

Benchmarking In-context Experiential Learning Through Repeated Product Recommendations

Gilbert Yang^{*1} Yaqin Chen^{*2} Thomson Yen^{*1} Hongseok Namkoong¹

¹Decision, Risk, and Operations Division, Columbia Business School

²School of Mathematics (Zhuhai), Sun Yat-sen University

{gy2322, ty2531, hongseok.namkoong}@columbia.edu, chenyaq077@gmail.com

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 reason through the experiences they accrued. We exemplify the need for this 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 a benchmark for experiential learning and active exploration (**BELA**) that combines (1) rich real-world products from Amazon, (2) a diverse collection of user personas to represent heterogeneous yet latent preferences, and (3) a LLM user simulator powered by the persona to create rich interactive trajectories. We observe that current frontier models struggle to meaningfully improve across episodes, underscoring the need for agentic systems with strong in-context learning capabilities.

 Website  Code

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 and 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.

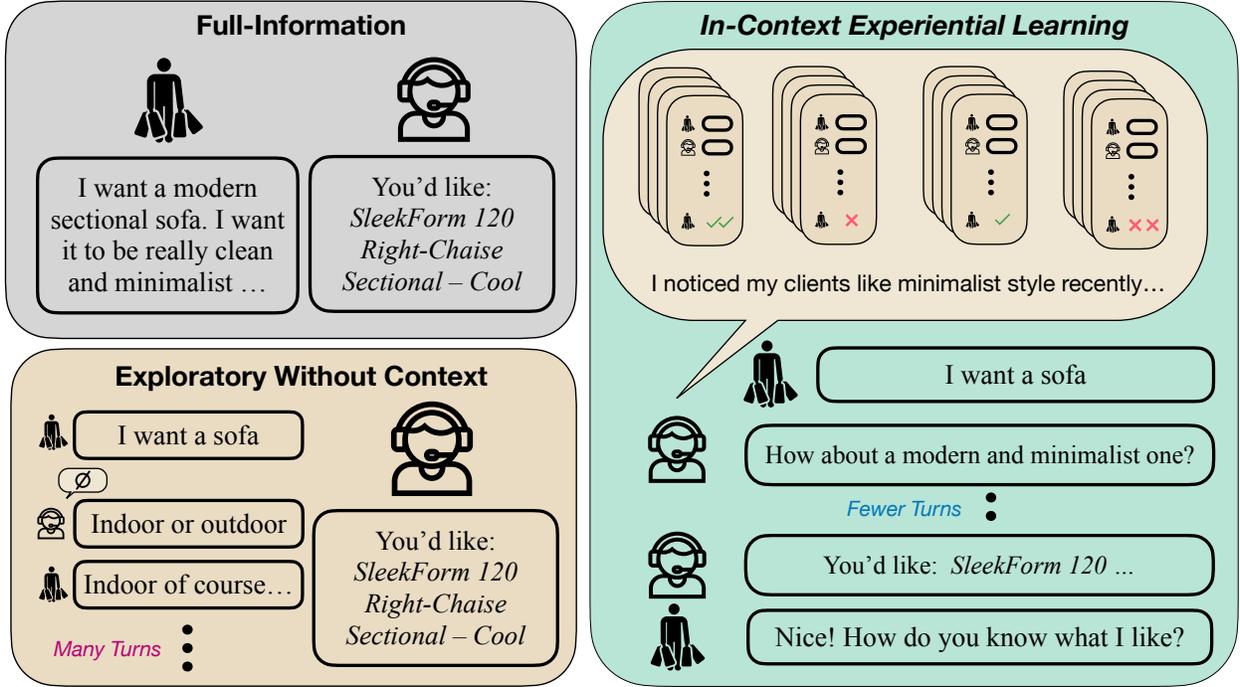


Figure 1: **Top Left.** Typical agentic benchmarks (e.g. Yao et al. [2024]) focus on settings where all information is provided initially, and the model is tasked with producing the correct answer in a zero-shot fashion. **Bottom Left.** Recent benchmarks (e.g. Li et al. [2024]) for LLM agents increasingly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

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 customers’ latent preferences through exploratory questions and iteratively refine recommendation based on ambiguous, text-based responses from the customers. 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.

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 agent that can actively

Examples of Recommendation Dialogues for In-Context Experiential Learning

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

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

discover user preferences by leveraging its experiences, strategically gathered over many interactions across multiple episodes. See Figure 2 for an example of our setting. In contrast to works on pluralistic alignment or uncertainty quantification that focus on a single interaction or episode [Castricato et al., 2025, Zollo et al., 2025, Li et al., 2024], we emphasize the agent’s ability to learn across multiple interactions and episodes (*experiential learning*). For example, in the personalization setting, we do not consider the ideal scenarios where all relevant customer preferences over the choice set are provided upfront. Instead, the agent must actively uncover and refine the parts of the customer’s preferences that are most relevant to the current choice set through iterative interactions, while leveraging knowledge about the customer accumulated from previous episodes involving different choice sets.

To instantiate this benchmark for experiential learning, 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 customer preferences over these products. The same customer-simulating LLM also serves as the backbone

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

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

of the interactive question-answering module, enabling agents to take strategic actions through which they acquire experience about their customers. The dataset we curated supports up to 2 billions unique choice set-persona pairs, each forming a rich multi-turn environment that enables the construction of diverse and challenging experiential learning scenarios. Furthermore, the underlying data-generation pipeline is inherently scalable, allowing for the continual expansion of the dataset as new products, choice sets, or personas are introduced.

Beyond serving as a benchmark for evaluating experiential learning capabilities, our dataset offers a versatile framework for studying recommendation systems. By combining real-world product data, diverse user personas, and interactive dynamics powered by LLM simulator, it enables systematic exploration of key challenges including user modeling, preference elicitation, and cold-start recommendation, to name a few. We believe this dataset stands as a valuable contribution in its own right, offering a realistic and extensible testbed for 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 measuring one-shot predictive capabilities with all relevant information provided, we focus on multi-episodic settings where the agent necessarily have to actively acquire experiences initially due to ambiguity. Key differences from prior benchmarks are summarized in Table 1.
- We develop a benchmark for **Benchmark for Experiential Learning and Active exploration (BELA)**, 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 product recommendation 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 from experiences, highlighting their inability to navigate 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 as RL environments: 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 environments, 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 in adapting and improving 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 As A 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 preferences unique to the customer. We refer to this group of functionally equivalent products as a *choice set* (\mathcal{S}). The recommender agent must discover

customer’s preferences over \mathcal{S} by engaging in *multiple turns* of queries: asking targeted questions, receiving, and interpreting the customer’s responses. 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 factor* (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 $e \in \{1, \dots, E\}$, 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

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.

a highest score, $y^* = \max(\{y_a \mid y_a = h(p_a, c, \mathcal{S}) \forall p_a \in \mathcal{S}\})$, and y_r is the score of the recommended product.

4 BELA: A Benchmark for Experiential Learning and Active Exploration

To instantiate the aforementioned benchmark for experiential learning, our product recommendation dataset 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.

A few desiderata are essential for a meaningful evaluation of experiential learning capability. First, the underlying sets of both modules (\mathcal{C} and \mathcal{S}) need to be scalable and diverse to support nuanced scoring and a wide variety of Θ and question-answering dynamics. Second, the customer module must maintain internal consistency between θ_e and $(o_{e,t}, f_e)$; naturally, since the latter constitutes the experiences accessible to the agents, it must reliably inform the agents about θ_e . In contrast, note that whether \mathcal{C} distributionally reflect the preferences of mirrors real-world customer populations is not a key requirement for the benchmark’s primary purpose. For use cases where one wishes to assess an agent’s recommendation quality for customers from a particular industry or company, our framework readily supports simulation of customized instantiations of \mathcal{C} .

4.1 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

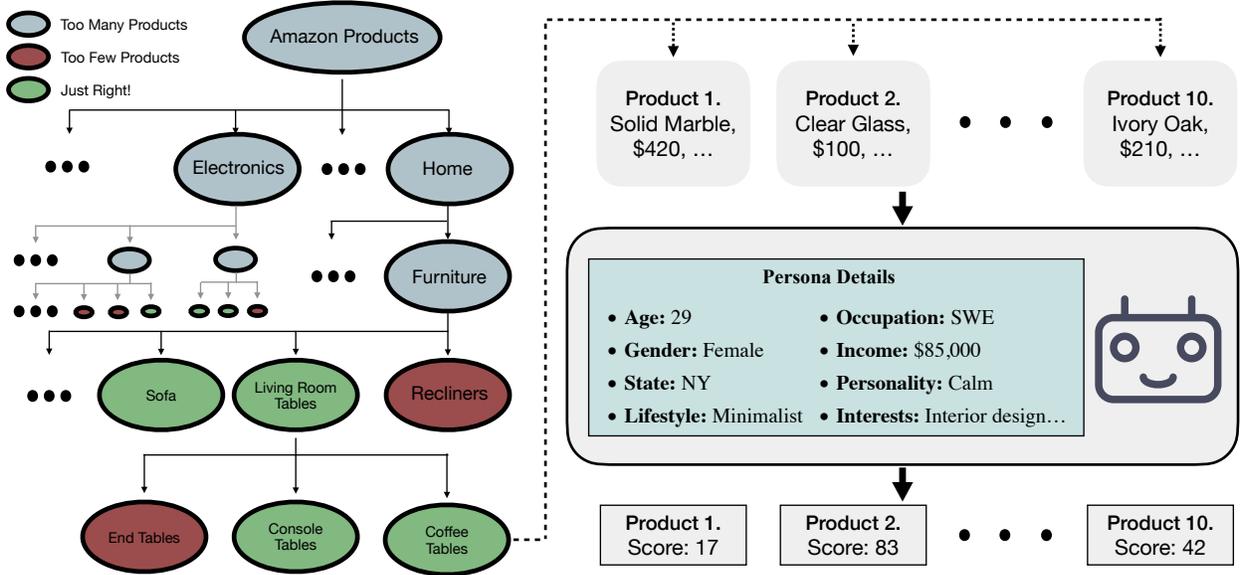


Figure 4: We use a predefined tree of categories [ASINSpotlight, 2023] and filtered the ones unsuitable as choice sets. We then score the products within each choice set with a persona-simulating LLM.

[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 F for additional details.

Then, we extract \mathcal{S} as a subset of predefined categories, leaving only 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.

Through the novel filtering schemes, we construct a large collection of products and choice sets, consisting of 71K products and 2K choice sets. Since some categories share products, the dataset contains in total 100K product-category pairs.

On scalability of available choice sets, our pipeline naturally supports expansion by sampling additional products from Hou et al. [2024] or other large-scale datasets. 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.

4.2 Persona-Based Customer Simulation

To power the customer module, we first need to curate a pool of persona descriptions that define the simulated customers. Next, to enable the evaluation of recommendation quality in regret, we

generate each persona’s preference over a given choice set ($y_a = h(p_a, c, \mathbf{S})$). Finally, we have a choice over the types of feedback f_e to be provided at the end of a produce recommendation episode.

Persona Curation. 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.

Preference Generation. For preference generation ($y_a = h(p_a, c, \mathbf{S})$), 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 \mathbf{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 \mathbf{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 respond to agent’s questions as the persona when shopping within a given product category. 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. We provide additional experiments for the consistency of module in Appendix E.

4.3 Dataset Robustness

To prevent unrealistic scenarios (e.g., an individual with no musical interests shopping for pianos), we exclude choice set-persona pair in which the highest product score is below 60. Using this filtering procedure, an average of 3.75 scenarios were skipped across 10 randomly drawn choice set-persona pairs.

Furthermore, 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 C. 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. Using the information elicited by expert questionings, 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 consider three sequential recommendation settings that exemplify experiential reasoning: (a) same customer across episodes with different choice sets each episode, (b) same category across episodes with different customers each episode, and (c) different choice sets and customers each episode. Of course, the rich sets of personas and choice set available in BELA can support many more settings than the ones considered in this study.

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 Do Models Learn From Experiences?

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 under setting (a). Across these experiments, we use the Free-form Text feedback.

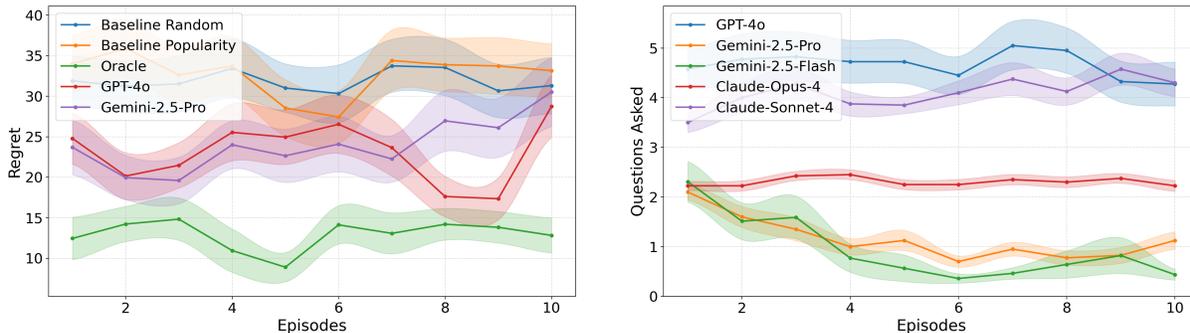


Figure 5: **LEFT**: Models perform better than random, but significantly worse than the oracle with no learning across episodes **RIGHT**: Gemini models tend to ask fewer questions in later episodes

As shown in the left panel of Figure 5, while the models perform better than simple baselines, all significantly underperform the oracle baseline. In particular, no models exhibit episode-over-episode improvements, highlighting the difficulty for current SOTA models to leverage experiences about the customer obtained in prior episodes.

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

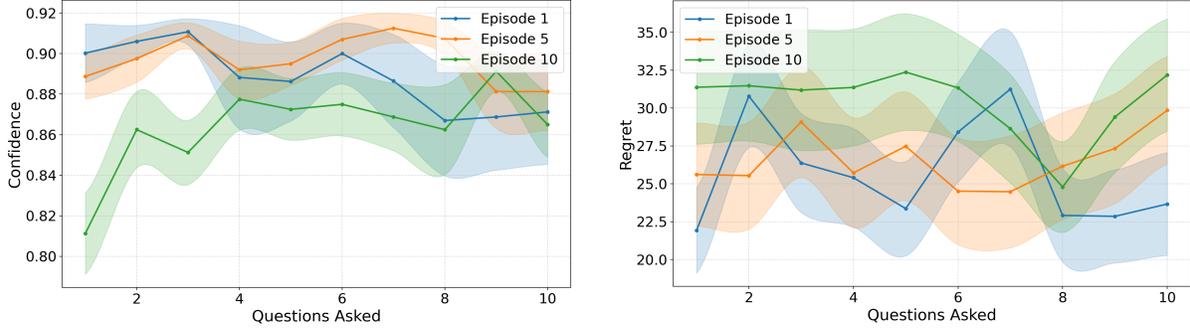


Figure 6: **LEFT:** GPT-4o maintains the same level of top-1 confidence across episodes. **RIGHT:** GPT-4o performs no better across episodes nor over questions asked.

declines over episodes. We highlight this as a major deficiency in the behavior patterns of the SOTA models.

Overall, across various recommendation settings, feedback types, prompts etc., the models fail to demonstrate experiential learning capability. We provide the results of all ablation studies in Appendix A.

5.2 Can Models Comprehend Their Uncertainties?

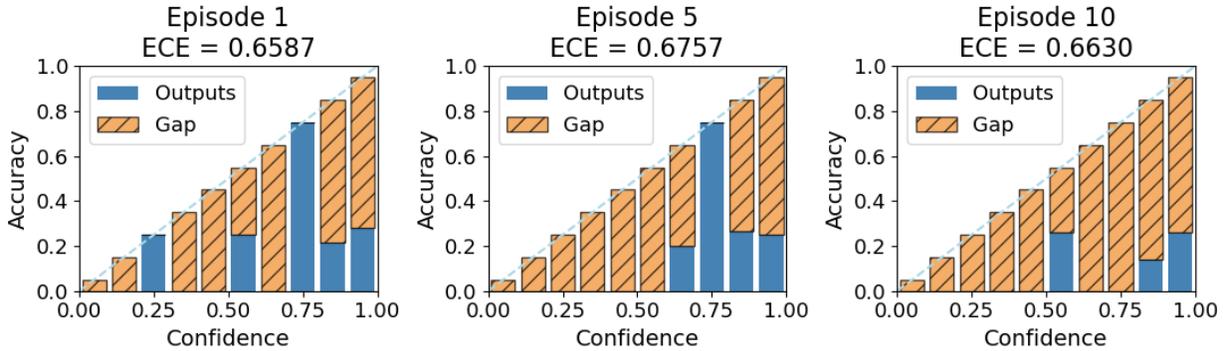


Figure 7: ECE plots for GPT-4o. The model is clearly poorly calibrated. Results are shown for confidence on regret within 10 across all turns.

Next, we examine whether agents can better quantify their own uncertainties from accrued experiences. To assess this, in the otherwise same setting as in Section 5.1, we prompt GPT-4o to output six kinds of confidences on: (1-2) whether the chosen item would fall into top-1 and top-5 favorite products of the customer, and (3-6) whether the realized regret would fall into the expected regret within 5, 10, 20, and 30 points. In all experiments, the agent was run for 10 episodes under setting (a), 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 and confidence at every turn.

Figure 6 shows that the model seems to grow more confident in episode 10, however, this does not match the agent’s stagnant performance. We also find that the model struggles to estimate its uncertainties for all episodes (see Figure 7). We provide complete experimental results for all confidences considered in Appendix B, where the model consistently fails to calibrate from experiences.

5.3 The Ideal Learning Trajectory

To demonstrate that models can sometimes successfully learn from experiences in our setting, we construct an *ideal* learning trajectory. To do so, for each k -th episode, we manually search for one high-quality episode that (i) results in lower regret compared to the previous episode and (ii) contains questions that target meaningful aspects of the persona. We then freeze this accepted episode and use it as the starting context for the next episode. Repeating this process for all 10 episodes, we find a trajectory consisting of increasingly insightful episodes with monotonically decreasing regret: $45.0 \rightarrow 22.5 \rightarrow 20.0 \rightarrow 20.0 \rightarrow 17.5 \rightarrow 12.5 \rightarrow 8.5 \rightarrow 6.5 \rightarrow 2.0 \rightarrow 0.0$. This decreasing regret illustrates how a capable agent *should* leverage previously accrued experiences to better recommend products.

Examples from the Ideal Trajectory of Experiential Learning

<p>Ep. 1: AGT : Do you prefer solar-powered or low-voltage lights? CUST: I prefer solar-powered lights because they align with my sustainability values. AGT : Do you prefer warm white or color-changing lights? CUST: Warm white—they create a cozy and inviting atmosphere.</p>	<p style="text-align: center;">⋮</p> <p>Ep. 6: AGT : Would you consider a product outside your price range if it meets your sustainability and quality criteria? CUST: Yes, I would—quality and sustainability matter more to me than price. ⋮</p>	<p style="text-align: center;">⋮</p> <p>Ep. 10: AGT : Do you prefer mules and clogs made from natural materials like leather or canvas? CUST: Yes—natural and sustainable materials fit my values best.</p>
---	--	--

Figure 8: **Ideal Trajectory of Experiential Learning.** An illustration of how an agent should evolve across episodes in the personalization setting.

Figure 8 visualizes representative excerpts from Episodes 1, 6, and 10. In Episode 1, the agent asks shallow feature-comparison questions such as “solar-powered or low voltage?”—capturing surface-level preferences but missing deeper latent values. By Episode 6, the agent begins to model higher-level attributes such as the user’s value hierarchy (e.g., sustainability vs. price), signaling a transition from inquiring specific attributes to modeling implicit preferences. By Episode 10, the agent has internalized the persona’s consistent emphasis on natural materials and sustainability, and begins posing *confirmation-style* questions that reflect an internal latent model of what the persona cares about.

6 Conclusion

We argue for a shift in focus toward measuring agents’ ability to reasoning through uncertainties by leveraging their experiences. To support this goal, we introduced **BELA**: a flexible interactive recommendation dataset designed to evaluate in-context experiential learning capabilities. Beyond serving as a benchmark for agent evaluation, we believe our dataset can be a valuable resource for the broader recommender systems community.

Our experiments reveal that SOTA models still fail to learn effectively from experiences, struggle to calibrate their uncertainties and, as a result, fail to proactively ask informative questions. By uncovering these limitations, we hope to motivate future research on experiential learning, which is essential for building agents capable of reliable decision-making under real-world uncertainties.

References

- 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>.
- 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>.
- 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, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf.
- 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, editors, *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.
- 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, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 3008–3021. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1f89885d556929e98d3ef9b86448f951-Paper.pdf.
- 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 Askill, 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, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong 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, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, 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>.

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>.

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>.

- David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 1, 2025.
- 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.
- Louis Castricato, Nathan Lile, Rafael Rafailov, Jan-Philipp Fränken, and Chelsea Finn. PERSONA: A reproducible testbed for pluralistic alignment. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert, editors, *Proceedings of the 31st International Conference on Computational Linguistics*, pages 11348–11368, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL <https://aclanthology.org/2025.coling-main.752/>.
- Thomas P Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. Person-LLM: Tailoring LLMs to individual preferences. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=2R7498e2Tx>.
- 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.
- ASINSpotlight. Amazon Categories List CSV, 2023. URL <https://www.asinspotlight.com/amz-categories-list-csv>. Accessed: 7 September 2025.
- 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>.
- 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>.
- 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, editors, *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 258–274, Sheffield, UK, June 2024. European Association for Machine Translation (EAMT). URL <https://aclanthology.org/2024.eamt-1.22/>.
- 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, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1777–1788, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1165. URL <https://aclanthology.org/P18-1165/>.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, 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>.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson,

- 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>.
- 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>.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shrutu 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>.
- 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, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 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/>.
- 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.

- 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>.
- 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=UeB3HdRhda>.
- 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>.
- 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>.
- 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, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11279–11298, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.775. URL <https://aclanthology.org/2022.emnlp-main.775/>.
- 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>.
- 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*, page 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>.
- 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.
- 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*, page 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>.
- 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*, page 720–730, New York, NY, USA, 2023. Associa-

- tion for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3614949. URL <https://doi.org/10.1145/3583780.3614949>.
- 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, editors, *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 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/>.
- 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.
- 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*, page 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>.
- 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=EMrnoPRvxe>.

A Ablation Studies

In this section, we discuss additional ablation studies performed using the dataset. Overall, SOTA models do not exhibit the ability to learn across episodes.

A.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 9, no statistically significant differences were observed across feedback types. Note that the Regret feedback type represents a less noisy source of information for the models. The fact that no models seem to utilize the more informative feedback indicates the models are not able to learn from customer feedback.

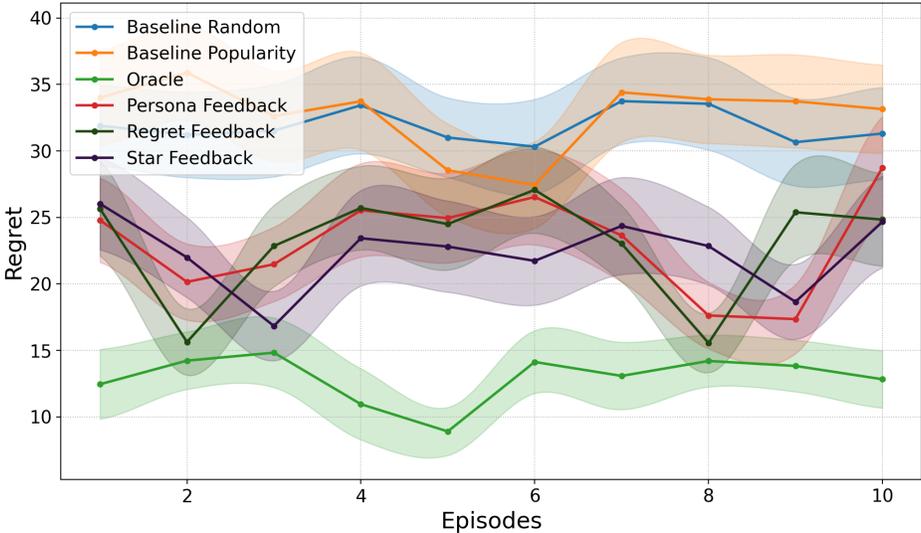


Figure 9: Feedback type did not improve model performance by any significant extent

A.2 Varying Prompts

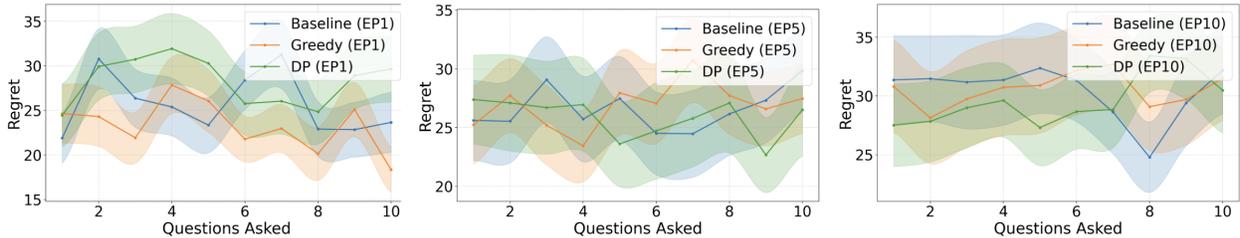


Figure 10: Performance of different prompting strategies across episodes (from left to right) with GPT-4o. The experiments were run over 60 seeds.

We investigate whether explicit prompting strategies could encourage more deliberate planning in recommendations. We consider three variants with explicitly instruction for the agent (GPT-4o) to (1) ask questions for T turns (**No Strategy**), (2) act greedily (**Greedy**), and (3) act as a DP-optimal planner in a POMDP (**DP**). We provide the prompts in Appendix D.4.

As shown in Figure 11, these prompting tricks did not lead to statistically significant improvements, and performance remained indistinguishable from the no-prompting baseline.

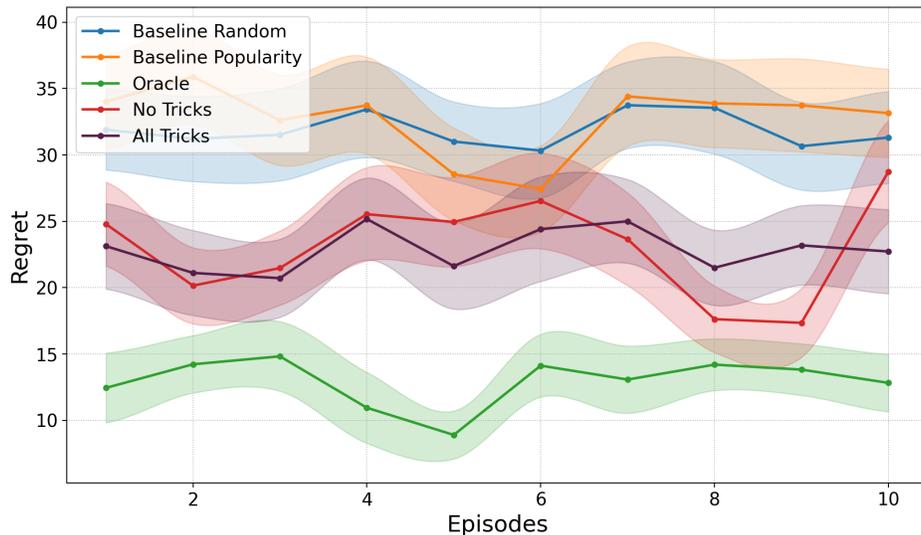


Figure 11: The presence and absence of prompting tricks did not impact model performance by any significant extent

A.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 12, this approach produced no statistically significant improvement.

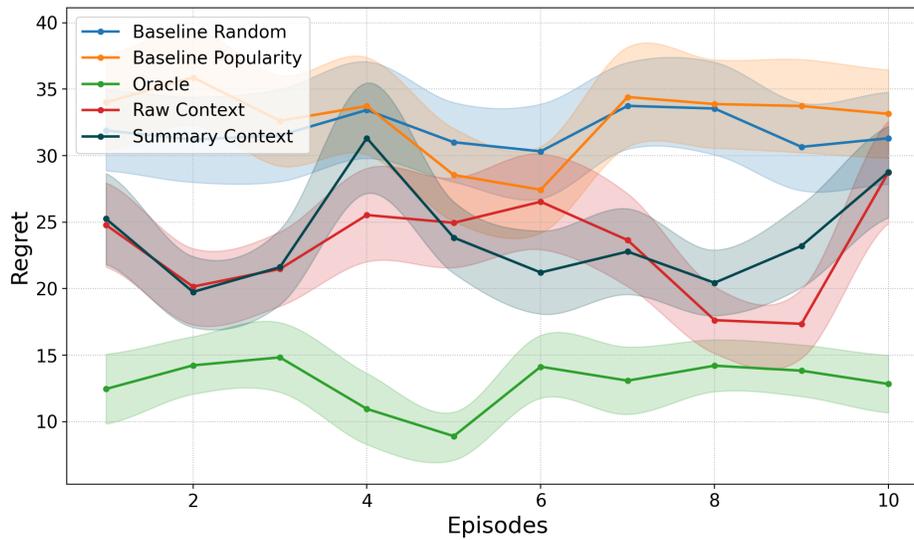


Figure 12: Context type did not improve model performance by any significant extent

A.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 13, none of these variants produced statistically significant differences.

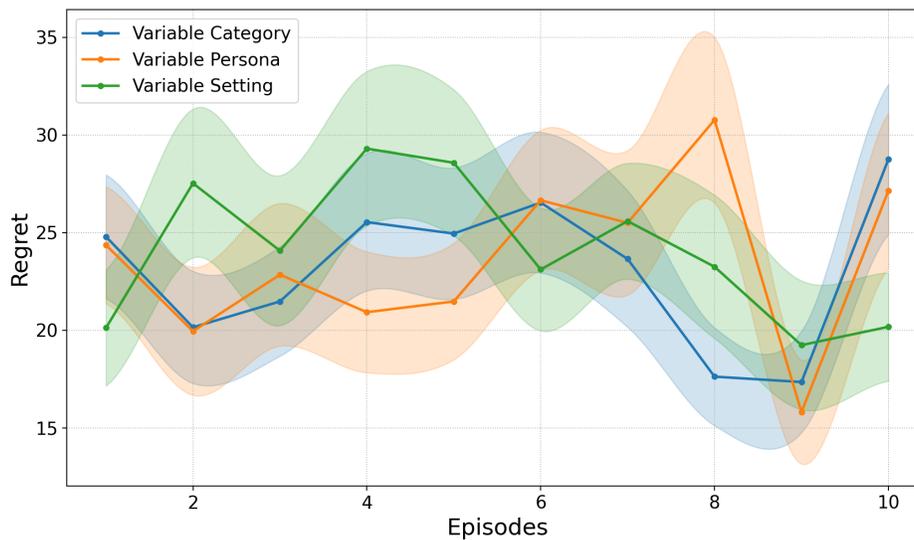


Figure 13: Whether there was variable personas, categories, or both did not impact model performance by any significant extent

B Model Confidence

As mentioned in the main text, 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 experiments with Baseline (no strategy) and DP-style planning prompts mentioned previously, and display the full grid of episode \times target results for each condition.

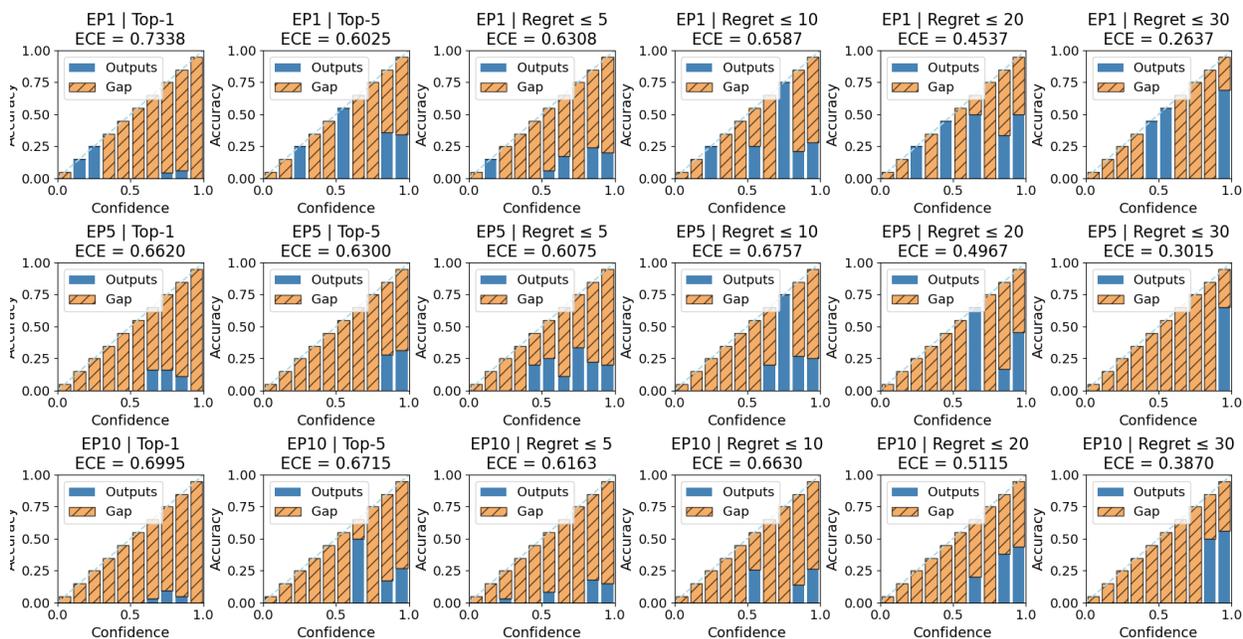


Figure 14: All ECE Runs using GPT-4o with No Strategy

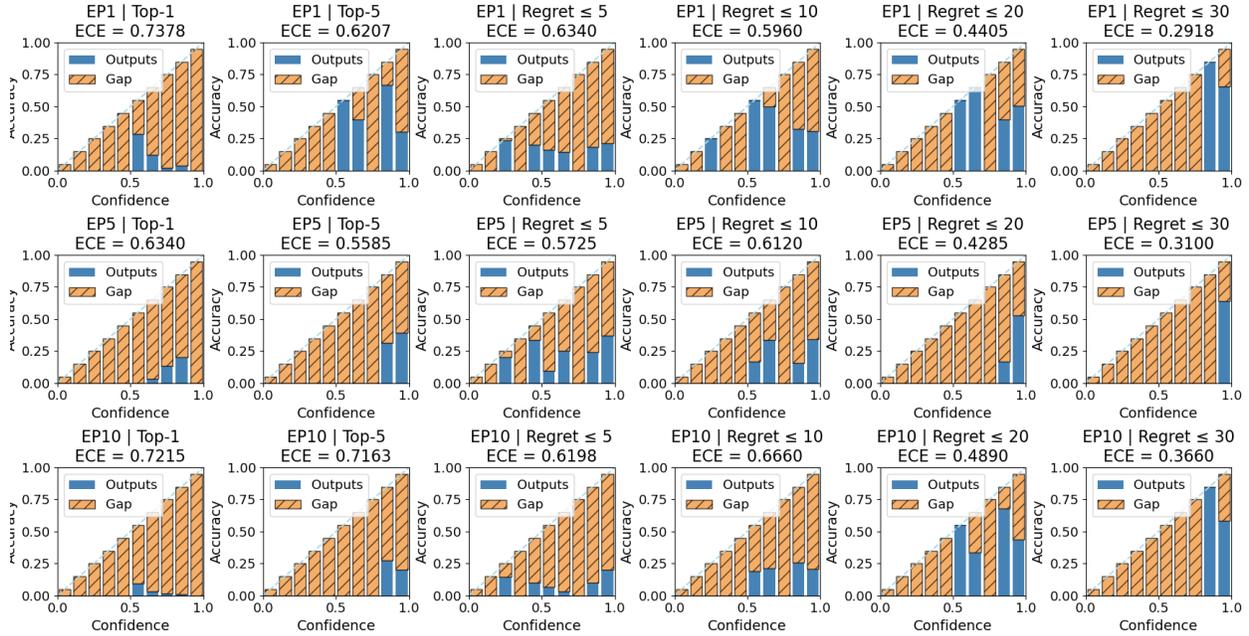


Figure 15: All ECE Runs using GPT-4o with DP-style Planning

C 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 16 shows an exemplar 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 16: An exemplar run of the manual questions experiment in interactive mode.

D Prompts

D.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 17: Prompt used to instantiate the oracle baseline.

D.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 in-  
clude any other keys or text."
```

(b) Response Simulation

```
"persona_des" : "..."  
"question"    : "..."  
"instructions": "You simulate a user with the  
given persona description. An-  
swer strictly as this persona  
would: – Only answer the ques-  
tion asked. – Do not restate  
persona or add rationale. – If  
a choice is requested, give one  
choice only. Return the answer  
as plain text."
```

Figure 18: 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 19: Prompt for eliciting naturalistic persona feedback following a recommendation. The tone is adjusted based on the quality of the recommendation, estimated through regret.

D.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 20, 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 21.

To test whether explicit reasoning instructions improve performance, we append a chain-of-thought style enhancement to the baseline prompt, shown in Figure 22. These prompt designs define the controlled conditions under which we ablate model behavior, with results reported in Appendix A.

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 20: 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 21: 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 22: Chain-of-Thought (CoT) enhancement appended to agent prompts to test the reasoning process.

D.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 23) 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 24) 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 10.

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 23: 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 24: POMDP-inspired prompt that instructs the agent to choose the question with the highest Expected Value of Information.

E Additional Experiments for Consistency of Customer Module

A central requirement of our benchmark is that the customer simulator behaves *consistently*—both with itself across repeated queries and with its own persona specification. We evaluate these two properties separately.

First, to measure how deterministically the simulator answers a fixed question, we conduct a $10\times$ replication test. Whenever the agent asks a question, we pause the trajectory and query the persona LLM (GPT-4o) independently 10 times, resetting the model at each run. All 10 answers are then evaluated by Claude-Sonnet-4, which identifies a *majority position* and counts how many responses align with it. Across 109 questions (1,090 responses), we find:

Aligned: 1,029 (94.4%), Misaligned: 61 (5.6%).

Manual review shows that nearly all “misaligned” cases reflect minor numeric or stylistic variations. For instance, “\$45” vs. “under \$50”, rather than substantive differences in preference. Because the evaluator treats these as distinct positions, the measured 94.4% rate represents a lower bound on the true semantic consistency of the simulator.

Next, we additionally assess whether simulator responses remain faithful to each persona’s description. For each of the 327 agent questions in 10 trajectories, Claude-Sonnet-4 classifies the simulator’s answer as: *Aligned*, *Neutral*, or *Conflict*. The raw breakdown is:

Aligned: 244 (74.6%), Neutral: 63 (19.3%), Conflict: 20 (6.1%).

However, manual inspection revealed that all 20 “Conflict” labels were false positives. The evaluator tended to flag contradictions based on the *absence* of certain hobbies, unwarranted assumptions (e.g., about what activities a persona “should” enjoy), or differences in specificity rather than actual logical conflict. No genuine contradictions with persona profiles were identified.

F 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.

Matching Product Categories. The first challenge 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** → **Blues**, while the canonical path is **CDs&Vinyl** → **Music Styles** → **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** → **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.

G Additional Models’ Failure Modes

In this section, we discuss some notable failure modes of the considered models.

Repetitive Questions. A recurring issue with some models 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 reaching a 20-turn limit. This was particularly evident with GPT-4o in all three experiments. Both Gemini models did not have this issue, while both Claude models showed repeated questions only under the variable setting experiment mode.

Failure to Memorize Requirements. Occasionally, explicit preferences stated by the customer are ignored. The final product chosen by the recommender agent can go against the stated preferences directly. This shows that SOTA models still struggle to respect specifications provided in context.

Model Specific Questions. 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.