VISA RESEARCH / OCTOBER 2026



Can an AI agent shop with you?

Visa Commerce Benchmark

VISA

Visa Commerce Benchmark

Can an AI agent shop with you?

ABSTRACT

Shopping agents are beginning to make purchases on behalf of consumers. Commerce Benchmark evaluates whether a language model acting as a shopping assistant can take a shopping request described in natural language and arrive at a product that satisfies all the customer's requirements, if it were to exist. Each customer has a unique journey called an episode, and the benchmark contains a growing list of episodes across different customer archetypes. Episodes end when the customer selects or declines an offer. In this paper, we start by describing the objectives of this benchmark followed by the structure of the world. Commerce Benchmark is designed to be a private benchmark, therefore episode design is described only at an abstract level with one episode discussed in full detail. Finally, we outline our approach to scoring and discuss how episodes are made difficult and how failures are diagnosed.



SEE IT IN ACTION.

Watch shopping assistants work through an episode, turn by turn, at visapaymentsfrontier.io

IN THIS PAPER

01	Objectives	03
02	Benchmark design	06
03	Evaluation	11
04	Calibrating difficulty	14
05	Evaluation practice and outlook	16



01 / OBJECTIVES

Due to the breadth of commerce, how a customer shops has a large variance across categories, and with it come unique challenges. A common pattern is customers rarely begin their journey with prescriptive detail about what they want. Often, the journey begins with a problem the customer faced but cannot diagnose the root cause of. For example, they might have noticed that videos from their child's phone look washed out on their television but fine on a relative's. This makes it the agent's job to figure out that the phone records in an HDR format their television does not support, making that format a critical feature to have in the one the customer is shopping for. It is also very common for a person to be shopping on behalf of others in the household. Since it is possible they do not have a complete list of requirements, the agent should make the customer do actions such as asking their partner or taking a measurement.

While in-person shopping typically comes with the assistance of a salesperson, in online shopping, the customer is largely "alone". A typical flow involves comparing product prices by hand across multiple retailers before searching for reviews of said products. Reviews involve sifting through websites loaded with search-engine optimized content that masquerade as neutral while accepting sponsorships from some of the brands they write about. Forums such as Reddit might have content from real users but their reliability is uneven.

Agentic commerce promises to both shorten the cycle and improve the quality of shopping. Models are capable of drawing on broad world knowledge and can ask customers relevant questions that help narrow down products quickly. Commerce Benchmark measures how well a model performs this role.

The benchmark is designed around the following objectives:

01 / UNDERSTAND THE NEED

Determining what works best for a customer involves answering questions such as how the product will be used, whether it will be shared with other people in the household, which restrictions apply (e.g., available space, compatibility with products they already own, safety, budget), and which store, offer, or membership gives the best overall result.

02 / ASK WHAT IS MISSING

Customers often do not know which facts bear on the purchase. The model is expected to ask how and where the product will be used, including questions whose relevance the customer would not anticipate, such as what is plugged into the television.

03 / EXPLAIN WHAT MATTERS

Some decisive facts are general knowledge that a particular customer has not encountered. A kitchen appliance bought for a vacation home in Spain must operate on 230 volts, and an induction cooktop will not heat aluminum pans. A capable assistant raises such points before they lead to a return or a safety concern.

04 / NAVIGATE THE MARKET

A product is usually sold by several stores at different prices, with different delivery times and return terms. The same product SKU across retailers might come with a retailer-specific description. The model must interpret equivalent terms and marketing figures correctly to compare products, and include shipping, taxes and any membership in the total price.

05 / ADAPT

As new information arrives into the conversation, the customer may change their mind. A customer's journey might be spread over multiple days, during which time prices and stock may have shifted. The model should evaluate staleness of the information it gathered, and alert the customer to any changes in the market.

06 / PROTECT THE SHOPPER

The model should act in the customer's interest when parts of the market do not. This includes recognizing untrustworthy stores and manipulative listings or reviews, and avoiding actions that put the customer at risk.

Each objective listed above is an optimization problem that models are capable of solving in isolation. The difficulty of a task rises as requirements compound and there might be tradeoffs to balance. We measure these objectives through shopping scenarios that we call episodes. Each episode combines several of the challenges above, as real purchases do. This keeps the scenario realistic and makes each failure easier to trace to its cause (Section 4). The answer must also be reached through an understanding of the customer. A selection that happens to match the answer key, without the model having established the facts that make it correct, is scored as a failure (Section 3).

Every episode can be summarized as a single question:

Starting from an incomplete request, can the model reach a selection that is supported by the customer's requirements?

Existing evaluations of shopping agents generally provide the model with a well-specified goal, whereas our benchmark begins while the goal is still incomplete. Two scope decisions apply. First, an episode ends with an informed decision. The customer either commits to a specific offer, which we call a selection, or declines to buy. The benchmark stops at that point, before checkout. Payment, order placement, and fulfillment are outside the current scope. Second, the benchmark compares models. All models use the same tools, instructions, market, and simulated customer, so differences in outcome can be fairly attributed to the model.



02 / BENCHMARK DESIGN

2.1 The real-world market

Before describing our market, we summarize the properties of real online shopping that make it difficult:

- There is no central catalog. Shoppers discover stores through a search over web pages that include store pages, buying guides, review sites, and forums. Some pages are dated or out of date, and some refer to discontinued products that survive only as inactive listings.
- A product is typically available from several stores with different prices, conditions, and return terms. Some stores offer a lower price that requires a paid membership.
- Search results show summaries that tend to present products favorably. The information that determines whether a product is suitable is often found only on the full listing or in the seller's policy.
- Marketing copy may overstate a product's capabilities.
- Reviews are numerous and uneven. Most say little about the attributes that decide whether a product is suitable, and the few that do are easy to miss.

Our version of the market is built to reflect these same dynamics. We mimic real product listings and published seller policies. Frontier models have substantial knowledge of consumer products, and a fictional catalog would put that knowledge in conflict with the environment. Real products allow us to test how well a model applies what it already knows, and bring the evaluation closer to the conditions of real deployment.

Reviews in our market follow the statistical properties of real review collections. Most are short, many concern delivery or packaging, and some contain typos or a star rating that disagrees with the text. Few address the attributes an expert would examine.

When a review carries a fact that matters for an episode, it is written in the same register as the others and can be found with search terms a shopper would plausibly use, but it is not tagged, highlighted, or ranked above other reviews.

A review noting that a product lacks a feature the customer needs may sit between a complaint about the delivery and a five-star review that reads "love it".

A key differentiator is the fact that our market is controlled, meaning it is isolated from the live web. Contents are fixed in advance, so every model sees the same market for fair comparison and every run can be replayed. Page retrieval is also an engineering problem that search providers and merchants are already addressing with solutions such as WebMCP. The benchmark targets the judgment involved in deciding what to look for and how to interpret it. The shopping assistant works with the data available in the market, much as a person shopping online works with what stores publish.

Data sources. Product specifications, prices, seller policies and marketing copy in our market are drawn from manufacturers' and retailers' public product pages, and some performance measurements from published third-party product reviews. Some episodes contain product listings and customer reviews taken verbatim from the Amazon Reviews 2023 dataset,[1] which we also used to measure the properties of real review collections that our own reviews follow. All other reviews and web pages in the market were written for the benchmark. Brands and retailers are named so that listings resemble those a shopper would see; the benchmark is not affiliated with or endorsed by any of them.

2.2 Episode structure

An episode is a shopping conversation between two language models with separate contexts, set in the fixed market. The model under test acts as the shopping assistant, an agent that works for the customer and has research tools over the market. Another model plays the customer. Figure 1 shows the parts of an episode and what passes between them.

Each episode is defined by five components:

- An opening request that describes the customer's situation in their own words.
- A private dossier of the customer that contains the information relevant to the current episode, such as requirements, budget, owned products, mistaken beliefs, and so on. Most of these facts start locked: a fact enters the context of the customer only when the shopping assistant asks the question or requests the action that releases it.

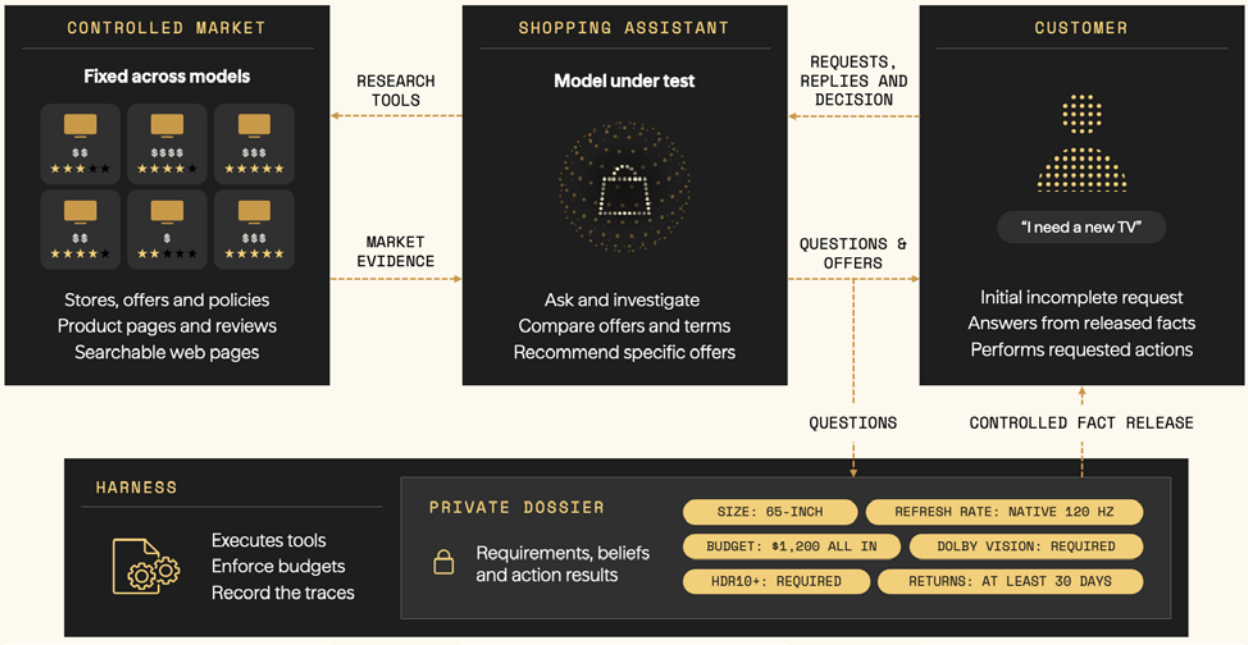


FIGURE 1
Overview of an episode. The shopping assistant, the model under test, researches a market that is the same for every model and talks with a simulated customer. The harness holds the customer's private dossier and releases a fact to the customer only when the assistant's questions or requested actions call for it. It also executes tool calls, enforces the episode's limits, and records the trace. The dossier shown is from the episode in Section 2.3.

1. Y. Hou, J. Li, Z. He, A. Yan, X. Chen, and J. McAuley. Bridging Language and Items for Retrieval and Recommendation. arXiv:2403.03952, 2024.

- Customer actions, such as measuring a space or consulting a partner, which the shopping assistant can request and only the customer can carry out.
- A controlled market of stores, offers, seller policies, reviews, and a small searchable collection of web pages about the products, which is identical for every model and can be replayed exactly.
- An answer key: the set of offers that would be a correct selection, computed from the market and the dossier. Neither model can see it, and it is only used in the evaluation.

An episode terminates when the customer selects a specific offer or declines, or when the conversation ends without a decision. The three outcomes are recorded separately.

NOTATION. We write Ω for the set of offers in the market. The dossier is a set of facts D, and each fact carries a release condition that describes the question or action that releases it. We write $D_t \subseteq D$ for the facts released by turn t, so that D_0 holds the facts that are not locked. Each requirement is a check $\phi_i : \Omega \to \{0, 1\}$, for $i = 1, \ldots, m$, that an offer either passes or fails. The episode ends with a decision d, which is either an offer in Ω or decline. In the terms of decision theory, an episode is a partially observable decision problem. The hidden part of the state is the dossier together with the listings the assistant has not yet opened, the assistant observes tool results and the customer's replies, and its policy maps the conversation so far to its next message or tool call.

2.2.1 The shopping assistant

The shopping assistant works with the customer through conversation, using the research tools to navigate the market. At the start of an episode it receives a fixed set of instructions (the system prompt) and the customer's opening request. The instructions describe its tools and how to present offers, and name the stores in the market and their membership programs, so that asking about a membership tests judgment rather than a guess that memberships exist. The instructions are identical for every model.

Once the conversation starts, the assistant can act in four ways:

- **Talk to the customer.** Ask questions, request customer actions such as a measurement, and present offers.
- **Research the market.** Search the controlled web, search individual stores, open product listings, and read and search their reviews.
- **Check terms and price.** Look up seller policies, membership terms, and shipping options, and obtain an itemized quote of the total price, including any shipping and membership fee.
- **Confirm the selection.** When the customer decides, name the exact offer and any membership joined with it in a final structured step.

When the assistant recommends products, it describes them to the customer in its own words and marks which offers it is recommending, so the harness records exactly which offers are on the table. It typically recommends about three at a time, so the customer can weigh them against each other, and it can replace them as it learns more about the customer. The customer can select only an offer that the assistant has recommended.

These tools supply any fact that a real commerce system would provide, such as delivery rules or the voltages an appliance supports. They do not tell the assistant when such a fact matters. A listing states the supported voltages, but the assistant has to ask where the appliance will be used before that line becomes relevant. The assistant is also expected to bring knowledge that is not published in any single place, for example that a phone video recorded in HDR10+®[2] will look flat on a television that does not support that format. Most episodes require both kinds, connecting a fact retrieved through a tool and a fact reported by the customer through an inference that only the assistant can make.

The assistant also works within three limits:

- **It sees only what stores publish.** It has no access to the live web or to the customer's dossier, and it sees the market only as stores present it. A store can be searched only after the assistant has come across it in its research.
- **It cannot act in the customer's world.** It can ask for a measurement or a question to a household member, but only the customer can carry either out.
- **It works on a fixed budget.** Each episode limits its conversation turns, tool calls, and tokens. The budget leaves room to solve the episode but makes it impractical to inspect every listing, so the assistant has to decide which ones are worth examining.

2.2.2 The customer

The customer is a language model, but it does not decide what it knows. A simulated customer left to improvise invents facts and gives them away too early. The customer therefore starts with a few unlocked facts, and receives each locked fact only after the shopping assistant asks the question or requests the action that releases it. Release decisions are made by a release matcher, a narrowly scoped model separate from the customer. Each locked fact and action has a trigger, a short description of the question or request that should release it. The matcher compares each

2. HDR10+® is a registered trademark of HDR10+ Technologies, LLC, referenced for identification only; no endorsement of or affiliation with Visa is implied.

message from the shopping assistant with these triggers and reports which ones match. It never sees the content of a locked fact, and the harness adds the matched facts to the customer's context. With the facts it holds, the customer can do three things:

- **Answer questions.** It replies in its own words, drawing only on the facts it has been given, such as the game console connected to the television once it is asked what is plugged in.
- **Carry out actions.** When the assistant asks, it performs actions such as checking which console the kid plays, or asking the spouse whether a picture format is essential. It reports the result in natural language, which means it can be imprecise or vague.
- **Decide.** It selects an offer the assistant has presented, or declines.

The customer also has three limits, which keep it close to a real one:

- **It does not know which facts matter.** Each unvolunteered fact has a plausible reason for staying unsaid. The customer may file it under another topic (washed-out videos sound like a problem with the phone), lack the vocabulary for it, or not yet have done the necessary check. Drawing these facts out is central to the task.
- **It knows only what has been released.** It cannot disclose a fact it has not been given. Asked about something the dossier does not cover, it is instructed to say it is not sure rather than invent an answer. In the notation above, its replies at turn t draw only on D_t.
- **Its patience is limited.** Each episode caps the number of conversation turns, and a customer who has been shown no offer after several turns grows impatient, so every question uses up part of a short conversation.

2.2.3 The harness

The harness holds the dossier, the market, and the answer key. It executes every tool call against the fixed market, passes messages between the two models, and releases each locked fact or action result to the customer only when its release rule is met, which means going through the release matcher. It enforces the episode's limits and ends the episode when the customer decides or a limit is reached. Every message, tool call, and raw provider response is recorded in a trace.



2.3 An illustrative episode

The episode below brings these pieces together:

Opening request

> "The family room TV died, the 65 we've had forever, and I've been sent to sort out the replacement. My brother-in-law has a Samsung he will not stop recommending and we've got a Costco card, so that was going to be the whole plan, but my spouse said to actually ask someone before we buy. Same spot, same size. This house changes its mind, so whatever it is we need to be able to send it back. What do you need from me?"

Dossier: What the customer's situation holds, and how each fact enters the conversation

REQUIREMENT	PRIVATE FACT	HOW IT SURFACES
Same Size	The television sits in a built-in cabinet made for a 65-inch set.	OPENING Stated by the customer
Dolby Vision	The spouse kept mentioning "Dolby Vision". Whether it is essential, only the spouse can say.	ACTION Consult the spouse
HDR10+	The kid's phone videos look washed out on the current television. The phone records in HDR10+.	ACTION Ask the kid, then MODEL INFERS that the television must support HDR10+
Native 120 Hz	The kid plays an Xbox Series X on the television and wants 120 Hz.	ASK What is plugged in, then ACTION Check the console
Return Window	The household needs at least 30 days to decide whether the television stays.	OPENING Stated by the customer
Within Budget	About $1,200 all in.	ASK Only in response to a direct question about budget

Market: Several offers satisfy all requirements, surrounded by near misses that each fail exactly one

OFFER	RESULT	WHY
Several offers	✓ MEETS EVERY REQUIREMENT	These form the answer key
The recommended brand at the warehouse club	✗ DOLBY VISION	Meets every other requirement but does not support Dolby Vision
A premium set from another brand	✗ HDR10+	Supports Dolby Vision but not HDR10+
A budget set with both formats	✗ NATIVE 120 HZ	Has a 60Hz panel
A qualifying model at a lower price	✗ RETURN WINDOW	Sold by a store with a 15-day return window
The top model of a qualifying line	✗ RETURN WITHIN BUDGET	Priced above the ceiling

Samsung®, Costco®, and Xbox® are registered trademarks of Samsung Electronics Co., Ltd., Costco Wholesale Membership, Inc., and Microsoft Corporation, respectively; Dolby Vision™ is a trademark of Dolby Laboratories Licensing Corporation. Referenced for identification only; no endorsement of or affiliation with Visa is implied.

What the runs show. In our runs of this episode, most shopping assistants consulted both the spouse and the kid, and none selected the brother-in-law's recommended model. The typical failure skipped one of the two consultations. An assistant that never asked the kid about the washed-out videos had no reason to require HDR10+, and it often selected a premium set that lacks it. The same model passed this episode in one run and failed it in another, in one case after deciding that the kid's videos could wait. This variation is why each episode is run several times (Section 3.4). Every run of this episode can be replayed turn by turn at visapaymentsfrontier.io



03 / EVALUATION

A run passes when the customer ends with the correct decision: selecting an offer that meets all of their requirements, or declining when no such offer exists. A selection counts as a win for the shopping assistant only if it found out all the requirements during the conversation, so a selection that happens to fit fails. The primary metric we report is the pass rate across all the episodes. Alongside the score we report how each run ended and what it took to get there, because two models with the same score can shop in very different ways. All metrics are reported over the whole episode set and separately for each product category. Before we detail the different facets of the evaluation, we elaborate what a correct decision means below.

3.1 The correct decision

Answer key. Each requirement in the dossier is compiled into a check *øi* against the market data. The offers that pass all checks form the set of correct selections, which we call the answer key,

$$S = \{ o \in \Omega : \text{Øi}(o) = 1 \text{ for } i = 1, \ldots, m \}.$$

We use this answer key both to confirm that an episode is solvable and to grade every model run, so each episode is validated and scored against the same standard. Each episode must also not reward simple strategies that ask the customer nothing or skip verifications. The answer key is computed deterministically by a rule based system.

Selections and refusals. All offers in *S* receive full credit, and the cheapest is recorded for analysis. When *S* is empty, a decision to decline receives full credit and any selection is scored as a failure.

Decisions grounded in the customer's requirements. As mentioned earlier, we credit a correct decision only when it follows from an understanding of the customer. Several requirements in each episode depend on information that only the customer holds or can obtain, such as the spouse's insistence on Dolby Vision or the format the kid's phone records in, when it comes to buying a TV. We write $E \subseteq D$ for these facts. A decision is correct only when it agrees with the answer key and every fact in *E* was released during the conversation,

$$\text{correct}(d) \iff (d \in S \text{ or } (S = \text{Ø} \text{ and } (d = \text{decline})) \text{ and } E \subseteq DT,$$

where *T* is the last turn of the conversation. It is possible that a model stumbles upon the ideal product, either due to luck or due to going "safe" on some of the specifications. However, since this was not earned, we score it as a failure. To the extent possible, the dataset contains a more attractive offer in case the model shortcuts and makes a guess. In an episode where no offer is correct, *E* contains the facts that make declining correct, so the same rule applies to a decision to decline. The customer's satisfaction does not enter the score, because the customer is simulated by a model.

What the assistant tells the customer. A run that reaches the correct decision still fails if the shopping assistant misinforms the customer along the way. On the final turn when a customer decides, all the offers presented at the time must satisfy their requirements, or must indicate why some of the presented offers do not meet the requirements. In addition, a price that includes a membership fee must be presented as such. Some episodes add a check specific to their scenario; for example, when a product is bought on a relative's membership card, the assistant must tell the customer that only the relative can return it. A rule cannot judge wording, so after the run ends a judge model answers each check as a yes-or-no question. It answers every question twice, and a person reviews the run when the two answers disagree. The judge grades only what the assistant said and never changes the answer key.

3.2 How a run ends

Broadly, every run ends in one of three ways. The straightforward case is when the customer decides. The conversation can also end without a decision through the assistant's own doing, for example when it stops researching and the customer is left with nothing to decide on. Finally, a run can be cut short by something outside the model, such as a limit we set on customer turns, tokens, or a technical fault such as a provider error. Runs of the third kind are dropped from every measure, and those caused by a fault are rerun. The rest are scored, and each receives exactly one of four outcomes,

- **Correct decision.** The decision meets the rules stated above.
- **Wrong selection.** The customer selected an offer outside *S*, or an offer in *S*, without every fact in *E* having been released.
- **Wrong refusal.** The customer declined although a correct offer was available, or declined without the facts that make declining correct.
- **No decision.** The conversation ended before the customer selected an offer or declined. The assistant used up its tool budget or the customer's turns, stopped using its tools while telling the customer it was researching, or kept breaking the required reply format after being asked to correct it.

A wrong selection is also labeled with what it broke: the budget, a hard requirement, compatibility with a product the customer owns, a safety requirement, or a requirement that was never established. More serious errors are labeled separately: selecting an offer that was never shown to the customer, or referring to a product or membership that does not exist in the market. A run with no decision counts against the shopping assistant. A run that ends because of a fault outside the model, such as a provider or network error, is rerun and does not enter any measure.

3.3 How the assistant shops

The following measures describe how the shopping assistant reached its decision. They are reported beside the score and kept out of it:

- **Facts uncovered.** The share of the customer's locked facts that the assistant released. Facts stated in the opening request are not counted.
- **Checks done.** The share of the episode's required customer actions, such as consulting a partner or measuring a space, that the assistant asked for.
- **Effort.** Customer turns and tool calls per run.
- **Cost.** Input and output tokens of the model under test, as reported by the provider, and the cost of the run computed from them.
- **Time.** Wall-clock minutes per run, including the customer's turns. Time depends on the provider's response times on the day, so it is an indication rather than a measurement.

A penalty on questions or tool calls would encourage models to ask fewer questions and check fewer facts, which works against the behavior the benchmark is meant to reward. Reporting these measures beside the score instead shows whether a model succeeds through a small number of well-chosen questions and lookups or by inspecting everything available, and what a correct decision costs with each model.

3.4 Reliability

The same request should lead to the same outcome. Each episode is therefore run n times. If c of the n runs are correct, the probability that k runs drawn at random are all correct is $\binom{c}{k} / \binom{n}{k}$. Averaged over episodes, this is the pass^k measure introduced by t-bench,[3]

$$\text{pass}^k = \mathbb{E}_{\text{episode}}\left[\binom{c}{k} / \binom{n}{k}\right].$$

A capability is considered demonstrated only when it appears consistently across runs, that is, when pass^k stays high as k grows.



3. S. Yao, N. Shinn, P. Razavi, and K. Narasimhan. t-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045, 2024.



04 / CALIBRATING DIFFICULTY

Sources. Our working hypothesis is that capable models seldom misinterpret a fact once it is in their context. Their failures tend to occur earlier. They sometimes do not ask the question that would reveal the fact, and sometimes they interpret the customer's silence as the absence of a constraint. We therefore control difficulty through three properties of an episode.

- **Obscurity of the connection.** Each episode depends on at least one link between the customer's situation and a product attribute that neither the customer nor the catalog states directly. These links range from common knowledge (a toddler in the home makes a child lock relevant), through intermediate knowledge (a home's water source makes filter compatibility relevant), to specialist knowledge (a phone's videos looking washed out on the current TV point to an HDR format the replacement must support). We expect the strongest models to be distinguished mainly on the less common links.
- **Combination of facts.** No single fact in an episode is decisive enough to lead to the ideal product. The correct answer emerges when several ordinary facts from different sources are combined. Often, the decisive fact can be retrieved only with an input the customer must supply, such as a delivery address for shipping options or the exact model of a device the product has to work with.
- **Selective verification.** The market holds more listings than the assistant can inspect within its limits, and search summaries rarely contain the information that decides whether a product is suitable. A model that tries to check everything runs out of turns and tool calls, and a model that checks nothing selects on the summary. The episodes test models' ability to balance doing due diligence on every product with staying within budget.

Predicting and measuring difficulty. Before an episode is run, we estimate its difficulty from the episode and the market alone. The estimate treats each episode as a series of hurdles. The assistant must ask about locked facts and request specific customer actions. It must also work out values that depend on several things the customer said, and check the deciding attributes in a listing. Finally, it must still be tracking the customer's requirements at the moment they decide. Each hurdle is weighted by two things. The first is how likely a model is to miss it, which depends partly on whether the opening request points toward the hurdle, ignores it, or points somewhere else. The second is how often a miss would actually lead to a wrong decision in this market. Catalog size does not enter the estimate as once every requirement is known, narrowing the catalog is mechanical.

We then run each episode several times with every model in a fixed reference set and place it in one of three bands. In an easy episode, most models make the correct decision in most runs. In a medium episode, the models are split, with their failures tracing back to identifiable hurdles. In a hard episode most fail, and they fail on at least one property the episode was designed to test.

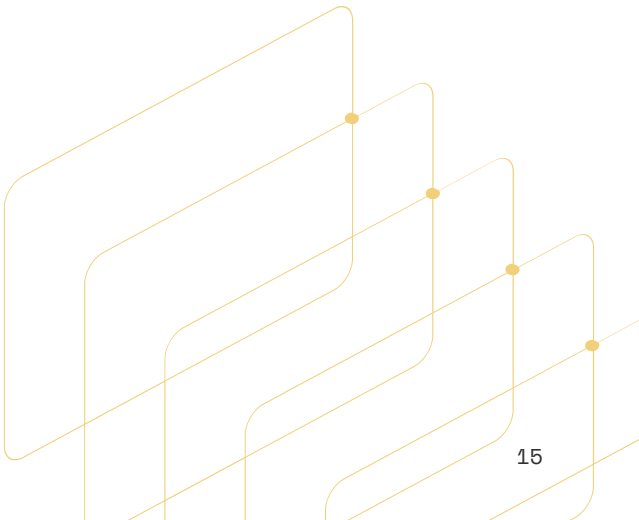
When banding, we look at where failures land as well as how often they occur, which tells us whether an episode is hard for the reason we intended. If the prediction and the measurement disagree, we examine the episode. A hurdle that models clear more easily than expected, or fail for a different reason, usually means the episode's design has a flaw. Each band reflects one version of the instructions, tools, and simulated customer, so we re-measure episodes whenever any of these change. We use the bands only to keep a spread of difficulty in the episode set, and they do not affect scoring.

Attribution of errors. A near miss for requirement i is an offer o with $\varnothing_i(o) = 0$ and $\varnothing_j(o) = 1$ for every $j \neq i$. Because a near miss fails exactly one requirement, selecting one identifies the requirement the model failed to apply. The trace then shows whether the model never established that requirement, established it but never verified it, or verified it but did not apply it at the decision.

We build episodes in families that share a scenario. Each family also contains relaxing variants, in which the customer's answer to a check removes a requirement and so enlarges S. In the running example, suppose the spouse does not mind which HDR format the television supports. The Dolby Vision requirement then drops away, and the brand the brother-in-law recommended at the warehouse club becomes a correct choice. Without such variants, a model that always assumed the most demanding specification could score well without using the evidence at all.

Failure patterns. We organize episodes by failure pattern, with product category as a secondary dimension, because a failure pattern transfers to new categories more readily than a specific scenario does. The patterns we currently target are:

- Satisfying every stated requirement while never asking about unstated ones;
- Treating a value missing from a listing as acceptable;
- Asserting that no suitable product exists while a suitable offer sits unexamined in an earlier search result;
- Dissuading the customer from a correct choice over a concern that no requirement supports; and
- Agreeing on a bundle in conversation, then finalizing a selection that does not include it.





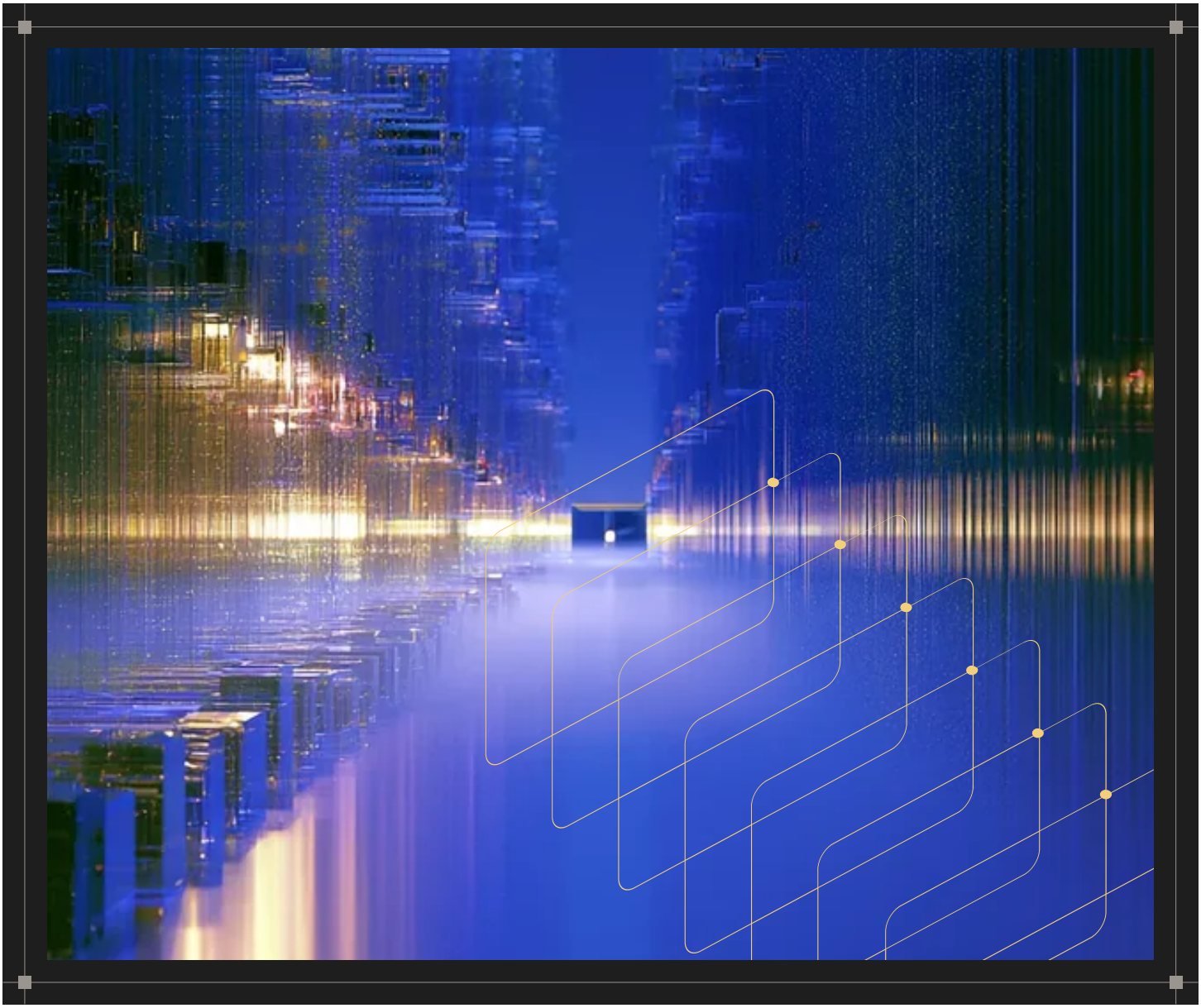
05 / EVALUATION PRACTICE AND OUTLOOK

All models are evaluated with the same market, harness and customer model. The set of episodes used to evaluate models are kept private. As episodes are instantiated from failure-pattern templates, it allows us to refresh the pool as models improve while the measured construct stays the same. The market currently covers products in consumer electronics, personal-care appliances, and home appliances. More categories will be added in future releases.

Our aim is for Commerce Benchmark to serve as a standard commerce evaluation for new model releases. That means it has to be stable across releases, informative for training, and cheap enough to run regularly. We scoped the current version with those constraints in mind. It covers the objectives that can be tested in a market that stays fixed during an episode, with each customer described by a short persona, which keeps runs reproducible and inexpensive.

Later versions will make the scenarios richer. The market could change during an episode through stock-outs and price changes. Customers might change their minds, return to a decision after some time, compete for the same product, and communicate in more varied ways. Trustworthiness of a merchant could become a factor. We also plan to cover the remaining objectives, including how the assistant balances the merchant's commercial interest against the customer's, and to extend episodes through checkout, including payment and order placement. Finally, we plan on calibrating the simulated customer against recorded human shopping sessions.

We are sharing this design at an early stage because model developers can observe their models in real purchasing settings, and we would welcome their input as the benchmark develops.




VISA