies for information gathering rather than defaulting to shallow heuristics.

The **Greedy** prompt (Figure 15) directs the agent to internally enumerate plausible candidate products and then select the single most informative clarifying question that would best differentiate among them.

The **POMDP** prompt (Figure 16) frames the interaction as a planning problem under uncertainty. Here, the agent is instructed to maintain a belief state over possible user preferences, evaluate the expected value of different candidate questions, and select the one with the highest information gain—even if its benefits only materialize in later turns.

Results are shown in Figure 8.

Prompt for Greedy Questioning

You are a product recommendation agent. Your goal is to find the best product for this user.

Context:

```
{context}
{feedback_context}
```

INTERNAL REASONING (do not share with customer):

- First, list all the possible products that you think the customer might like based on what you know so far.
- Then, think about what is the best question you could ask the customer to eliminate the most number of products from the list.

Your question should:

- Help you distinguish between the products you think the customer might like
- Focus on the most important decision factor that's still unclear
- Ask about preferences, needs, and requirements — NOT about specific products or product numbers

CRITICAL OUTPUT FORMAT (MUST FOLLOW EXACTLY):

QUESTION: [your question here]

Figure 15: Greedy prompt used to bias the recommender agent toward asking the most informative single question at each turn.

Prompts for Dynamic Programming (POMDP) Questioning

System Message: You are a product recommendation agent. Your goal is to find the best product for this user.

Context:

```
{context}
{feedback_context}
```

Think like a planner solving a **POMDP** with a **single terminal reward** from the score that the customer would assign to what you recommend. Note that you have `{questions_remaining}` turn(s) left.

1. **Maintain a belief state**—a probability distribution over possible customer preferences given past answers.
2. For each possible next question:
 - Predict how each possible answer will **update your belief**.
 - Estimate how that updated belief will affect your **final recommendation quality**.
 - Compute the **expected value of information (EVI)** for that question.
3. Choose the question with the **highest expected value**, even if it has no immediate payoff.

Customer Interaction: Ask the question that maximizes the expected value of information for your final recommendation.

CRITICAL OUTPUT FORMAT (MUST FOLLOW EXACTLY):

QUESTION: [your question here]

Figure 16: POMDP-inspired prompt that instructs the agent to choose the question with the highest Expected Value of Information.

A.5 MANUAL QUESTIONING TESTS

To verify that poor agent performance is not due to an unsolvable environment, we introduce manual questioning experiments where a human designs the persona interactions. Figure 17 shows an exem-

plar run in interactive mode, where it can be noted that carefully chosen questions enable recovery of the best product with zero regret.

An Example Manual Questions Experiment (Interactive Mode)

Running Manual Questions Experiment (Interactive Mode):

Persona: 25905

Category: Office Racks & Displays

=== Manual Questions Experiment (Interactive) ===

Persona: 25905

Category: Office Racks & Displays

Model: gpt-4o, Feedback: persona

Category Office Racks & Displays: Max score 82.5, proceeding

Q: How do you usually organize your desk at home or in the office?

A: I organize my desk by keeping it tidy and functional, with my laptop in the center, a notepad and pen to the right, and any necessary documents or files neatly stacked on the left.

Q: Do you prefer everything stored neatly in trays, or do you like quick vertical access to files and folders?

A: I prefer everything stored neatly in trays.

Q: How many papers, notebooks, or files do you typically keep on your desk at one time?

A: Usually around three or four.

Q: Do your kids or family members often use your desk space, or is it mainly your own?

A: It's mainly my own.

Q: Do you like darker neutral tones (black, brown, gray), or brighter finishes like gold or rose?

A: I like darker neutral tones.

Q: Do you want a compact organizer to save space, or a larger one with multiple tiers for capacity?

A: A compact organizer to save space.

Q: /done

Products in Office Racks & Displays: 26

Top 3 products by score:

1. SimpleHouseware Mesh Desk Organizer with Sliding D... (Score: 82.5)
2. Figella Leather Office File Rack Document Filing T... (Score: 80.0)
3. Febou 2 Pack Mail Organizer Desktop File Holder, 3... (Score: 80.0)

Final recommendation: Product 107921

Score: 82.5, Best: 82.5

Top1: True, Top3: True

Feedback: *Perfect! This is exactly what I was looking for. Great recommendation!*

Experiment completed successfully!

Chosen product rank: 1

Final regret: 0.0

Figure 17: An exemplar run of the manual questions experiment in interactive mode.

B ABLATION STUDIES

We discuss additional ablation studies performed using the dataset. Overall, SOTA models do not exhibit the ability to learn across turns or episodes.

B.1 VARYING USER FEEDBACK

Fixing the model to GPT-4o and holding all else constant, we evaluated the three feedback types described in Section 4.3 across 40 random seeds and 10 episodes. As shown in Figure 18, no

statistically significant differences were observed across feedback types, indicating that the form of feedback does not materially influence model performance.

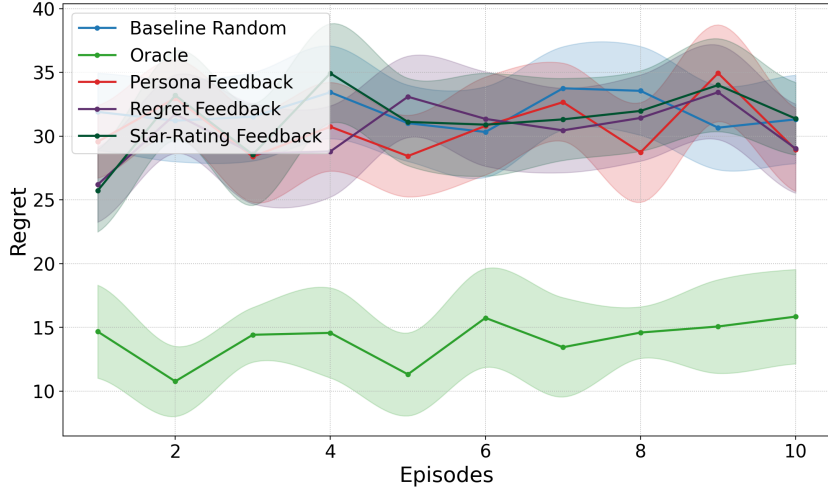


Figure 18: Feedback type did not improve model performance by any significant extent

B.2 PROVIDING REASONING PROMPTS

Fixing the model to GPT-4o and holding all else constant, we also experimented with prompting strategies, such as inserting reasoning prompts (e.g., Think Again) before the agent issued a recommendation or question. As shown in Figure 19, these prompting tricks did not lead to statistically significant improvements, and performance remained indistinguishable from the no-prompting baseline.

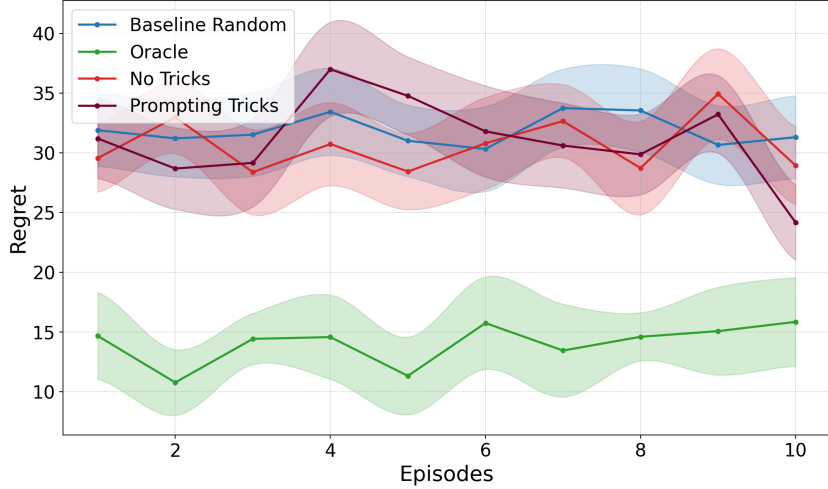


Figure 19: The presence and absence of prompting tricks did not impact model performance by any significant extent

B.3 HOW TO UTILIZE PAST EXPERIENCES

Fixing the model to GPT-4o and holding all else constant, we further examined whether providing agents with access to their own context summaries could improve performance. In this variant, the recommender agent generated a summary of each episode, which was then carried forward into

subsequent interactions. As shown in Figure 20, this approach produced no statistically significant improvement over the baseline.

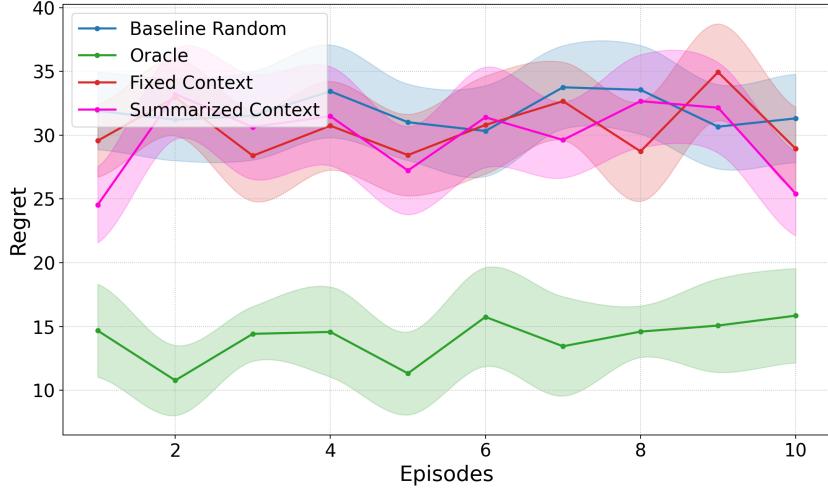


Figure 20: Context type did not improve model performance by any significant extent

B.4 LEARNING ACROSS DIFFERENT USERS

Using GPT-4o with all other factors held constant, we tested whether varying product categories, personas, or both across episodes influenced model performance. As shown in Figure 23, none of these variants produced statistically significant differences.

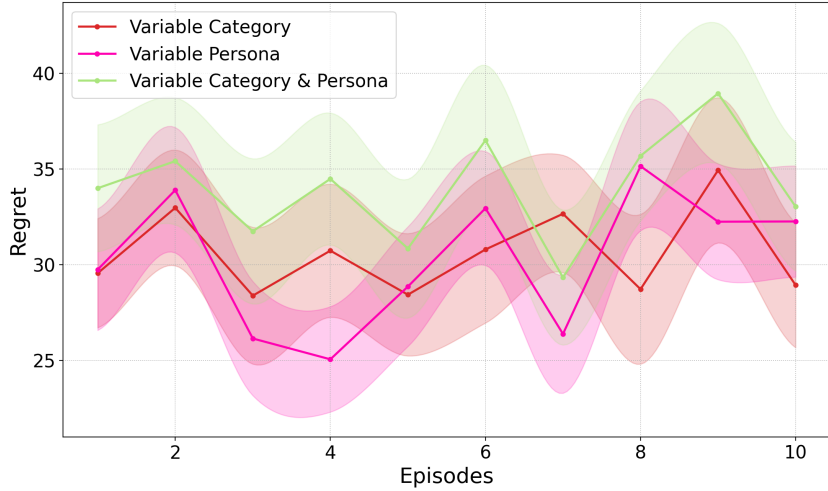


Figure 21: Whether there was variable personas, categories, or both did not impact model performance by any significant extent

B.5 ECE

We prompted the agent, at every recommendation turn, to report confidences for five binary targets: (1) chosen item is top-1 for the persona, (2) top-5, and (3–5) regret $\leq 5, 10, 20, 30$. For each target, we evaluated calibration per episode slice (Episodes 1, 5, and 10), pooling all turns across categories and seeds for that slice. We produced bin-wise accuracy-vs-confidence plots and summarized

misalignment with a single score. We ran the protocol under two prompting conditions, Baseline (no strategy) and DP-style planning, and display the full grid of episode $\times$ target results for each condition.

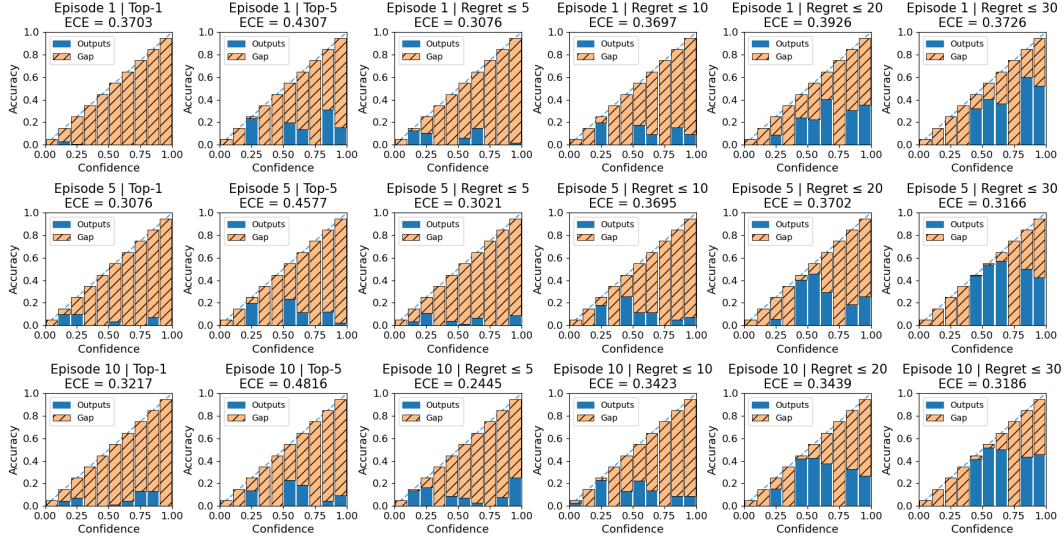


Figure 22: All ECE Runs using GPT-4o with No Strategy

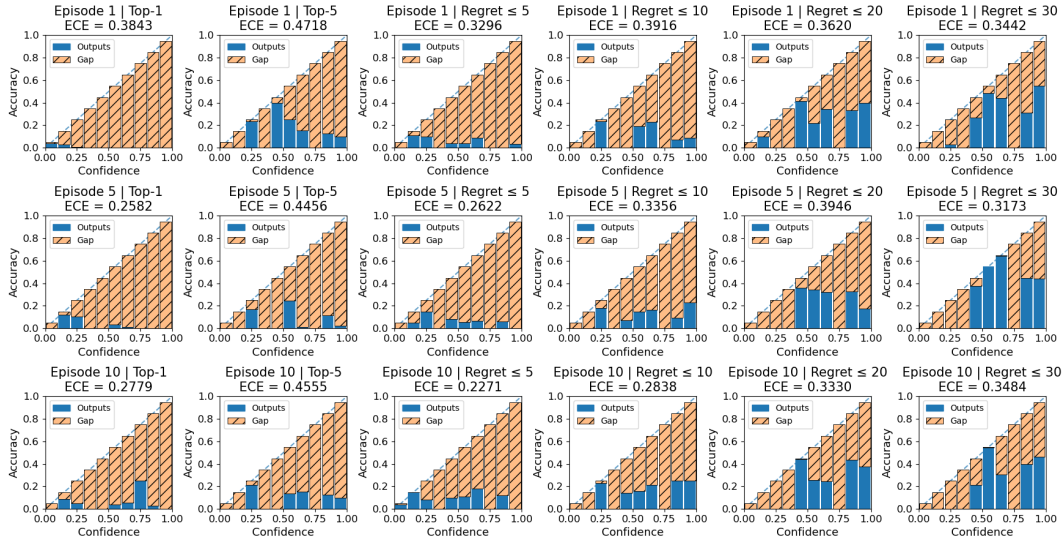


Figure 23: All ECE Runs using GPT-4o with DP-style Planning

C DATASET COLLECTION

Collecting Products. The Amazon Reviews dataset (Hou et al., 2024), consisting of 34 product categories, is set up such that one has to pick one of the 34 categories to sample products. Therefore, to preserve the original distribution of categories, we performed weighted random sampling over the categories. We provided the necessary scripts to draw more products in our codebase (Section 7). **Matching Product Categories.** The first is inconsistent naming conventions; for example, a product’s category path might be ...→Women → Shoes, Sandals, whereas our standardized path is ...→ Women’s Fashion → Women’s Shoes → Women’s Sandals. The second challenge

is structural discrepancy, where a product’s path omits an intermediate level, such as `CDs&Vinyl` $\rightarrow$ `Blues`, while the canonical path is `CDs&Vinyl` $\rightarrow$ `Music Styles` $\rightarrow$ `Blues`. To solve these issues, our algorithm employs a two-pronged approach at each step of the path traversal:

- **Fuzzy Matching for Naming Inconsistencies:** To resolve differing names, we apply a series of prioritized fuzzy matching heuristics. For instance, the substring detection heuristic is precisely what allows the algorithm to map a product’s simpler category `Sandals` to the more descriptive canonical term `Women’s Sandals`. Other heuristics, like word subset validation (mapping `Action Figures` to `Action & Toy Figures`) and normalization (handling case and suffix differences), ensure robustness against a wide range of naming variations.
- **Subtree Traversal for Structural Gaps:** To handle missing intermediate levels, our algorithm does not merely search the immediate children of the last matched node. Instead, it performs a breadth-first search through the entire subtree of descendants. This is how it bridges structural gaps. In the example `CDs & Vinyl` $\rightarrow$ `Blues`, once the algorithm successfully matches the `CDs & Vinyl` node, it then searches all descendants for a `Blues` node. It will find the correct `Blues` node even though it is a grandchild (nested under `Music Styles`), effectively “skipping over” the missing level in the product’s path data.

This matching enables us to successfully map about 70% of the sampled products to a valid path within the target hierarchy. Despite the flexibility of these heuristics, the overall assignment criteria remain strict: a product is considered successfully matched only when its entire category sequence corresponds to a valid, continuous path from the root.

Final Database. From the remaining category levels, we extracted individual category levels and flattened the hierarchical structure into a relationship database schema. The processed data was organized into three tables: (1) a products table containing product metadata, (2) a categories table, and (3) a product-category table implementing a many-to-many relationship between products and categories. From this database, we then pruned categories that were still too broad (e.g., `Cooking & Baking`) and would not be searched for by the typical user and niche categories that had fewer than 15 products. The final database consists of 71088 products, 2030 categories, and 100485 product-category links.

D KNOWN ISSUES WITH LLMs

Simulator Faithfulness The customer simulator’s intermediate responses are at times unfaithful to the persona’s true underlying interests. Across the experiments, two main patterns of unfaithfulness were observed. First, in cases of consistent unfaithfulness, some personas consistently feign interest in the presented product category during the dialogue, only to reveal their true, unrelated intent in the final feedback. For instance, a persona might claim they are buying for a friend but, in the feedback stage, reveal personal annoyances with the product. Second, inconsistent faithfulness was observed in other personas (e.g., the woodworking enthusiast, No. 2601), who sometimes truthfully state their interests but at other times actively deceive the agent by fabricating plausible but false needs, such as inventing a specific camera model they do not own. In all cases, the simulators are designed to provide misleading or incomplete information, forcing the agent to learn from indirect signals and negative feedback.

Impact of Reasoning Prompts and Behavior It is not clear if the models’ behavior materially changes with a reasoning prompt. Models, both with and without prompting tricks, sometimes ask a significantly higher volume of questions. This strategy can show signs of rigidity and inefficiency, as the model tends to ask many repetitive questions and relies heavily on a simple yes/no format. For instance, in the `Fabric Dyes` episode, the agent asks the exact same question three times consecutively. Similarly, models from the Gemini series exhibit their own specific rigid patterns. They frequently ask questions like, “Are you looking for a free option?” and then often proceed immediately to a recommendation. This behavior suggests the model is following a pre-determined conversational script rather than dynamically adapting to the user’s needs, showing no interest in asking more questions and prematurely ending the conversation.

The reasoning process, as observed in the `thinking_block`, consists of a short textual monologue. It typically summarizes the user’s explicitly stated needs from the dialogue and then outlines

a simple strategy for its next action, which is often a recommendation. The reasoning itself is sensible and logical based on the information the agent has at that moment. The agent correctly processes the user’s statements and devises a rational plan. The failures observed are not due to flawed reasoning but are a direct result of the agent reasoning from the intentionally false or misleading premises provided by the unfaithful simulator.

E LLM USE

LLMs were used to aid writing, as well as for finding some of the related works.