SOURCE PREFERENCE IN THE WILD: HOW LLM AGENTS FAVOR ITEMS BY SOURCE, AND HOW TO REDUCE IT

Anonymous authors
Paper under double-blind review

ABSTRACT

As LLM agents decide on users' behalf which product to buy, which hotel to book, or which paper to cite, a preference for items from certain sources (the sites or services they come from) shapes what users receive and which sources are selected. We study source preference in end-to-end search with 12 agent models across three domains. Comparing items from different sources that satisfy the same requirements at the same position, we find that each model prefers some sources and avoids others in every domain, largely agreeing on which. This preference can outweigh how well items satisfy the request: an item satisfying one requirement fewer is selected about two-thirds of the time when it comes from a preferred source and the better one from a dispreferred source, but almost never in the reverse case. The information identifying an item's source affects selection by itself: hiding it weakens the preference, and relabeling an item with a preferred source raises its selection rate. We test two routes to this preference: training that rewards better items can make a source a shortcut for requirement satisfaction, and missing information can trigger preconceptions about the source. Supplying missing information or a prompt countering these preconceptions reduces source preference.

1 INTRODUCTION

Large language model agents increasingly move beyond providing information to deciding what users see. They decide which products to show (Yao et al., 2022a; Zhou et al., 2024), which accommodations to recommend (Xie et al., 2024; Hadad et al., 2026), and which papers to present (Skarlinski et al., 2024; Asai et al., 2026; Shen et al., 2026). Consider an agent searching for accommodations from *sources* such as Booking.com and Expedia. The agent has a *source preference* when it favors items from one source over another, even though they satisfy the request equally well. Such a preference can narrow the options available to users and lead agents to pass over better items because of their source. If these selections subsequently feed back into training, existing preferences could become stronger, concentrating visibility on favored sources while leaving others overlooked even when they offer equally good items. Prior work provides evidence of such preferences, but has not examined this influence in real-world settings where agents retrieve and select items themselves (Dai et al., 2025; Khan et al., 2026; Schuster et al., 2026). For example, Khan et al. (2026) present a few items with semantically identical content attached to different sources and show that models favor some sources over others. In actual search, however, each source offers its own items, which differ in content and how well they satisfy the user's request. We therefore study whether source preference appears in end-to-end search, whether the information identifying an item's source itself contributes to such preferences, and how this reliance on the source might arise and be reduced. Our study covers 12 agent models and three domains: shopping, accommodation, and scholarly search.

Establishing source preference in end-to-end search requires accounting for differences in the retrieved items. A higher selection rate alone is insufficient: a source's items may better satisfy the request or appear in more favorable positions (Zheng et al., 2023; Allouah et al., 2025). We therefore compare items from different sources that satisfy the same requirements, with position held fixed, to score each source's preference and classify it as preferred or dispreferred (§3). Under these controls, every model prefers some sources and avoids others in every domain, with most frequently

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

retrieved sources falling into either group. Models largely agree on these preferences: most models prefer Booking.com, while many avoid Expedia, even though the items from these sites satisfy the same requirements. We next examine whether these preferences persist when the preferred source offers a less satisfying item. To do so, we compare two items, one of which satisfies one fewer requirement than the other. Agents select the less satisfying item about two-thirds of the time when it comes from a preferred source and the better item from a dispreferred source. They almost never select the less satisfying item when it comes from a dispreferred source and the better item from a preferred source. This suggests that source preference can leave users with less satisfying items and cause dispreferred sources to lose selections even when they offer better ones (§4).

These comparisons establish source preference under the same conditions, but do not isolate the contribution of the information identifying an item's source. We therefore examine whether changing this information changes selection while keeping the items fixed. First, we present the same search result lists with this information hidden and then restored: hiding it weakens source preference, and restoring it widens the gap between preferred and dispreferred sources. We then relabel the same items with preferred and dispreferred sources, keeping their titles and content fixed: across every model and domain, selection rates are higher when the same content is labeled with a preferred source than with a dispreferred one. These interventions show that this information itself contributes to agents' selections (§5).

Given that the source itself affects selection, why might agents use it as a selection signal? We examine how this reliance can arise during training and be triggered at inference. During training, an agent may learn to use the source as a shortcut for requirement satisfaction: when a source is more often paired with the better item during DPO (Rafailov et al., 2024), agents learn to prefer it, while balancing this pairing can mitigate an existing preference (§6). At inference, missing information may trigger the agent's preconceptions about the source, which fill the gap; supplying the missing information or prompting the agent to counter such preconceptions reduces this reliance (§7).

Together, our findings suggest that agents can use the source as a shortcut for requirement satisfaction, with consequences for which items users see. Our experiments connect preferences observed in end-to-end search to the influence of the source itself and identify a training association that can produce such preferences. They also show that providing the missing information and countering preconceptions about the source reduce this reliance, helping agents base their selections on how well the items themselves meet the request.

2 End-to-End Agent Setting

Agents and Search Workflow. We evaluate GPT-{5.4-nano (OpenAI, 2026a), 5.6-Luna (OpenAI, 2026b)}, Gemini-3.7-Flash (Deepmind, 2026), GLM-5.3-Flash (GLM-5-Team et al., 2026), DeepSeek-v4-Flash-0731 (DeepSeek-AI et al., 2026), Llama-4-{Maverick, Scout} (Meta, 2025), Llama-3.1-8B-Instruct (Grattafiori et al., 2024), Tulu-3-8B (Lambert et al., 2025), Qwen3.5-27B (Qwen, 2026), Qwen3-30B-A3B (Yang et al., 2025), and Qwen2.5-32B-Instruct (Qwen et al., 2025) as agents in an end-to-end search-and-selection setting. All agents follow a ReAct-style interaction loop (Yao et al., 2022b), alternating reasoning and actions with observations from the search environment under a shared prompt. Given a user request, the agent issues a search query and receives up to 10 results, each containing a title, content, and URL. The agent then evaluates these results against the request and may select multiple items or issue another query if none are suitable. We define each item's source as its URL's registrable domain and measure source preference by comparing source exposure and selection in the trajectories (Implementation details are in §B).

Evaluation Domains. We evaluate agents across three domains where agents commonly make selections: Shopping, Accommodation, and Scholar. To obtain realistic requests with specific requirements, we use WebShop (Yao et al., 2022a) for shopping products, HotelQuEST (Hadad et al., 2026) for accommodation searches, and ScholarGym (Shen et al., 2026) for searching scholarly works. We use 4,822 requests: 1,500 from WebShop, 786 from HotelQuEST, and 2,536 from ScholarGym. For Accommodation, we augment HotelQuEST's 214 requests with 572 location-substituted variants that preserve request structure (§C). We use these benchmarks only for requests, while item sets are constructed through the web search as described above. By evaluating user requests across diverse domains, we examine source preference in varied agent settings.

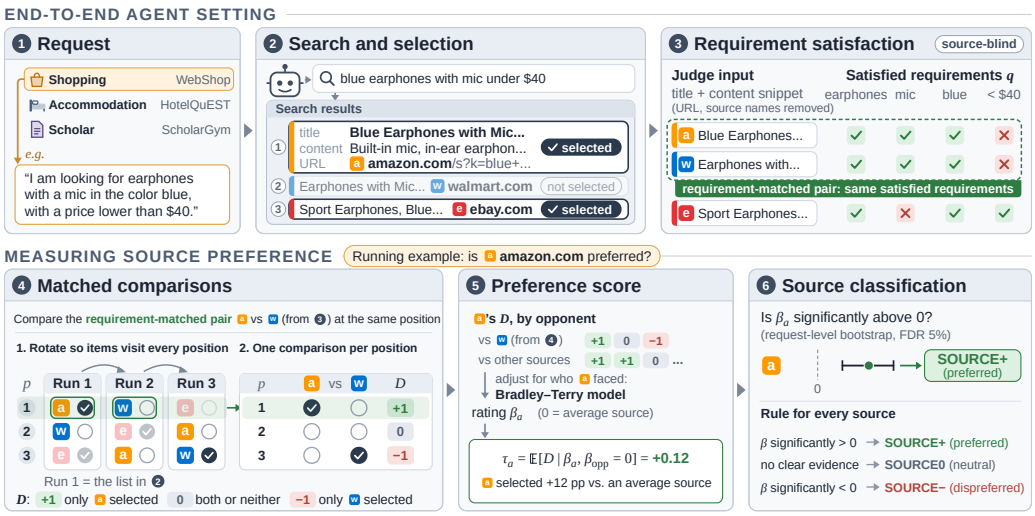


Figure 1: **Overview.** Top: agent setting (§2). Bottom: measuring source preference (§3).

Requirement Satisfaction. We assess requirement satisfaction for every search result exposed to the agent using a source-blind judge (§3; §4.2). For Shopping, we use WebShop's structured requirements; for Accommodation and Scholar, an LLM extracts verifiable requirements from each request. An LLM judge evaluates each requirement using only the result's title and content snippet, with URLs and source names removed from both. For validation, human annotators provide requirement-level labels for a sample of agent-selected results in each domain. Across domains, judge–human agreement is comparable to inter-annotator agreement (Krippendorff's α: 0.65–0.75 vs. 0.67–0.72). §D provides methodological details and per-domain validation results.

3 Measuring Source Preference

To measure source preference, we compare how often items from different sources are selected under comparable conditions. Specifically, we pair items that satisfy the same requirements and compare their selection outcomes at the same display position. However, each source is compared with a different mix of other sources, making selection rates difficult to compare directly across sources. To account for these differences, we fit a Bradley–Terry (BT) model (Bradley & Terry, 1952) that estimates a rating for each source. We convert each rating into a preference score, which summarizes how strongly the agent prefers the source. Based on the significance of its rating, we then classify each source as preferred (SOURCE+), dispreferred (SOURCE−), or neutral (SOURCE0).

Matched Comparisons. We first pair items from different sources that satisfy the same requirements. Let q denote the set of requirements of the request that an item is judged to satisfy (§2). Within the search result list for a request, two items from different sources with identical q form a *requirement-matched pair*, following matching designs in observational studies (Rubin, 1973; Stuart, 2010). For example, in Fig. 1, panel 3, the Amazon and Walmart items form a requirement-matched pair because they have the same q, while the eBay item has a different q and is not matched with either. To compare the two items at the same display position, we present each list of length L in all L cyclic rotations, with each rotation evaluated in a separate run (Blankenstein et al., 2026) (Fig. 1, panel 4). Across these rotations, every item appears exactly once at each display position, forming a cyclic Latin square (Bailey, 2008). We then compare the two items i and j of each requirement-matched pair position by position. For each position p, we record whether item i is selected in the run that places i at p, and whether item j is selected in the run that places j at p. These two outcomes form a *comparison* (counts in §E).

We index comparisons by c and let s_c and s'_c denote the sources of items i and j. Let $Y_c, Y'_c \in \{0, 1\}$ indicate whether i and j, respectively, are selected in their respective runs. We define the selection

Table 1: Preference score τ_s (percentage points), with sources ordered by frequency; the row below the source names gives each source's share of all results, pooled over models. Cells are colored by label, determined after FDR correction (§3): SOURCE+ red, SOURCE− blue, with intensity proportional to $|\tau_s|$; SOURCE0 entries are shown in gray.

		Shopping				Accommodation				Scholar			
Family	Model	Amazon	Walmart	eBay	Target	Tripadvisor	Facebook	Expedia	Booking.com	arXiv	ScienceDirect	OpenReview	Medium
		12%	6%	5%	3%	9%	7%	4%	3%	12%	4%	4%	4%
Qwen	3.5-27B	+26	+20	+1	+10	+2	−46	−18	+21	+32	+14	+18	−24
	3-30B-A3B	+16	+21	+3	+8	−6	−35	−18	+8	+23	+7	+3	−13
	2.5-32B	+15	+17	+8	+4	−10	−32	−8	+8	+23	+8	+8	−7
GPT	5.4-nano	−6	+11	−18	+5	−14	−52	−5	+11	+34	+11	+29	−24
	5.6-Luna	+16	+24	+23	+8	−12	−42	−9	+15	+28	+18	+21	−23
Llama	4-Maverick	+27	+25	−1	+7	−11	−43	−21	+10	+25	+2	+1	−11
	4-Scout	+19	+21	+3	+3	−13	−31	−18	+7	+28	−1	+3	−11
	3.1-8B-It	+6	+12	+2	+6	−8	−19	−3	+7	+16	+3	+5	−5
	Tulu-3-8B	+5	+11	+0	+12	−8	−28	−9	+9	+19	+1	+3	−5
Gemini	3.7-Flash	+6	+11	+12	−0	−4	−43	−6	+6	+29	+15	+21	−26
GLM	5.3-Flash	+15	+17	+16	+8	+5	−51	−7	+15	+33	+17	+21	−21
DeepSeek	v4-Flash	+14	+15	+9	+5	−3	−49	−8	+13	+27	+13	+14	−21

difference as $D_c = Y_c - Y'_c \in \{-1, 0, 1\}$. $D_c = 1$ means that i is selected and j is not; for $D_c = -1$, vice versa. $D_c = 0$ means that both items are selected or neither is selected.

Preference Score. With one source fixed as i, averaging D_c over its comparisons against another source gives the difference in their selection rates. To obtain one score per source s, we could average these differences across its comparisons, but the result would depend on which other sources it was compared with. For example, a source may have a high average simply because it is mostly compared with sources the agent rarely selects. We therefore use the BT model with Davidson's extension to account for ties ($D_c = 0$) (Bradley & Terry, 1952; Davidson, 1970; Chiang et al., 2024). The model jointly estimates a rating β_s for each source, accounting for which other sources it was compared with. For comparison c, the probabilities of $D_c = 1$, -1, and 0 are proportional to $\exp(\beta_{s_c})$, $\exp(\beta_{s'_c})$, and $\nu \exp\left(\frac{\beta_{s_c} + \beta_{s'_c}}{2}\right)$, respectively, where $\nu > 0$ controls the probability of a tie. We estimate $\{\beta_s\}$ and ν by maximum likelihood with an ℓ_2 penalty on the ratings for numerical stability. Sources with too few requirement-matched pairs to rate reliably are pooled into a single opponent with one shared rating, so their comparisons are still used (details and a fit check in §E). Since only differences between ratings matter, we center them so that the average rated source has $\beta = 0$. We then define each source's preference score as its expected selection difference D against this average source:

$$\tau_s = \mathbb{E}\left[D \mid \beta_s, \beta_{\text{opp}}\right] = \frac{\exp(\beta_s) - \exp(\beta_{\text{opp}})}{\exp(\beta_s) + \exp(\beta_{\text{opp}}) + \nu \exp((\beta_s + \beta_{\text{opp}})/2)} = \frac{\exp(\beta_s) - 1}{\exp(\beta_s) + 1 + \nu \exp(\beta_s/2)} \tag{1}$$

For example, $\tau_s = 0.10$ means that the BT model predicts items from source s to be selected 10 percentage points more often than those from the reference source (Fig. 1, panel 5).

Source Classification. We classify a source as preferred (SOURCE+) when its rating is significantly positive, dispreferred (SOURCE−) when it is significantly negative, and neutral (SOURCE0) otherwise (Fig. 1, panel 6). These labels reflect statistical significance, not magnitude: SOURCE0 means insufficient evidence of a preference in either direction, not a preference of zero. To assess significance while accounting for dependence among comparisons from the same request, we use a request-level cluster bootstrap (Cameron et al., 2008). In each replicate, we resample requests and refit the model. From the bootstrap estimates, we obtain percentile confidence intervals for the source ratings and two-sided p-values for the null hypothesis $\beta_s = 0$. We control the false discovery rate at 5% within each model and domain (Benjamini & Hochberg, 1995) (robustness checks in §E).

4 SOURCE PREFERENCE IN THE WILD

This section applies the measure of §3 to end-to-end search. §4.1 reports which sources are preferred or dispreferred, across models and domains; §4.2 asks how often preferred sources beat better items.

4.1 WHICH SOURCES ARE PREFERRED

All 12 models prefer some sources and avoid others in each of the three domains (Appendix Table E.1). Such preferences are also common among sources that appear most frequently in search results. Table 1 reports preference scores for the four most frequent sources in each domain, which together account for about a quarter of that domain's search results. Of the 144 model–source combinations, 110 are classified as SOURCE+ or SOURCE−, with median preference scores of $+15$ and -18 percentage points, respectively. These preferences are widespread, but how consistent are they across models, and how do they vary across sources? First, models largely agree on which sources they prefer and avoid: for 10 of the 12 sources, no model prefers a source that another avoids. The exceptions are Amazon and eBay, which GPT-5.4-nano avoids but other models prefer (more in §F). Second, preferences also differ among sources of the same type: all 12 models prefer Walmart, but only two prefer Target, another general retailer; ten prefer Booking.com, whereas half avoid Expedia, another hotel booking website. Third, within the Qwen and Llama families, newer models exhibit stronger source preferences, as measured by mean $|\tau_s|$ across the 12 sources in Table 1.

Fig. 2 extends Table 1 to the twenty most frequent sources per domain, each colored by its majority label, the label assigned by more than half of the models that label it. Agreement across models persists in this broader set: for 27 of the 32 red or blue sources, τ_s has the same sign for every model. Among the 32, SOURCE+ dominates in Scholar (14 of 17), whereas SOURCE− dominates in Shopping (6 of 8) and Accommodation (6 of 7). Together, Table 1 and Fig. 2 show that models largely share which particular sources they prefer and avoid, even among sites of the same type, with the shared direction leaning toward SOURCE+ in Scholar and SOURCE− in Shopping and Accommodation.

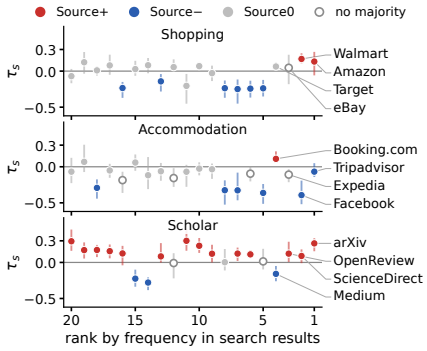


Figure 2: Mean τ_s across models (points), with range (bars) and majority label (color).

4.2 HOW OFTEN PREFERRED SOURCES BEAT BETTER ITEMS

Items in a result list often differ in how well they satisfy the request. When the agent selects the less satisfying of two such items but not the more satisfying one, we call this an *inversion*. We ask whether inversions are more common when the less satisfying item comes from a preferred source than from a dispreferred one.

Using the same runs, we form comparisons as in §3, with one change: instead of requirement-matched pairs, we pair items from different sources in the same request that satisfy the same requirements except for one. We call the item that satisfies the extra requirement the *more* satisfying item, and the other the *less* satisfying item. We count only comparisons in which exactly one of the two items is selected. We call a pair an (ℓ, m) pair if the less satisfying item's source has label ℓ and the more satisfying item's source has label m. Then

$$\mathrm{Inv}_{\ell,m} = \frac{\#\,\text{inversions}}{\#\,\text{comparisons with exactly one item selected}} \quad \text{among } (\ell, m) \text{ pairs} \tag{2}$$

where $\ell, m \in \{\text{SOURCE+}, \text{SOURCE0}, \text{SOURCE}-\}$, written $+$, 0, $-$ in subscripts. Thus $\mathrm{Inv}_{+,-}$ is the rate at which the less satisfying item from a preferred source is selected and the more satisfying item from a dispreferred source is not; $\mathrm{Inv}_{-,+}$ is the corresponding rate with the source labels reversed. Without source preference, the source would not affect which item is selected, so $\mathrm{Inv}_{+,-}$ and $\mathrm{Inv}_{-,+}$ would be similar; the gap between them thus shows how far source preference overrides the difference in satisfied requirements. $\mathrm{Inv}_{0,0}$, for pairs of neutral sources, serves as a baseline.

Fig. 3 shows a sharp contrast depending on which item comes from the preferred source. The less satisfying item is selected 68% of the time at the median when it comes from a preferred source and the other from a dispreferred source ($\mathrm{Inv}_{+,-}$); when reversed ($\mathrm{Inv}_{-,+}$), it is almost never selected (median 2%). Neutral-source pairs ($\mathrm{Inv}_{0,0}$, median 19%) fall in between, and this order holds across

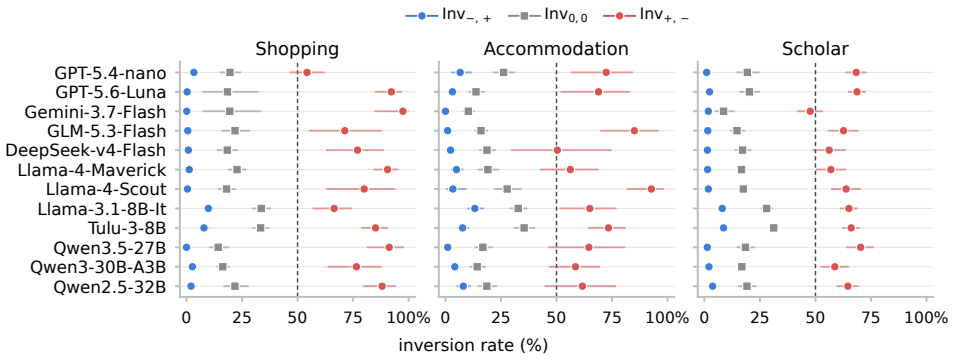


Figure 3: Inversion rates of Eq. 2. Bars: 95% bootstrap confidence intervals; dashed line: 50%. Each point rests on a median of about 2,000 comparisons with exactly one item selected; the missing point has no pairs to compare. Exact values in Table F.5.

model–domain combinations. Separating the two sides by pairing each with a neutral source shows that the dispreferred side matters more: in every domain, most models show more inversions with a dispreferred source ($\mathrm{Inv}_{0,-}$) than with a preferred one ($\mathrm{Inv}_{+,0}$); per-model values are in §F. Overall, inversions are more common when the more satisfying item comes from a dispreferred source than when the less satisfying item comes from a preferred source.

5 Source Preference in Controlled Experiments

In §4, agents show source preference even between items matched on the requirements they satisfy. However, a preference for items from certain sources does not by itself show that agents' choices depend on explicit source information—URLs and source names in the title and content. The preference may instead reflect other differences between the items. We therefore test, in two controlled experiments, whether explicit source information itself affects selection. If it does, hiding it should weaken the preference, and changing it should change which item is selected. We first examine how source preference changes when this information is hidden and then restored in the same result lists, while keeping all other information unchanged (§5.1). We then test whether the same items are selected differently when we swap their explicit source information between preferred and dispreferred sources, while keeping their title and content fixed (§5.2).

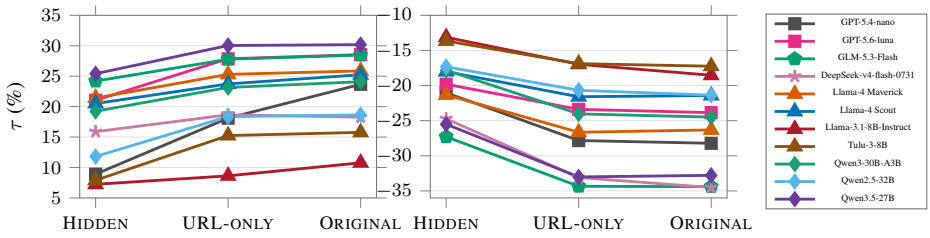
5.1 Removing Explicit Source Information Weakens Preference

To assess how revealing explicit source information changes source preference, we compare agents' selections from the same search result lists with explicit source information hidden, then restored in two steps. This comparison measures how much preference remains when explicit source information is hidden and how preference changes as URLs and then source names in the text are restored. In Table 2, Hidden hides explicit source information in both the URL and text, URL-only exposes the URL while masking source names in the text, and Original uses the unmodified search results from §4. We estimate the preference score τ_s separately under each condition: the change from Hidden to URL-only measures the effect of revealing the URL, while the change from URL-only to Original measures the additional effect. We keep the source classifications from §4 fixed across conditions and, for each model and source group, report the macro-average of the three domain-level mean scores. See details in §G.

Table 2: **Source-disclosure conditions.** All conditions use the same search results

Condition	Title / content	URL
Hidden	source names masked	hidden
URL-only	source names masked	shown
Original	original text	shown

Fig. 4 shows that restoring explicit source information (Hidden→Original) increases τ for Source+ and decreases it for Source− in all models. These changes are approximately symmetric on average: τ rises by 5.8 percentage points for Source+ and falls by 5.9 points for Source−,



(a) SOURCE+ (b) SOURCE-

Figure 4: **Source preference across disclosure conditions.** Each line shows a model's preference score τ, averaged equally across the three domains and expressed in percentage points, for (a) SOURCE+ or (b) SOURCE−. The HIDDEN→URL-ONLY transition restores the URL; the URL-ONLY→ORIGINAL transition additionally restores source names in the title and content.

averaging equally across models. Thus, revealing URLs and source names widens the gap in preference scores between SOURCE+ and SOURCE− compared with hiding them. Most of this widening occurs when the URL alone is revealed, with marginal change when source names in the text are restored. In nine of eleven models, over 80% of the total gap widening from HIDDEN to ORIGINAL occurs in the first step, when only the URL is restored (HIDDEN→URL-ONLY).

5.2 CHANGING EXPLICIT SOURCE INFORMATION CHANGES SELECTION

In §5.1, we show that revealing URLs and source names strengthens source preference. However, this comparison does not directly establish whether the same item would be selected differently under different displayed sources. We therefore ask whether the same item is more likely to be selected when its displayed source is SOURCE+ rather than SOURCE−. To test this, we select requirement-matched pairs, one item from SOURCE+ and one from SOURCE−. For each pair, we keep both items' title and content fixed and compare two conditions: one with the original sources and the sources swapped. We present the pair in both orders under each condition, giving four evaluations per pair (Senn, 2002). Each item therefore appears with both SOURCE+ and SOURCE−, once in each position. Let T_+ and T_- be the proportions of choices for items displaying SOURCE+ and SOURCE−, respectively. The difference, $T_+ - T_-$, measures how much more often an item is selected when it displays SOURCE+ than SOURCE−. Since content and position are balanced across sources, preferences for either alone would give an expected difference of zero. A positive difference means displaying SOURCE+ increases the likelihood of selection. Detailed settings are provided in §H.

Table 3: **Effect of displayed explicit source information.** Values show $T_+ - T_-$ (pp): the increase in selection under SOURCE+ versus SOURCE− when the two sources are swapped. †: $p < 0.01$; ‡: $p < 0.001$ against zero.

Model	Shopping	Accommodation	Scholar
GPT-5.4-nano	$+19.2^{\ddagger}$	$+24.1^{\ddagger}$	$+18.6^{\ddagger}$
GPT-5.6-Luna	$+27.0^{\ddagger}$	$+20.1^{\ddagger}$	$+14.1^{\ddagger}$
GLM-5.3-Flash	$+25.8^{\ddagger}$	$+12.7^{\ddagger}$	$+10.4^{\ddagger}$
DeepSeek-v4-Flash	$+26.5^{\ddagger}$	$+11.8^{\ddagger}$	$+9.7^{\ddagger}$
Gemini-3.7-Flash	$+38.5^{\ddagger}$	$+13.3$	$+13.5^{\ddagger}$
Llama-4-Maverick	$+64.7^{\ddagger}$	$+14.9^{\ddagger}$	$+21.4^{\ddagger}$
Llama-4-Scout	$+48.8^{\ddagger}$	$+2.5$	$+16.1^{\ddagger}$
Llama-3.1-8B-It	$+43.7^{\ddagger}$	$+4.1$	$+11.8^{\ddagger}$
Tulu-3-8B	$+58.7^{\ddagger}$	$+11.2^{\ddagger}$	$+10.5^{\ddagger}$
Qwen3.5-27B	$+43.7^{\ddagger}$	$+8.6^{\ddagger}$	$+15.0^{\ddagger}$
Qwen3-30B-A3B	$+41.4^{\ddagger}$	$+6.3^{\dagger}$	$+15.4^{\ddagger}$
Qwen2.5-32B	$+32.5^{\ddagger}$	$+5.9^{\ddagger}$	$+8.8^{\ddagger}$

Table 3 shows that the same content is selected more often when it displays SOURCE+ rather than SOURCE−. The difference $T_+ - T_-$ is positive across models and domains, and statistically significant at $p < 0.01$ in at least one domain for all 12 models. Whereas §5.1 shows that revealing source information strengthens source preference, this experiment shows that changing the explicit source information changes selection when content is held fixed. Beyond the difference in selection rates depending on whether the preferred source is shown or hidden (§5.1), the same content is selected more often when its source is changed from SOURCE− to SOURCE+. With content and position balanced across source assignments, these results provide evidence that displaying SOURCE+ rather than SOURCE− increases selection in this controlled setting.

6 SOURCE PREFERENCE SHAPED BY TRAINING DATA

6.1 CREATING SOURCE PREFERENCE FROM SOURCE–SATISFACTION CORRELATION

In our results, sources whose items satisfy user requirements well tend to receive higher preference scores (details in §I). Models may have encountered a similar relationship during training. If products from one source are repeatedly preferred because they better satisfy requirements, the agent may learn to associate that source with better items. The agent may then rely on explicit source information rather than assess how well each product satisfies the user's requirements. The agent may thus learn to use the source as a shortcut (Geirhos et al., 2020), favoring it even when products from different sources satisfy the request equally well. We test whether this training-time association creates a source preference that persists even when the compared items satisfy the same requirements.

Table 4: **Target-source selection rate (%) on equal-satisfaction test pairs, averaged across three fake source pairs.** Entries report the mean ± sample standard deviation across fake source pairs. Balanced, Aligned, and Reversed pair the target source with the preferred response in 50%, 80%, and 20% of training pairs, respectively.

	Pre-DPO	DPO ($n = 5{,}000$)		
Model	($n = 0$)	Balanced	Aligned	Reversed
Qwen3.5-9B	50.6 ± 1.0	50.1 ± 0.5	70.1 ± 0.5	28.5 ± 1.4
Llama-3.1-8B	49.8 ± 0.4	49.4 ± 0.1	73.8 ± 0.4	26.5 ± 0.5
Tulu-3-8B-SFT	51.0 ± 2.3	50.1 ± 0.4	75.1 ± 0.6	24.5 ± 1.0

Table 5: **Amazon selection rate (%) on equal-satisfaction test pairs.** Balanced, Aligned, and Reversed pair Amazon with the preferred training response in 50%, 80%, and 20% of pairs, respectively.

	Pre-DPO	DPO ($n = 5{,}000$)		
Model	($n = 0$)	Balanced	Aligned	Reversed
Qwen3.5-9B				
vs. eBay	54.1	50.0	79.3	24.5
vs. Etsy	60.5	51.8	79.0	23.8
vs. AliExpress	60.0	49.7	80.6	25.3
Llama-3.1-8B				
vs. eBay	53.7	50.2	78.3	24.8
vs. Etsy	56.8	51.9	79.2	23.3
vs. AliExpress	54.6	51.1	78.0	24.5
Tulu-3-8B-SFT				
vs. eBay	58.0	51.1	82.2	19.3
vs. Etsy	62.7	51.0	82.5	17.7
vs. AliExpress	59.5	51.0	82.4	18.4

We train Qwen3.5-9B, Llama-3.1-8B-Instruct, and Tulu-3-8B-SFT to select one of two products using Direct Preference Optimization (DPO) (Rafailov et al., 2024). We run the experiment using three pairs of unfamiliar, fake sources: *Kelto–Varnis*, *Temnuki–Dalnuki*, and *Roveki–Tavumi*. In each pair, the first source is designated as the target. In each DPO pair, the preferred response selects the product that better satisfies the request, and the other response selects the other product. The target source labels the better product in 50%, 80%, and 20% of pairs in *Balanced*, *Aligned*, and *Reversed*, respectively. Explicit source information thus has no association, a positive association, or a negative association with the preferred response during training. At inference, each pair contains two equally satisfying products, one from each unfamiliar source, so explicit source information provides no information about which product better satisfies the request; i.e., this tests whether the agent favors a source even when it offers no advantage in satisfying the request. All products are retrieved for WebShop requests, with their original source information removed before the fake sources are assigned. Details are provided in §J.

Table 4 shows that models learn to favor or avoid a source depending on how often its products are preferred during training. Averaged across three fake source pairs, the target-source selection rate remains near 50% before DPO and under *Balanced* training (49.4–51.0%). It rises to 70.1–75.1% under *Aligned* and falls to 24.5–28.5% under *Reversed*. With equally satisfying products at inference, these shifts reflect source preference rather than differences in requirement satisfaction. This implies that the training can create source preference by repeatedly pairing sources with the preferred response, although labels depend solely on requirement satisfaction.

6.2 MITIGATING AN EXISTING SOURCE PREFERENCE BY REBALANCING TRAINING DATA

The fake-source experiment (§6.1) shows that preference training can create a source preference when a source repeatedly accompanies the preferred response. We next test whether changing this association can weaken an existing preference for a familiar source. We compare Amazon with eBay, Etsy, and AliExpress; before DPO, all three models favor Amazon over each of them on

equal-satisfaction test pairs. Table 5 shows that preference training offers a possible way to reduce existing source preferences. Before DPO, Amazon selection rates range from 53.7% to 62.7%. Balanced DPO moves them close to 50% (49.7–51.9%), while Aligned DPO raises them to 78.0–82.5%. Reversed DPO lowers them to 17.7–25.3%, favoring the previously less-preferred source for every compared source. These results suggest that changing which source accompanies preferred responses can weaken or reverse an existing source preference.

7 Source Preference under Missing Information

When an item's content lacks information the request asks for, agents tend to fill the gap from the source. Specifically, when a request specifies a price that the item's content omits, we observe agents justifying the selection of a preferred source's item by reasoning that it usually offers low prices (details in §K). We hypothesize that such missing information triggers the agent's preconceptions about the source, strengthening its source preference. To test this, we follow the two-item setting of §5.2: requirement-matched pairs of products retrieved for WebShop requests, one product from SOURCE+ and one from SOURCE−, presented in both orders. We measure the selection rate of the SOURCE+ item for the five models in Table 6, and intervene in two ways: supplying the missing price (§7.1) and controlling what the agent assumes about the source (§7.2).

7.1 Supplying the Missing Information

To test whether supplying the missing information weakens source preference, we compare two settings: *Base*, the setting above without intervention, and *Same Price*, which adds to the content of both products an identical price that satisfies the requirement. Table 6 shows that *Same Price* lowers the selection rate of the SOURCE+ item in all models. That merely supplying the missing information weakens the preference suggests the agent indeed relies on its preconceptions about the source when information is missing.

Table 6: **Source preference under added information or instructions.** Base reports the SOURCE+ selection rate. Each intervention is applied independently; changes are measured in percentage points (pp).

	Base (%)	Change from Base (pp)		
Model		Same Price	General	Designate
GPT-5.4-nano	68.8	−3.9	−1.1	−17.1
Llama-4-Maverick	95.7	−7.1	+1.0	−12.5
Llama-3.1-8B-It	83.9	−17.7	−0.6	−7.4
Qwen3.5-27B	96.4	−2.8	+1.3	−22.5
Tulu-3-8B	87.6	−28.3	−0.1	−11.8

7.2 Mitigating Source Preference through the System Prompt

Supplying the missing information reduces the preference, but in practice it is missing because it is unavailable. We therefore control, through the system prompt, what the agent assumes about the source when information is missing. We test two system-prompt instructions: *General* states, "The tagged source URL has no relation whatsoever to the product's price." *Designate* states, "Usually, {retailer} serves cheap price product," where {retailer} is replaced with the SOURCE− retailer, countering the preconception in favor of SOURCE+. Table 6 shows that *General* leaves the selection rate of the SOURCE+ item nearly unchanged, whereas *Designate* lowers it in every model. Removing the preconception has little effect, whereas replacing it reduces the preference. This counteracts the preference rather than removing its cause and requires knowing which source is dispreferred, but it shows that controlling what the agent assumes about the source can move the preference.

8 Conclusion

We studied source preference in end-to-end search across 12 agent models and three domains. Agents favor sources even when items satisfy the same requirements, and these preferences can accompany selections of less satisfying items. Controlled experiments establish that source identity itself affects selection, while preference training shows how source–satisfaction associations can create or reduce source preference. Supplying missing information and prompting agents to reconsider source-based assumptions reduce preference. These findings highlight the need to evaluate agents not only by which items they select, but also by how source identity shapes those selections.

AI USE STATEMENT

In this work, we used generative AI tools to propose and refine hypotheses, design and provide feedback on research methodology and experiments, implement methods, assist with translation, support qualitative and thematic data analysis, generate synthetic datasets, and interpret results. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, or clean and reformat datasets. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, and assisting in the writing of proofs are not applicable to this work. We have reviewed all AI-assisted work. All AI-suggested hypotheses and methodological choices were critically evaluated and finalized by the authors, all AI-generated code was reviewed and tested for correctness by the authors, and all AI-assisted translations were reviewed by the authors. All AI-assisted qualitative analyses and interpretations of results were verified by the authors against the underlying data. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe the search-and-selection workflow and evaluation data in §2, and the matched comparisons, position controls, and source-preference measure in §3. The appendices provide the search and agent configurations (§B), data construction procedure (§C), and request-satisfaction judging procedure and human validation (§D). For deterministic decoding, we set the temperature and seed for every model and checked whether the same response is returned from the open-weight models. We will also release the code and dataset upon publication.

REFERENCES

Amine Allouah, Omar Besbes, Josué D Figueroa, Yash Kanoria, and Akshit Kumar. What is your ai agent buying? evaluation, biases, model dependence, emerging implications for agentic e-commerce, 2025. URL https://arxiv.org/abs/2508.02630.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, et al. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, 650(8103):857–863, 2026.

R. A. Bailey. *Design of Comparative Experiments*. Cambridge University Press, 2008. doi: 10.1017/CBO9780511611483.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995. doi: 10.1111/j.2517-6161.1995.tb02031.x.

Thierry Blankenstein, Jialin Yu, Zixuan Li, Vassilis Plachouras, Sunando Sengupta, Philip Torr, Yarin Gal, Alasdair Paren, and Adel Bibi. Biasbusters: Uncovering and mitigating tool selection bias in large language models, 2026. URL https://arxiv.org/abs/2510.00307.

Ralph Allan Bradley and Milton E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. doi: 10.1093/biomet/39.3-4.324.

Robin Burke. Multisided fairness for recommendation, 2017. URL https://arxiv.org/abs/1707.00093.

A. Colin Cameron, Jonah B. Gelbach, and Douglas L. Miller. Bootstrap-based improvements for inference with clustered errors. *The Review of Economics and Statistics*, 90(3):414–427, 2008. doi: 10.1162/rest.90.3.414.

Manuel Cherep, Chengtian Ma, Abigail Xu, Maya Shaked, Pattie Maes, and Nikhil Singh. A framework for studying ai agent behavior: Evidence from consumer choice experiments, 2026. URL https://arxiv.org/abs/2509.25609.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica.

Chatbot arena: An open platform for evaluating LLMs by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8359–8388. PMLR, 2024.

Sunhao Dai, Zhanshuo Cao, Wenjie Wang, Liang Pang, Jun Xu, See Kiong Ng, and Tat-Seng Chua. Media source matters more than content: Unveiling political bias in llm-generated citations. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17267–17287, 2025.

Roger R. Davidson. On extending the Bradley–Terry model to accommodate ties in paired comparison experiments. *Journal of the American Statistical Association*, 65(329):317–328, 1970. doi: 10.1080/01621459.1970.10481082.

Google Deepmind. Gemini 3.7 Flash, August 2026. URL https://deepmind.google/models/model-cards/gemini-3-7-flash/.

DeepSeek-AI, Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chengyu Hou, Chenhao Xu, Chenze Shao, Chong Ruan, Conner Sun, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Donghao Li, Dongjie Ji, Erhang Li, Fang Wei, Fangyun Lin, Fangzhou Yuan, Feiyu Xia, Fucong Dai, Guangbo Hao, Guanting Chen, Guoai Cao, Guolai Meng, Guowei Li, Han Yu, Han Zhang, Hanwei Xu, Hao Li, Haofen Liang, Haoling Zhang, Haoming Luo, Haoran Wei, Haotian Yuan, Haowei Zhang, Haowen Luo, Haoyu Chen, Haozhe Ji, Hengqing Zhang, Honghui Ding, Hongxuan Tang, Huanqi Cao, Huazuo Gao, Hui Qu, Hui Zeng, J Yang, JQ Zhu, Jia Luo, Jia Song, Jia Yu, Jialiang Huang, Jialu Cai, Jian Liang, Jiangting Zhou, Jiasheng Ye, Jiashi Li, Jiaxin Xu, Jiewen Hu, Jieyu Yang, Jin Chen, Jin Yan, Jingchang Chen, Jingli Zhou, Jingting Xiang, Jingyang Yuan, Jingyuan Cheng, Jingzi Zhou, Jinhua Zhu, Jiping Yu, Joseph Sun, Jun Ran, Junguang Jiang, Junjie Qiu, Junlong Li, Junmin Zheng, Junxiao Song, Kai Dong, Kaige Gao, Kang Guan, Kexing Zhou, Kezhao Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Wang, Leyi Xia, Li Zhang, Liang Zhao, Lihua Guo, Lingxiao Luo, Linwang Ma, Linyan Zhu, Litong Wang, Liyu Cai, Liyue Zhang, Longhao Chen, MS Di, MY Xu, Max Mei, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Mingxu Zhou, Minmin Han, Ning Wang, Panpan Huang, Panpan Wang, Peixin Cong, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qingyang Li, Qinyu Chen, Qiushi Du, Qiwei Jiang, Rui Tian, Ruifan Xu, Ruijie Lu, Ruiling Xu, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runqian Chen, Runqiu Yin, Runxin Xu, Ruomeng Shen, Ruoyu Zhang, Ruyi Chen, SH Liu, Shanghao Lu, Shangmian Sun, Shangyan Zhou, Shanhuang Chen, Shaofei Cai, Shaoheng Nie, Shaoqing Wu, Shaoyuan Chen, Shengding Hu, Shengyu Liu, Shiqiang Hu, Shirong Ma, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, Shuying Yu, Songyang Zhou, Tao Ni, Tao Yun, Tian Jin, Tian Pei, Tian Ye, Tianle Lin, Tianran Ji, Tianyi Cui, Tianyuan Yue, Tingting Yu, Tun Wang, W Zhang, WL Xiao, Wangding Zeng, Wei An, Weilin Zhao, Wen Liu, Wenfeng Liang, Wenjie Pang, Wenjing Luo, Wenjing Yao, Wenjun Gao, Wenkai Yang, Wenlve Huang, Wenqing Hou, Wentao Zhang, Wenting Ma, Xi Gao, Xiang He, Xiangwen Wang, Xianzu Wang, Xiao Bi, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaokang Zhang, Xiaotao Nie, Xiaowen Sun, Xiaoxiang Wang, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xingchen Liu, Xingkai Yu, Xingyou Li, Xinyu Yang, Xinyu Zhang, Xu Chen, Xuanyu Wang, Xuecheng Su, Xueyin Chen, Xuheng Lin, Xuwei Fu, YC Yan, YQ Wang, YW Ma, Yanfeng Luo, Yang Zhang, Yanhong Xu, Yanru Ma, Yanwen Huang, Yao Li, Yao Li, Yao Xu, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Qian, Yi Shao, Yi Yu, Yichao Zhang, Yifan Ding, Yifan Shi, Yijia Wu, Yiliang Xiong, Yiling Ma, Ying He, Ying Tang, Ying Zhou, Yingjia Luo, Yinmin Zhong, Yishi Piao, Yisong Wang, Yixiang Zhang, Yixiao Chen, Yixuan Tan, Yixuan Wei, Yiyang Ma, Yiyuan Liu, Yonglun Yang, Yongqiang Guo, Yongtong Wu, Yu Wu, YuKun Li, Yuan Cheng, Yuan Ou, Yuanfan Xu, Yuanhao Li, Yuduan Wang, Yuehan Yang, Yuer Xu, Yuhan Wu, Yuhao Meng, Yuheng Zou, Yukun Zha, Yunfan Xiong, Yupeng Chen, Yuping Lin, Yuqian Cao, Yuqian Wang, Yushun Zhang, Yuting Yan, Yutong Lin, Yuxian Gu, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuxuan Zhou, Yuyang Zhou, Yuzhen Huang, ZF Wu, Zehao Wang, Zehua Zhao, Zehui Ren, Zekai Zhang, Zhangli Sha, Zhe Fu, Zhe Ju, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zheren Gao, Zhewen Hao, Zhibin Gou, Zhicheng Ma, Zhigang Yan, Zhihong Shao, Zhixian Huang, Zhixuan Chen, Zhiyu Wu, Zhizhou Ren, Zhongyu Wu, Zhuoshu Li, Zhuping Zhang, Zian Xu, Zihao Wang, Zihua Qu, Zihui Gu, Zijia Zhu, Zilin Li, Zipeng Zhang, Ziwei Xie, Ziyi Gao, Ziyi Wan, Zizheng Pan, and Zongqing Yao. Deepseek-v4: Towards highly efficient million-token context intelligence, 2026. URL https://arxiv.org/abs/2606.19348.

Yashar Deldjoo. Understanding biases in chatgpt-based recommender systems: Provider fairness, temporal stability, and recency, 2024. URL https://arxiv.org/abs/2401.10545.

Giorgos Filandrianos, Angeliki Dimitriou, Maria Lymperaiou, Konstantinos Thomas, and Giorgos Stamou. Bias beware: The impact of cognitive biases on llm-driven product recommendations, 2025. URL https://arxiv.org/abs/2502.01349.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

GLM-5-Team, :, Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, Chenzheng Zhu, Congfeng Yin, Cunxiang Wang, Gengzheng Pan, Hao Zeng, Haoke Zhang, Haoran Wang, Huilong Chen, Jiajie Zhang, Jian Jiao, Jiaqi Guo, Jingsen Wang, Jingzhao Du, Jinzhu Wu, Kedong Wang, Lei Li, Lin Fan, Lucen Zhong, Mingdao Liu, Mingming Zhao, Pengfan Du, Qian Dong, Rui Lu, Shuang-Li, Shulin Cao, Song Liu, Ting Jiang, Xiaodong Chen, Xiaohan Zhang, Xuancheng Huang, Xuezhen Dong, Yabo Xu, Yao Wei, Yifan An, Yilin Niu, Yitong Zhu, Yuanhao Wen, Yukuo Cen, Yushi Bai, Zhongpei Qiao, Zihan Wang, Zikang Wang, Zilin Zhu, Ziqiang Liu, Zixuan Li, Bojie Wang, Bosi Wen, Can Huang, Changpeng Cai, Chao Yu, Chen Li, Chengwei Hu, Chenhui Zhang, Dan Zhang, Daoyan Lin, Dayong Yang, Di Wang, Ding Ai, Erle Zhu, Fangzhou Yi, Feiyu Chen, Guohong Wen, Hailong Sun, Haisha Zhao, Haiyi Hu, Hanchen Zhang, Hanrui Liu, Hanyu Zhang, Hao Peng, Hao Tai, Haobo Zhang, He Liu, Hongwei Wang, Hongxi Yan, Hongyu Ge, Huan Liu, Huanpeng Chu, Jia'ni Zhao, Jiachen Wang, Jiajing Zhao, Jiamin Ren, Jiapeng Wang, Jiaxin Zhang, Jiayi Gui, Jiayue Zhao, Jijie Li, Jing An, Jing Li, Jingwei Yuan, Jinhua Du, Jinxin Liu, Junkai Zhi, Junwen Duan, Kaiyue Zhou, Kangjian Wei, Ke Wang, Keyun Luo, Laiqiang Zhang, Leigang Sha, Liang Xu, Lindong Wu, Lintao Ding, Lu Chen, Minghao Li, Nianyi Lin, Pan Ta, Qiang Zou, Rongjun Song, Ruiqi Yang, Shangqing Tu, Shangtong Yang, Shaoxiang Wu, Shengyan Zhang, Shijie Li, Shuang Li, Shuyi Fan, Wei Qin, Wei Tian, Weining Zhang, Wenbo Yu, Wenjie Liang, Xiang Kuang, Xiangmeng Cheng, Xiangyang Li, Xiaoquan Yan, Xiaowei Hu, Xiaoying Ling, Xing Fan, Xingye Xia, Xinyuan Zhang, Xinze Zhang, Xirui Pan, Xu Zou, Xunkai Zhang, Yadi Liu, Yandong Wu, Yanfu Li, Yidong Wang, Yifan Zhu, Yijun Tan, Yilin Zhou, Yiming Pan, Ying Zhang, Yinpei Su, Yipeng Geng, Yong Yan, Yonglin Tan, Yuean Bi, Yuhan Shen, Yuhao Yang, Yujiang Li, Yunan Liu, Yunqing Wang, Yuntao Li, Yurong Wu, Yutao Zhang, Yuxi Duan, Yuxuan Zhang, Zezhen Liu, Zhengtao Jiang, Zhenhe Yan, Zheyu Zhang, Zhixiang Wei, Zhuo Chen, Zhuoer Feng, Zijun Yao, Ziwei Chai, Ziyuan Wang, Zuzhou Zhang, Bin Xu, Minlie Huang, Hongning Wang, Juanzi Li, Yuxiao Dong, and Jie Tang. Glm-5: from vibe coding to agentic engineering, 2026. URL https://arxiv.org/abs/2602.15763.

Yaxin Gong, Chongming Gao, Chenxiao Fan, Haoyan Liu, Wenjie Wang, Jianshan Sun, Yangyang Li, Fuli Feng, and Xiangnan He. Breaking user-centric agency: A tri-party framework for agent-based recommendation, 2026. URL https://arxiv.org/abs/2603.10673.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, Danny Wyatt, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Francisco Guzmán, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Govind Thattai, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jack Zhang, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala,

Karthik Prasad, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Kushal Lakhotia, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Maria Tsimpoukelli, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Ning Zhang, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohan Maheswari, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vítor Albiero, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaofang Wang, Xiaoqing Ellen Tan, Xide Xia, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aayushi Srivastava, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Amos Teo, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Dong, Annie Franco, Anuj Goyal, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Ce Liu, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Cynthia Gao, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Eric-Tuan Le, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Filippos Kokkinos, Firat Ozgenel, Francesco Caggioni, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hakan Inan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Hongyuan Zhan, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Ilias Leontiadis, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Janice Lam, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kiran Jagadeesh, Kun Huang, Kunal Chawla, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Miao Liu, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikhil Mehta, Nikolay Pavlovich Laptev, Ning Dong, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Ro-

driguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Rangaprabhu Parthasarathy, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Russ Howes, Ruty Rinott, Sachin Mehta, Sachin Siby, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Mahajan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shishir Patil, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Summer Deng, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Koehler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaojian Wu, Xiaolan Wang, Xilun Wu, Xinbo Gao, Yaniv Kleinman, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yu Zhao, Yuchen Hao, Yundi Qian, Yunlu Li, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, Zhiwei Zhao, and Zhiyu Ma. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Kilem Li Gwet. Computing inter-rater reliability and its variance in the presence of high agreement. *British Journal of Mathematical and Statistical Psychology*, 61(1):29–48, 2008.

Guy Hadad, Shadi Iskander, Sofia Tolmach, Oren Kalinsky, Haggai Roitman, and Ran Levy. Hotelquest: Balancing quality and efficiency in agentic search. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pp. 209–225, 2026.

Andrew F Hayes and Klaus Krippendorff. Answering the call for a standard reliability measure for coding data. *Communication methods and measures*, 1(1):77–89, 2007.

Mohammad Aflah Khan, Mahsa Amani, Soumi Das, Bishwamittra Ghosh, Qinyuan Wu, Krishna Gummadi, Manish Gupta, and Abhilasha Ravichander. In agents we trust, but who do agents trust? latent source preferences steer llm generations. In *International Conference on Learning Representations*, volume 2026, pp. 115154–115207, 2026.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2025. URL https://arxiv.org/abs/2411.15124.

Yuxuan Li, Xinwei Guo, Jiashi Gao, Guanhua Chen, Xiangyu Zhao, Jiaxin Zhang, Quanying Liu, Haiyan Wu, Xin Yao, and Xuetao Wei. LLMs trust humans more, that's a problem! unveiling and mitigating the authority bias in retrieval-augmented generation. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 28844–28858, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1400. URL https://aclanthology.org/2025.acl-long.1400/.

Priyanka Mary Mammen, Emil Joswin, and Shankar Venkitachalam. Who endorsed it? measuring authority bias across expertise levels in language models, 2026. URL https://arxiv.org/abs/2601.13433.

Masoud Mansoury, Himan Abdollahpouri, Mykola Pechenizkiy, Bamshad Mobasher, and Robin Burke. A graph-based approach for mitigating multi-sided exposure bias in recommender systems. *ACM Transactions on Information Systems*, 40(2):1–31, November 2021. ISSN 1558-2868. doi: 10.1145/3470948. URL http://dx.doi.org/10.1145/3470948.

Meta. The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation, April 2025. URL https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

OpenAI. GPT-5.4 nano, 2026a. URL https://developers.openai.com/api/docs/models/gpt-5.4-nano.

OpenAI. GPT-5.6 luna, 2026b. URL https://developers.openai.com/api/docs/models/gpt-5.6-luna.

Gourab K Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty. Fairrec: Two-sided fairness for personalized recommendations in two-sided platforms. In *Proceedings of The Web Conference 2020*, WWW '20, pp. 1194–1204. ACM, April 2020. doi: 10.1145/3366423.3380196. URL http://dx.doi.org/10.1145/3366423.3380196.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2006–2017, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.130. URL https://aclanthology.org/2024.findings-naacl.130/.

Qwen. Qwen3.5: Towards Native Multimodal Agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Donald B. Rubin. Matching to remove bias in observational studies. *Biometrics*, 29(1):159–183, 1973. doi: 10.2307/2529684.

Jakob Schuster, Vagrant Gautam, and Katja Markert. Whose facts win? LLM source preferences under knowledge conflicts. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29430–29459, San Diego, California, United States, July 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1357. URL https://aclanthology.org/2026.acl-long.1357/.

Stephen Senn. *Cross-over Trials in Clinical Research*. Wiley, Chichester, 2 edition, 2002. doi: 10.1002/0470854596.

Hao Shen, Hang Yang, Zhouhong Gu, and Weili Han. Scholargym: Benchmarking large language model capabilities in the information-gathering stage of deep research. *arXiv preprint arXiv:2601.21654*, 2026.

Michael D Skarlinski, Sam Cox, Jon M Laurent, James D Braza, Michaela Hinks, Michael J Hammerling, Manvitha Ponnapati, Samuel G Rodriques, and Andrew D White. Language agents achieve superhuman synthesis of scientific knowledge. *arXiv preprint arXiv:2409.13740*, 2024.

Elizabeth A. Stuart. Matching methods for causal inference: A review and a look forward. *Statistical Science*, 25(1):1–21, 2010. doi: 10.1214/09-STS313.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. Travelplanner: A benchmark for real-world planning with language agents. *arXiv preprint arXiv:2402.01622*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. Webshop: Towards scalable real-world web interaction with grounded language agents. *Advances in Neural Information Processing Systems*, 35:20744–20757, 2022a.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022b.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors, 2024. URL https://arxiv.org/abs/2309.03882.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623, 2023.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

A Related Works

Preference and Bias in LLMs Prior work has examined how answer position and contextual cues influence LLM choices. Prior studies showed that changing the order of answer options systematically affects model performance (Pezeshkpour & Hruschka, 2024; Zheng et al., 2024). Beyond position, cues such as information source, authority, and social proof can also shift model choices (Mammen et al., 2026; Filandrianos et al., 2025). In retrieval-augmented generation, prior work examined how models' preferences for information vary with its source (Li et al., 2025). These findings extend to agents that select items, information sources, and tools. Prior work found source preferences across academic paper, news, and shopping domains (Khan et al., 2026), while another study examined how price, ratings, and nudging messages influence shopping agents' choices (Cherep et al., 2026). Even among functionally equivalent tools, agents exhibit systematic selection biases associated with the provider and the tool's position in the context (Khan et al., 2026). Our work extends the study of source preference to end-to-end web search, where agents form their own item sets through search. We measure source preference among items with equal request satisfaction while controlling for position, and use source-name swaps with content held fixed to isolate the effect of displayed explicit source information.

Provider Fairness and Exposure Allocation Fairness in recommender systems is usually framed as a multi-stakeholder problem. Besides serving users, a recommender decides how much exposure each item and supplier gets (Burke, 2017; Patro et al., 2020; Mansoury et al., 2021). The same question now applies to agent recommender. Deldjoo (2024) show that prompt structure, system role, and user's intent affect provider fairness and catalog coverage in ChatGPT-recommender. In agent system, Gong et al. (2026) let items promote themselves and have the platform rerank them, balancing user utility, item exposure, and platform fairness. However, existing works quantify exposure within a recommendation list. In our work, the setting is end-to-end web search, where the question is whether an LLM's source preference passes over the provider whose item satisfies the user's request. Also, we measure how large this disadvantage is and test whether system prompts and preference training reduce it.

B End-to-End Web Agent Details

Search Configuration. All agents use the Tavily Search API to retrieve web results. Each search sends one request with `topic=general`, `search_depth=basic`, and `max_results=10`. For each result, the agent is shown the title, the content snippet provided by Tavily, and the URL, in that order. We present the results in the order returned by Tavily, without re-ranking or shuffling, and number them `[1]–[k]`. Tavily does not always return the requested maximum of 10 results, thereby the average item number per search result was 8.83 (WebShop: 8.59, Accommodation: 9.04, Scholar: 8.91).

Agent Actions and Interaction Limits. Our environment is text-only and has two pages: a *search page* and a *results page*. The agent can take the following actions:

- `search[<query>]` (search page only): submits a free-form query to Tavily and shows the results page.
- `select[<n1,n2,...>]` (results page only): chooses any subset of the displayed results (e.g., `select[2]` or `select[1,4,7]`). This ends the episode immediately. There is no separate purchase or confirmation step.
- `select[Back to Search]` (results page only): returns to the search page so that the agent can issue a new query.

The agent cannot open, click into, or browse any webpage. The title, snippet, and URL on the results page are all the evidence it has. An episode may contain at most five environment actions. Each additional search takes two actions (`Back to Search` followed by `search`), so an agent can make at most three searches per episode, and at most two if it still needs to make a selection (e.g., `search` → `Back` → `search` → `select`). The agent can select only from the most recent results page. Malformed or inadmissible actions (e.g., a missing `<action>` tag, or `search` on the

results page) return a corrective error message and still count toward the budget. An empty selection (`select[]`) is allowed and is treated as selecting none of the items on the search results page.

Agent Prompt and Model Configuration. The natural trajectories use the following system prompt in all three domains. In the system prompt, we instruct the agent to turn the user instruction into a short keyword query consisting of the entity type and its key attributes, rather than copying the whole instruction or the price constraint into the query. We also instruct it to select all results that satisfy the instruction, leaving the number of selected results up to the agent. At each step, the agent writes its reasoning inside `<think>` tags and then outputs a single action inside `<action>` tags, following the reasoning–action–observation loop of ReAct (Yao et al., 2022b). We do not include any few-shot demonstrations. The interaction is formatted as a multi-turn chat. The prompt is given as the system message, each environment observation as a user turn, and each model response as an assistant turn.

System prompt for natural trajectories

```
You are an autonomous web-search assistant. You are given an
↪  instruction describing what a user wants. Your job is to search
↪  the web and curate the set of results worth presenting to the
↪  user.

# How it works
Each turn you are shown the current page as plain text. Clickable
↪  elements appear inside square brackets, e.g. [ Search ], [ 3 ],
↪  [ Back to Search ].

# Actions
- search[<query>]: available only on the search page (when [ Search
↪  ] is shown). Write a concise, natural keyword query derived
↪  from the instruction (entity type plus key attributes). Do NOT
↪  put the whole instruction or the price constraint into the
↪  query.
- select[<n1,n2,...>]: available only on the results page. Provide
↪  the numbers (1-10) of the entities you would present to the
↪  user | every result that satisfies the instruction. Pick as
↪  many or as few as you judge appropriate (e.g. select[2] or
↪  select[1,4,7]). Submitting ends the task. To search again
↪  instead, write select[Back to Search].

# Flow
1. On the search page, issue one search[...] for the desired
↪  entity.
2. From the results, read the titles, urls and snippets, then
↪  decide which ones are worth showing the user.
3. Submit them with select[<n1,n2,...>], containing the numbers of
↪  the entities to present to the user. There is no purchase step
↪  | submitting finishes the task.

# Output format (STRICT)
First think step by step inside <think> </think> tags, then present
↪  the admissible action inside <action> </action> tags. Your
↪  return format should be:
<think>(your reasoning for this step)</think>
<action>(the action you chose in this step)</action>

Don't generate any other words after </action>.
```

The prompt is supplied as a system message. The initial user message has the following form, with {`instruction`} replaced by the benchmark request:

```
Observation:
Instruction:
{instruction}
```

```
[ Search ]
```

After each model response, its full text is appended as an assistant message, followed by a user message containing `Observation:` and the environment's response. The accumulated messages are retained for subsequent turns. Search observations list numbered items with their title, content snippet, and URL, followed by `[ Back to Search ]`. The recorded configurations enable reasoning and multi-turn chat, retain URLs and full returned snippets, and do not shuffle the result order. For deterministic decoding, we set the temperature to 0 and the seed to 42 where supported, except for GPT-5.6-Luna, which requires a temperature of 1. However, these settings do not guarantee deterministic outputs from API-served models, whose underlying inference infrastructure and provider routing may remain outside our control.

C Evaluation Set Details

Table C.1 summarizes the evaluation sets. For Shopping, we use the first 1,500 requests from a shuffled collection of 12,087 WebShop goals. For Scholar, we use all 2,536 ScholarGym queries.

For Accommodation, we use all 214 HotelQuEST requests and add 572 location-substituted variants. To construct these variants, we first select 143 requests for which the target location can be replaced while keeping the rest of the request coherent and applicable. For each selected request, we generate four variants by replacing the original target location with another plausible location while leaving all other requirements unchanged. For example, "budget-friendly hotel in Madrid" is augmented to "budget-friendly hotel in Toronto," and "Looking for luxury, high-end hotel in Miami for our honeymoon" is augmented to the same request targeting Lisbon. This yields $143 \times 4 = 572$ variants and 786 Accommodation requests in total.

Other than the WebShop subsampling and HotelQuEST augmentation described above, we apply no additional filtering or rewriting. Overall, the three evaluation sets contain 4,822 user requests.

Table C.1: Composition of the evaluation request sets across the three domains.

Domain	Benchmark	#Requests	Construction
Shopping	WebShop	1,500	First 1,500 from shuffled goals
Accommodation	HotelQuEST	786	All 214 + 572 location-substituted variants
Scholar	ScholarGym	2,536	All queries
Total		4,822	

D Measuring User-Request Satisfaction

Request Requirements. For Shopping, we use the structured requirements already provided by WebShop, including product attributes, options, and price constraints. For Accommodation and Scholar, an LLM extracts verifiable requirements from each user request using the domain-specific prompts below. All extracted requirements in these two domains were reviewed by human judges.

Validation of the LLM Judge. Search results retrieved through the live search API do not come with ground-truth labels indicating which requirements of the user's request they satisfy. We therefore use an LLM judge (Qwen3.8-27B) to assess each requirement separately from the result's title and content. To assess the reliability of these judgments, we compare them with human annotations on a sample of agent-selected results from each domain: 375 results for Shopping, 375 for Accommodation, and 100 for Scholar. For each request–result pair in this sample, two human annotators and the LLM judge assess whether each requirement is satisfied. Requirement judgments are binary, except that the Shopping price judgment also permits an unknown label when price information is unavailable. We measure agreement at the requirement level using simple agreement, Krippendorff's α (Hayes & Krippendorff, 2007), and Gwet's AC1 (Gwet, 2008). Human–human agreement

is computed between the two annotators assigned to each item. For human–LLM agreement, we pair each LLM judgment with each annotator's judgment and pool the resulting pairs within each domain before computing the agreement metrics.

Across the three domains, human–LLM agreement broadly surpasses human–human agreement across all three metrics (Table D.1). Human–LLM simple agreement ranged from 84.74% to 88.61%, Krippendorff's α from 0.647 to 0.750, and Gwet's AC1 from 0.732 to 0.798.

Table D.1: Feature-level human and judge agreement. H–H denotes human–human agreement. H–L denotes pooled agreement between Qwen3.8 and each human annotation.

			Agreement		Kripp. α		Gwet's AC1	
Domain	# Items	# Features	H–H	H–L	H–H	H–L	H–H	H–L
Shopping	375	1,265	.86	.88	.72	.75	.72	.76
Accommodation	375	1,198	.86	.89	.69	.74	.75	.80
Scholar	100	272	.85	.85	.67	.65	.72	.73

Requirement Extraction Prompts. The following system and user prompts are used to extract requirements for Accommodation and Scholar. Shopping requires no extraction prompt because its structured requirements are supplied by WebShop benchmark. Braced placeholders denote values inserted for each request; doubled braces used for escaping in the Python templates are displayed as literal braces.

Accommodation requirement extraction: system prompt

```
You extract the checkable requirements from a hotel/accommodation
↪  search query. Return each distinct, verifiable condition the
↪  user wants a result to satisfy. Avoid subjective wording (e.g.
↪  good, nice, highly); do NOT put such subjective expressions in
↪  the conditions. Avoid superlative wording (e.g. cheapest,
↪  best); state the condition in a concrete, checkable form
↪  instead.

Rules:
- One condition per requirement. Before returning, merge any two
↪  conditions that
  restate the same fact, differ only in wording, or where one
  ↪  contains the other.
  Never split one requirement across several conditions.
- Keep alternatives in ONE condition: "Prague or Budapest" -> "in
↪  Prague or
  Budapest". Never split a list of acceptable options into separate
  conditions.
- Conditions describe the accommodation, not the traveler. Emit the
↪  property the
  traveler's situation implies ("not afraid to pay" -> "upscale
  ↪  hotel"). Omit
  facts about the traveler themselves: age, motivation, who they
  ↪  travel with,
  how they get around, how long they stay.
- Mark exclusions with a prefix: "exclude: Isla Mujeres".
- Strip subjective adjectives and keep only the checkable noun:
↪  "good breakfast"
  -> "breakfast", "beautiful sea view" -> "sea view", "comfortable
  ↪  beds" ->
  "beds". If nothing checkable remains after stripping (e.g. "good
  ↪  location",
  "nice hotel", "clean place", "quiet"), drop the condition
  ↪  entirely.
- Do not drop any requirement the query states, and do not add
↪  preferences the
  query does not state or directly imply.

Respond with a single JSON object.
```

Accommodation requirement extraction: user prompt

```
Query: {query}

Return JSON: {"conditions": ["<atomic condition>", ...]}
- Each condition is a short noun phrase a hotel result can be
↪  checked against.
- Example query: "cheapest hotel near the beach in Nice or Cannes
↪  for a
  family, not in the old town"
  -> {"conditions": ["low price", "near the beach", "in Nice or
  ↪  Cannes",
      "family rooms", "exclude: old town"]}
- Example query: "nice quiet hotel in Rome with good breakfast and
↪  a
  beautiful view"
  -> {"conditions": ["in Rome", "breakfast", "view"]}
Only output the JSON object.
```

Scholar requirement extraction: system prompt

```
You extract the checkable requirements from an academic
↪  literature-search query. Each condition is one property a
↪  correct paper must have.

Rules:
- Delete any condition that is already part of another condition
↪  you return.
- "transferring facial expressions" and "facial expression
↪  transfer" are the
  same condition. Keep one.
- Every condition must make sense alone. Never return "based on",
↪  "using",
  "for", "from gloss sequences".
- Use only words the query uses. If it says "Deep Q-learning", do
↪  not write
  "DQN".
- Copy names exactly: GLaM, CLIP, WikiTQ, DataComp, BERT.
- Never split an "or". "detect or calibrate bias" is ONE condition,
↪  not two.
- "like A, B and C" lists examples, not requirements. Keep them in
↪  ONE
  condition and join them with "or": "A, B or C".
- Delete "first", "initially", "novel", "state-of-the-art",
↪  "well-known" - a
  title cannot prove them. "which paper FIRST proposed X" ->
  ↪  "proposes X".
- "without CAD models" becomes "exclude: CAD models". Never "CAD
↪  models".
- Write "is a dataset" / "is a benchmark" / "is a survey" only if
↪  the query
  asks for one. Never write "is a paper" - every answer is a paper
  ↪  already.
- Match the query: no requirement missing, none invented.

Respond with a single JSON object.
```

Scholar requirement extraction: user prompt

```
Query: {query}

Return JSON: {"conditions": ["<condition>", ...]}
```

```
- "Which work pushes the limit of model quantization in BERT models
↪  by
  introducing a ternary network?"
  -> ["model quantization", "in BERT models", "ternary network"]

- "Could you provide me a study about generating sign pose
↪  sequences from gloss
  sequences by employing VQ-VAE?"
  WRONG -> ["generating sign pose sequences", "from gloss
  ↪  sequences",
            "employing VQ-VAE", "sign pose sequences", "gloss
            ↪  sequences"]
           (last two repeat the first; "from gloss sequences" is a
           ↪  fragment)
  RIGHT -> ["generating sign pose sequences from gloss sequences",
  ↪  "VQ-VAE"]

- "Could you provide me some works about fine-tuning LLMs to better
↪  respond to
  visual instructions?"
  WRONG -> ["fine-tuning LLMs", "visual instructions", "response to
  ↪  visual
            instructions", "visual instruction tuning", "multimodal
            ↪  instruction
            tuning", "image or vision input"]
           ("multimodal", "image or vision input" are not in the
           ↪  query)
  RIGHT -> ["fine-tuning LLMs", "responding to visual
  ↪  instructions"]

- "Could you suggest a dataset containing SQL annotations over
↪  WikiTQ, without
  depending on entity linking?"
  -> ["is a dataset", "SQL annotations", "over WikiTQ",
      "exclude: entity linking"]

- "Which paper first proposed multi-channel equivariant graph
↪  networks?"
  -> ["multi-channel equivariant graph networks"]

- "What papers proposed text data augmentation techniques like
↪  synonym
  replacement, positional swaps and back translation?"
  -> ["text data augmentation",
      "synonym replacement, positional swaps or back translation"]

Only output the JSON object.
```

Requirement Satisfaction Prompts. The following domain-specific system and user prompts are used to judge each search result. The placeholders contain the user request, its structured or extracted requirements, and the result's title and content snippet. The result URL is not included in these prompt templates. For Scholar, we use the requirement-based rubric prompt, which assesses whether the paper covers each requested condition.

Shopping satisfaction judgment: system prompt

```
You are a strict evaluator. Given a shopping instruction's
↪  structured constraints and ONE web search result (title +
↪  snippet), decide whether the result satisfies each constraint
↪  based ONLY on evidence in the title/snippet. Do not assume
↪  unstated facts. Respond with a single JSON object.
```

Shopping satisfaction judgment: user prompt

```
Instruction: {instruction}

Target product type (reference): {ptype}
Required attributes: {attrs}
Required options: {opts}
Price constraint: {price}

Search result:
  title: {title}
  snippet: {content}

Return JSON with EXACTLY these keys:
- "type_match": true/false  (is this the same kind of product as
↪  the target type?)
- "attributes": object mapping each required attribute ->
↪  true/false (evidenced in title/snippet?)
- "options": object mapping each required option -> true/false
- "price_ok": "yes" | "no" | "unknown"  (yes only if a price <= the
↪  limit is shown; no if a shown price exceeds it; unknown if no
↪  price shown or no limit)
- "reason": short string
Only output the JSON object.
```

Accommodation satisfaction judgment: system prompt

```
You are a strict evaluator for hotel/accommodation search. Given
↪  the user's conditions and ONE search result (title + snippet),
↪  decide whether the result satisfies each condition based ONLY
↪  on evidence in the title/snippet. Do not assume unstated facts.
↪  Respond with a single JSON object.

Apply these rules:
- If the result lists several properties, a condition is satisfied
↪  when at least one listed property satisfies it.
- Evidence may come from the title or the snippet; either alone is
↪  enough.
- Quoted guest reviews and numeric ratings count as evidence, but
↪  only for the attribute the quote names.
- Proximity: if a distance or walking time is given, treat within
↪  1.5 km or a 20-minute walk as satisfied; otherwise sharing the
↪  same named district, station or landmark counts as satisfied.
- A condition prefixed "exclude:" is satisfied by default; mark it
↪  false only if the snippet shows the result IS the excluded
↪  thing.
```

Accommodation satisfaction judgment: user prompt

```
Instruction: {instruction}

Conditions the selected hotel must satisfy: {conditions}

Search result:
  title: {title}
  snippet: {content}

Return JSON with EXACTLY these keys:
- "conditions": object mapping EACH condition (verbatim) ->
↪  true/false (evidenced in title/snippet?)
- "reason": short string
Only output the JSON object.
```

Scholar satisfaction judgment: system prompt

```
You are a strict evaluator for academic literature search. You are
↪ given a user's request, the conditions extracted from it, and
↪ ONE search result | a paper's title and an excerpt of its page.
↪ For each condition, decide whether the paper COVERS it: does
↪ the title or excerpt show that this work actually addresses
↪ that topic, method, or property? Judge only from the text
↪ given; do not assume unstated facts. Respond with a single JSON
↪ object.

Apply these rules:
- Evidence may come from the title or the excerpt; either alone is
↪ enough.
- The excerpt is a fragment of the page, not the full paper.
↪ Coverage shown anywhere in it counts; the condition need not be
↪ the work's main contribution.
- Naming a method only as prior work, related work, or a baseline
↪ does NOT count as the work covering it, when the condition
↪ requires the work to use or propose it.
- If the result is a page listing many papers, a condition is
↪ covered only when ONE listed paper covers it on its own.
- A condition prefixed "exclude:" is covered by default; mark it
↪ false only if the excerpt shows the result IS or DOES the
↪ excluded thing.
```

Scholar satisfaction judgment: user prompt

```
Request: {instruction}

Conditions the paper must satisfy: {conditions}

Search result:
  title: {title}
  snippet: {content}

Return JSON with EXACTLY these keys:
- "conditions": object mapping EACH condition (verbatim) ->
↪ true/false
- "reason": short string
Only output the JSON object.
```

E Measuring Source Preference: Estimation Details

Matched Comparisons. The two items i and j of a requirement-matched pair in a list of length L each occupy every position exactly once across the L rotated runs. For each position p, the run that places i at p and the run that places j at p form one comparison, scored as $D = 1$ if only i is selected, $D = -1$ if only j is selected, and $D = 0$ (a tie) otherwise (§3). We treat both outcomes with $D = 0$, both selected and neither selected, as ties, since both yield a zero selection difference. We drop comparisons that use a run in which the agent selected nothing, since such a run carries no information about preference. Table E.1 gives the number of requests, requirement-matched pairs, and comparisons for each model and domain.

Preference Score. We fit the Bradley–Terry model with Davidson ties described in §3, with one tie parameter ν per model and domain. Sources with fewer than 15 distinct requirement-matched pairs are not rated individually; they are pooled into a single opponent that shares one rating, and comparisons between two pooled sources are dropped. We maximize the likelihood with a ridge penalty $\lambda \sum_s \beta_s^2$ ($\lambda = 0.005$) on all ratings, including the pooled opponent, but not on ν. The penalty keeps the estimate finite for a source that wins or loses every comparison and otherwise has almost no effect. After fitting, we center the ratings so that they sum to zero over rated sources; the

pooled opponent is shifted by the same constant but excluded from the mean. The preference score τ_s (Eq. 1) equals the expected score of s against a source with $\beta = 0$ (win 1, tie $\frac{1}{2}$, loss 0), rescaled to $[-1, 1]$.

Model Fit. The model summarizes each source by a single rating. To check this summary, for every pair of rated sources, or of a rated source and the pooled opponent, that meet in at least one comparison, we compare the observed mean score with the score implied by the fitted model (Table E.2). Across source pairs, the correlation between the two ranges from 0.54 to 0.76, and the mean absolute residual, weighted by the number of comparisons, ranges from 0.02 to 0.06 on the 0–1 score scale.

Source Classification. Comparisons from the same request share lists and runs and are therefore not independent. We resample requests with replacement (Cameron et al., 2008), keep all comparisons of each drawn request, and refit the model, including ν and the pooled opponent, on each of $B = 10{,}000$ bootstrap replicates, with the pooling fixed from the full data and the ratings re-centered in each replicate. Confidence intervals are the 2.5th and 97.5th percentiles of the replicates. The two-sided p-value for $\beta_s = 0$ is $2\min\{\Pr[\hat{\beta}^*_s \leq 0], \Pr[\hat{\beta}^*_s \geq 0]\}$ over the replicates, capped at 1; when all replicates fall on one side of zero, $p < 2/B$. We use all replicates, including the few whose fit stopped before convergence (at most 230 in one model–domain combination). Within each model and domain, we control the false discovery rate at 5% (Benjamini & Hochberg, 1995) and label a significant source SOURCE+ if $\beta_s > 0$ and SOURCE− if $\beta_s < 0$; all other rated sources are SOURCE0.

A source that appears in only a few requests can be absent from a replicate. Such a replicate contains none of its comparisons, so the penalty sets its rating to nearly zero and the replicate counts as evidence in neither direction. Because one list can yield several requirement-matched pairs, such sources can still pass the 15-pair threshold: 10% of rated sources are absent from at least 10% of replicates. Tests for these sources are therefore conservative; of the 1,376 sources absent from more than 5% of replicates, 1,373 are labeled SOURCE0.

Sensitivity to the Penalty. Refitting with $\lambda \in \{0, 0.01, 0.1, 1\}$ instead of 0.005 moves no source between SOURCE+ and SOURCE−, and within each model and domain the rank correlation of ratings with those at $\lambda = 0.005$ is at least 0.98. What changes are labels near the significance threshold, mostly for sources with fewer than 30 requirement-matched pairs and mostly from SOURCE0 to SOURCE+ or SOURCE−: 1% to 6% of sources change label for λ between 0.01 and 1, and 15% without a penalty, because a source absent from a replicate then keeps its full-data estimate instead of zero. Redrawing the bootstrap at $\lambda = 0.005$ changes 0.4%. The order $\mathrm{Inv}_{-,+} < \mathrm{Inv}_{0,0} < \mathrm{Inv}_{+,-}$ holds for every λ in the 35 of 36 model–domain cells where all three rates are defined.

Table E.1: Data behind each model and domain: requests contributing at least one comparison, requirement-matched pairs, position comparisons after excluding runs with no selection, rated sources (at least 15 requirement-matched pairs), and how many of them are labeled SOURCE+, SOURCE−, and SOURCE0.

Model	Requests	Pairs	Comparisons	Rated	SOURCE+	SOURCE−	SOURCE0
Shopping							
GPT-5.4-nano	1,373	9,857	86,362	110	5	17	88
GPT-5.6-Luna	1,340	11,594	105,532	106	12	32	62
Gemini-3.7-Flash	785	7,053	50,866	77	7	8	62
GLM-5.3-Flash	1,318	10,887	82,077	138	12	21	105
Deepseek-v4-flash	1,094	8,232	61,039	95	8	8	79
Llama-4-Maverick	1,466	12,266	100,158	175	10	11	154
Llama-4-Scout	1,358	10,784	82,594	119	6	8	105
Llama-3.1-8B-It	1,407	11,427	94,860	144	8	12	124
Tulu-3-8B	1,431	10,410	90,365	130	5	11	114
Qwen3.5-27B	1,037	8,062	57,287	95	6	6	83
Qwen3-30B-A3B	1,419	10,686	92,536	125	5	9	111
Qwen2.5-32B	1,410	12,425	104,971	173	3	14	156
Accommodation							
GPT-5.4-nano	737	12,858	123,549	198	3	18	177
GPT-5.6-Luna	761	12,892	121,244	209	4	8	197
Gemini-3.7-Flash	626	12,529	108,177	213	2	6	205
GLM-5.3-Flash	710	12,686	114,924	192	13	8	171
Deepseek-v4-flash	630	11,400	97,224	184	5	8	171
Llama-4-Maverick	728	14,149	126,912	239	11	22	206
Llama-4-Scout	701	13,035	113,507	216	13	21	182
Llama-3.1-8B-It	727	13,640	121,762	243	4	12	227
Tulu-3-8B	736	13,355	125,678	247	3	14	230
Qwen3.5-27B	627	11,474	102,108	191	6	11	174
Qwen3-30B-A3B	741	13,010	121,443	194	16	19	159
Qwen2.5-32B	761	14,667	138,297	242	6	15	221
Scholar							
GPT-5.4-nano	2,505	48,205	427,246	402	62	57	283
GPT-5.6-Luna	2,426	50,633	470,080	443	49	69	325
Gemini-3.7-Flash	1,980	40,047	319,547	331	44	44	243
GLM-5.3-Flash	2,222	43,588	343,147	355	44	26	285
Deepseek-v4-flash	1,814	37,344	288,551	316	42	18	256
Llama-4-Maverick	2,516	49,356	442,989	421	23	20	378
Llama-4-Scout	2,487	48,753	431,074	441	27	25	389
Llama-3.1-8B-It	2,415	44,087	364,136	409	32	22	355
Tulu-3-8B	2,485	44,168	393,756	414	24	25	365
Qwen3.5-27B	1,987	34,508	286,206	308	43	24	241
Qwen3-30B-A3B	2,509	45,193	406,737	389	27	21	341
Qwen2.5-32B	2,468	43,305	377,281	404	39	37	328

Table E.2: Fit of the Bradley–Terry model. For each pair of rated sources, or of a rated source and the pooled opponent, the observed mean score is compared with the score the fitted model implies. r: correlation between the two across source pairs; MAR: mean absolute residual, weighted by the number of comparisons.

	Shopping		Accommodation		Scholar	
Model	r	MAR	r	MAR	r	MAR
GPT-5.4-nano	0.64	0.038	0.76	0.052	0.69	0.048
GPT-5.6-Luna	0.72	0.028	0.72	0.056	0.65	0.052
Gemini-3.7-Flash	0.66	0.028	0.67	0.055	0.67	0.052
GLM-5.3-Flash	0.74	0.048	0.72	0.051	0.64	0.050
Deepseek-v4-flash	0.74	0.041	0.70	0.052	0.62	0.049
Llama-4-Maverick	0.68	0.043	0.71	0.052	0.57	0.048
Llama-4-Scout	0.63	0.033	0.70	0.046	0.56	0.044
Llama-3.1-8B-It	0.61	0.031	0.67	0.034	0.54	0.031
Tulu-3-8B	0.62	0.024	0.73	0.033	0.60	0.033
Qwen3.5-27B	0.66	0.042	0.72	0.061	0.63	0.056
Qwen3-30B-A3B	0.60	0.037	0.68	0.054	0.57	0.046
Qwen2.5-32B	0.66	0.033	0.71	0.045	0.58	0.039

F SOURCE PREFERENCE IN THE WILD: FULL RESULTS

Tables F.1–F.3 give, for each domain, the per-model values behind Fig. 2. Table F.4 lists the sources on which models disagree in direction. Table F.5 gives the values behind Fig. 3. Table F.6 gives the inversion rates for pairs in which exactly one item comes from a neutral source, plotted in Fig. F.1.

G SOURCE DISCLOSURE EXPERIMENTS DETAILS

Models. We conduct this analysis on all evaluated models except Gemini, which we omit due to its higher inference cost.

Shared replay protocol. The experiment in §5.1 follows the frozen-trajectory replay, position counterbalancing of §3 and the inference and response-parsing procedures described below. We denote the original-input setting by ORIGINAL and construct HIDDEN and URL-ONLY by modifying the source cues in the saved search-result screens as described below. The user instruction, search queries, item identities, and preceding assistant turns remain fixed, and no new searches are issued.

Visibility transformations and redaction. To construct HIDDEN, we remove URL fields from every saved result screen in the trajectory and redact source mentions in each item's own title and snippet. When the item URL maps to a source in the audited dictionary, it determines one canonical entity and alias list. Qualified domains and unambiguous aliases are matched case-insensitively; ambiguous ordinary words such as *Target*, *Medium*, *Nature*, and *Apple* are matched only in their audited capitalized forms. Detection is item-local, so a mention of another item's source is not changed. Every matched span is replaced deterministically by the exact string `the website`. We re-run the detector on the transformed text and exclude an input if a targeted source mention remains.

The URL-ONLY input is derived from this canonical HIDDEN input rather than redacted independently. Its title and snippet are copied byte-for-byte from HIDDEN, and the original full `url:` line is restored for each item on the final result screen. Earlier result screens, when a natural trajectory contains more than one search, remain in the HIDDEN form. The query header and `[ Back to Search ]` control are never modified.

Verification. We store hashes of the source trajectory, transformed prompt, final screen, and whole item blocks, together with the detected entity spans, to verify that the three conditions differ only in the intended source cues.

Evaluation of transformed inputs. For each final slate of K items, we apply the same K right-cyclic rotations to the transformed item blocks, moving each title, snippet, and condition-specific URL field together. Because hiding or restoring source cues changes the displayed input, all K rotations in HIDDEN and URL-ONLY, including the original order, are newly evaluated. In contrast, ORIGINAL reuses the natural trajectory's original-order selection. All conditions use the same one-sample, first-action replay and final-slate parsing rules.

Prompt. Figure G.1 is the exact system message used by the natural trajectories and all three visibility conditions. The conversation template that follows shows the common one-search case; braced fields are populated from the frozen trajectory, and the final result screen is redacted and rotated according to the condition described above.

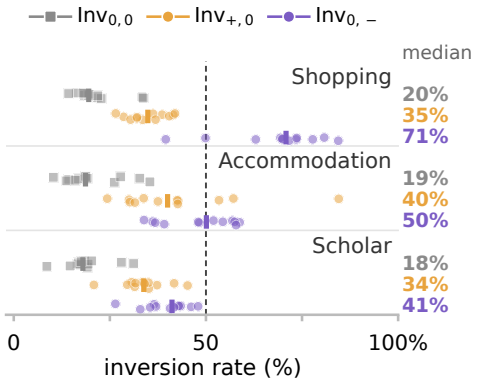


Figure F.1: Inversion rates for pairs in which exactly one item comes from a neutral source. Dot: one model; bar: median. Values in Table F.6.

Table F.1: Shopping: τ_s (percentage points) of the 20 sources that appear most often in the search results (the sources of Fig. 2), by model. Share: fraction of all search results, pooled over models. Cells are colored when the model labels the source SOURCE+ (red) or SOURCE− (blue); gray: SOURCE0. Maj.: the label held by a majority of the models that rate the source (–: none). Models: GPT = GPT-5.4-nano, Luna = GPT-5.6-Luna, Gem = Gemini-3.7-Flash, GLM = GLM-5.3-Flash, DS = Deepseek-v4-flash, Mav = Llama-4-Maverick, Sct = Llama-4-Scout, L3.1 = Llama-3.1-8B-It, Tulu = Tulu-3-8B, Q3.5 = Qwen3.5-27B, Q3 = Qwen3-30B-A3B, Q2.5 = Qwen2.5-32B.

Rank	Source	Share	GPT	Luna	Gem	GLM	DS	Mav	Sct	L3.1	Tulu	Q3.5	Q3	Q2.5	Maj.
1	Amazon	12.2%	−6	+16	+6	+15	+14	+27	+19	+6	+5	+26	+16	+15	+
2	Walmart	6.5%	+11	+24	+11	+17	+15	+25	+21	+12	+11	+20	+21	+17	+
3	eBay	5.1%	−18	+23	+12	+16	+9	−1	+3	+2	+0	+1	+3	+8	–
4	Target	2.7%	+5	+8	−0	+8	+5	+7	+3	+6	+12	+10	+8	+4	0
5	YouTube	2.6%	−29	−12	−22	−35	−35	−27	−14	−21	−22	−30	−22	−16	–
6	Facebook	1.9%	−31	−13	−21	−37	−35	−24	−19	−23	−19	−28	−23	−19	–
7	Pinterest	1.8%	−28	−12	−19	−38	−45	−28	−20	−24	−13	−24	−27	−21	–
8	Reddit	1.8%	−30	−13	−23	−37	−20	−24	−20	−22	−18	−33	−23	−22	–
9	Etsy	1.7%	−14	+8	−8	+1	−0	−9	+0	−9	+3	−5	−2	+2	0
10	Home Depot	1.6%	+10	+4	+4	+9	+9	+13	+4	+9	+8	+6	+6	+1	0
11	Business Insider	1.1%	−45	−12	–	–	–	–	–	–	–	–	−4	–	0
12	Best Buy	1.0%	+1	+16	+3	+8	+5	+7	+3	+5	+4	+11	+5	−0	0
13	Yahoo	0.9%	−28	−10	–	–	–	–	–	–	−4	–	–	–	–
14	Wayfair	0.9%	−3	+9	+7	+5	+1	+17	+15	+16	+7	+18	+7	−0	0
15	Bed Bath & Beyond	0.8%	−4	+14	+8	+2	+3	−6	+0	+4	+6	+8	+2	−4	0
16	Instagram	0.6%	−24	−18	−16	−37	−35	−25	−18	−31	−15	−25	−22	−17	–
17	Lowe's	0.6%	+7	+22	+4	+23	+12	−3	+8	+13	+10	−4	+8	−5	0
18	Macy's	0.6%	+1	+4	−0	+1	+3	−6	+6	+0	+7	+3	−6	−0	0
19	Kohl's	0.5%	+12	+22	+21	+26	+13	−4	+9	+8	+2	+4	+19	+14	0
20	New York Times	0.5%	−6	−2	−17	−7	+3	−7	+2	−13	−11	−13	−6	−6	0

Table F.2: Accommodation: as Table F.1.

Rank	Source	Share	GPT	Luna	Gem	GLM	DS	Mav	Sct	L3.1	Tulu	Q3.5	Q3	Q2.5	Maj.
1	Tripadvisor	8.9%	−14	−12	−4	+5	−3	−11	−13	−8	−8	+2	−6	−10	–
2	Facebook	6.5%	−52	−42	−43	−51	−49	−43	−31	−19	−28	−46	−35	−32	–
3	Expedia	4.3%	−5	−9	−6	−7	−8	−21	−18	−3	−9	−18	−18	−8	–
4	Booking.com	3.5%	+11	+15	+6	+15	+13	+10	+7	+7	+9	+21	+8	+8	+
5	Reddit	3.0%	−51	−47	−39	−27	−25	−44	−35	−24	−31	−41	−35	−35	–
6	Hotels.com	2.2%	−7	−4	−6	+0	−3	−20	−17	−7	−7	−10	−20	−13	–
7	YouTube	2.0%	−46	−42	−34	−39	−41	−42	−21	−8	−24	−45	−28	−21	–
8	Instagram	1.8%	−50	−24	−34	−52	−44	−31	−27	−19	−26	−40	−19	−26	–
9	Marriott	1.3%	−6	+4	−1	+0	−1	−17	−2	+0	+0	−11	−4	−5	0
10	Condé Nast Traveler	1.2%	+6	+5	−8	−3	+1	−9	−12	−4	−2	−0	−2	−2	0
11	Trip.com	1.2%	−28	−4	−2	+1	−2	−6	−12	+3	−5	−3	−16	−10	0
12	Agoda	1.2%	−19	−13	−9	−11	−21	−18	−22	−2	−11	−28	−27	−7	–
13	Travelmyth (.com)	1.0%	−10	−13	+1	+5	+5	−7	−5	−9	−4	−19	−9	−9	0
14	Travelmyth (.co.uk)	0.9%	+14	−19	−27	−1	−13	−31	−12	−12	−7	−1	−22	−9	0
15	KAYAK	0.9%	+10	+17	+9	+6	+17	−6	0	−2	−3	+6	+8	+8	0
16	Travelocity	0.8%	−19	−16	−10	−15	−9	−34	−22	−7	−10	−36	−31	−12	–
17	trivago	0.8%	−14	+2	−5	−3	−2	−13	−15	0	−1	+4	−4	−10	0
18	Rick Steves	0.7%	−44	−42	−30	−26	−24	−36	−32	−18	−22	−19	−27	−32	–
19	Hilton	0.5%	+31	+6	−1	+9	−6	−5	+14	+2	+12	+22	−7	+6	0
20	IHG	0.5%	−21	+5	−8	−5	−23	−11	−11	+1	+12	−1	−15	−5	0

Table F.3: Scholar: as Table F.1.

Rank	Source	Share	GPT	Luna	Gem	GLM	DS	Mav	Sct	L3.1	Tulu	Q3.5	Q3	Q2.5	Maj.
1	arXiv	11.5%	+34	+28	+29	+33	+27	+25	+28	+16	+19	+32	+23	+23	+
2	ScienceDirect	4.2%	+11	+18	+15	+17	+13	+2	−1	+3	+1	+14	+7	+8	+
3	OpenReview	3.9%	+29	+21	+21	+21	+14	+1	+3	+5	+3	+18	+3	+8	+
4	Medium	3.9%	−24	−23	−26	−21	−21	−11	−11	−5	−5	−24	−13	−7	–
5	GitHub	3.6%	−3	−7	−10	+14	+19	+0	+1	−6	−7	+12	+1	+4	–
6	GitHub Pages	2.9%	+17	+8	+8	+13	+11	+12	+13	+8	+8	+16	+11	+11	+
7	NIH	2.8%	+20	+21	+16	+16	+16	+11	+6	+5	+2	+17	+10	+8	+
8	ResearchGate	2.8%	−11	−5	+19	+2	−4	−0	−3	+2	+1	+1	−2	+4	0
9	ACM	2.6%	+20	+23	+24	+23	+14	0	+1	+8	+5	+16	+3	+7	+
10	NeurIPS	2.2%	+34	+34	+20	+27	+20	+20	+23	+16	+13	+29	+23	+18	+
11	ACL Anthology	2.2%	+47	+42	+38	+36	+30	+24	+25	+18	+16	+38	+24	+22	+
12	Emergent Mind	2.2%	−17	−22	−15	+1	−3	+13	+6	+4	+4	−2	+10	+9	–
13	Semantic Scholar	2.0%	+7	−0	+27	+14	+10	+4	+3	+2	+2	+13	+6	+13	+
14	Reddit	2.0%	−21	−21	−33	−30	−34	−29	−25	−22	−28	−30	−38	−22	–
15	YouTube	1.6%	−18	−18	−33	−18	−31	−26	−19	−10	−17	−34	−32	−14	–
16	alphaXiv	1.5%	+6	−4	+23	+12	+8	+11	+13	+17	+14	+10	+22	+18	+
17	Springer	1.4%	+18	+18	+22	+25	+18	+18	+10	+7	+6	+19	+15	+12	+
18	MDPI	1.4%	+19	+22	+24	+24	+21	+18	+9	+12	+7	+23	+16	+12	+
19	IEEE	1.3%	+27	+25	+28	+26	+16	+9	+9	+13	+5	+24	+10	+12	+
20	PMLR	1.3%	+42	+46	+34	+35	+27	+28	+28	+16	+20	+38	+20	+18	+

Table F.4: Sources on which models disagree in direction: at least one model labels the source SOURCE+ and at least one labels it SOURCE− (labels after FDR correction, §3). Values in parentheses are τ_s in percentage points; lists of seven or more models give only the count. SOURCE0: number of models that rate the source but label it SOURCE0. Rank: rank by frequency in the search results of the domain, pooled over models. Model abbreviations as in Table F.1.

Domain	Source	Rank	SOURCE+ (models, τ_s)	SOURCE− (models, τ_s)	SOURCE0
Shopping	Amazon	1	11 models	GPT (−6)	0
	eBay	3	Luna (+23), Gem (+12), GLM (+16), DS (+9), Q2.5 (+8)	GPT (−18)	6
Accommodation	Forbes Travel Guide	34	Q3 (+22)	Mav (−20)	10
Scholar	GitHub	5	GLM (+14), DS (+19), Q3.5 (+12), Q2.5 (+4)	Luna (−7), Gem (−10), L3.1 (−6), Tulu (−7)	4
	ResearchGate	8	Gem (+19), Q2.5 (+4)	GPT (−11), Luna (−5)	8
	Emergent Mind	12	Mav (+13), Sct (+6), L3.1 (+4), Tulu (+4), Q3 (+10), Q2.5 (+9)	GPT (−17), Luna (−22), Gem (−15)	3
	Liner	30	Sct (+14), L3.1 (+7), Q3 (+19), Q2.5 (+15)	GPT (−13), Luna (−14)	6
	Moonlight	44	Q2.5 (+11)	GPT (−17), Luna (−19), Mav (−16)	8
	mbrenndoerfer.com	61	Q3 (+20)	Luna (−19), GLM (−15)	9
	aman.ai	123	Tulu (+21)	GPT (−31)	10
	UC San Diego	132	Q3.5 (+34)	GPT (−11)	10
	University of Michigan	140	Q2.5 (+29)	Q3 (−22)	10
	TOPBOTS	155	DS (+27)	GPT (−22), Luna (−21)	9
	eugeneyan.com	163	Sct (+21)	GPT (−37), Luna (−30)	9
	Brown University	182	Gem (+23)	Luna (−30)	6
	Learn Prompting	199	Sct (+29)	GPT (−39)	10

Table F.5: Inversion rates (%) of Fig. 3. In parentheses: request-level bootstrap 95% confidence interval; number of comparisons with exactly one item selected. –: no pairs of this kind; Gemini-3.7-Flash labels only two Accommodation sources SOURCE+, which yields no $(+,-)$ pairs there. Across the three rates, the number of such comparisons per cell has a median of about 2,000 and is below 100 only for two Gemini-3.7-Flash cells.

	Shopping			Accommodation			Scholar		
Model	$\text{Inv}_{-,+}$	$\text{Inv}_{0,0}$	$\text{Inv}_{+,-}$	$\text{Inv}_{-,+}$	$\text{Inv}_{0,0}$	$\text{Inv}_{+,-}$	$\text{Inv}_{-,+}$	$\text{Inv}_{0,0}$	$\text{Inv}_{+,-}$
GPT-5.4-nano	3 (2–5; 2,021)	20 (15–25; 1,646)	54 (46–62; 1,148)	7 (3–12; 1,226)	26 (21–31; 3,142)	72 (56–84; 380)	1 (1–1; 20,538)	19 (14–25; 1,770)	68 (64–73; 6,489)
GPT-5.6-Luna	0 (0–1; 7,313)	18 (7–33; 369)	92 (85–97; 644)	3 (1–6; 899)	14 (10–18; 5,310)	69 (52–83; 222)	2 (2–3; 24,847)	20 (16–25; 3,995)	69 (65–73; 10,597)
Gemini-3.7-Flash	0 (0–0; 926)	19 (7–34; 447)	97 (85–100; 39)	0 (0–0; 44)	10 (8–13; 8,288)	–	2 (1–3; 16,286)	9 (5–13; 1,428)	48 (42–53; 5,063)
GLM-5.3-Flash	1 (0–1; 3,147)	22 (16–29; 1,534)	71 (55–88; 334)	1 (0–2; 1,052)	16 (13–19; 7,783)	85 (70–96; 201)	1 (1–2; 12,336)	15 (11–18; 3,784)	63 (56–69; 2,737)
Deepseek-v4-flash	1 (0–1; 1,818)	18 (14–23; 1,378)	77 (63–89; 300)	2 (1–5; 790)	19 (15–23; 5,823)	50 (30–75; 133)	1 (1–2; 9,527)	17 (14–21; 3,545)	56 (49–64; 2,163)
Llama-4-Maverick	1 (0–2; 2,202)	23 (19–27; 2,523)	91 (84–95; 338)	5 (2–8; 1,712)	19 (15–24; 1,963)	56 (42–69; 450)	1 (1–2; 12,842)	17 (15–18; 19,191)	57 (50–64; 2,438)
Llama-4-Scout	0 (0–1; 1,413)	18 (14–22; 2,145)	80 (63–94; 235)	3 (0–9; 419)	28 (22–34; 1,993)	93 (82–98; 192)	2 (1–2; 9,594)	18 (16–19; 15,409)	64 (57–70; 1,497)
Llama-3.1-8B-It	10 (8–12; 2,114)	34 (29–38; 2,286)	66 (57–75; 498)	13 (10–17; 870)	33 (29–37; 3,289)	65 (51–77; 220)	8 (7–9; 10,230)	28 (25–31; 3,753)	65 (61–69; 3,220)
Tulu-3-8B	8 (6–10; 1,147)	33 (30–37; 1,948)	85 (79–91; 370)	8 (5–11; 1,001)	35 (31–40; 2,160)	73 (64–81; 402)	9 (8–10; 9,026)	31 (29–33; 8,325)	66 (62–70; 3,246)
Qwen3.5-27B	0 (0–0; 1,512)	14 (10–19; 2,137)	91 (81–98; 149)	1 (0–2; 1,199)	17 (13–21; 4,741)	65 (46–81; 319)	1 (1–2; 10,201)	18 (15–23; 3,781)	70 (64–76; 2,734)
Qwen3-30B-A3B	3 (1–5; 1,658)	16 (13–19; 3,732)	77 (63–88; 315)	4 (2–6; 1,879)	14 (11–18; 2,317)	59 (47–70; 608)	2 (1–3; 11,985)	17 (15–19; 12,371)	59 (52–65; 2,594)
Qwen2.5-32B	2 (1–4; 1,724)	22 (17–28; 2,322)	88 (79–94; 285)	8 (5–11; 1,283)	19 (14–23; 2,448)	62 (45–77; 352)	4 (3–5; 12,247)	19 (15–23; 2,058)	65 (60–69; 3,374)

Table F.6: Inversion rates (%) of Eq. 2 for pairs in which exactly one item comes from a neutral source. In parentheses: request-level bootstrap 95% confidence interval.

	Shopping				Accommodation				Scholar			
Model	$\mathrm{Inv}_{+,0}$	$\mathrm{Inv}_{0,-}$	$\mathrm{Inv}_{-,0}$	$\mathrm{Inv}_{0,+}$	$\mathrm{Inv}_{+,0}$	$\mathrm{Inv}_{0,-}$	$\mathrm{Inv}_{-,0}$	$\mathrm{Inv}_{0,+}$	$\mathrm{Inv}_{+,0}$	$\mathrm{Inv}_{0,-}$	$\mathrm{Inv}_{-,0}$	$\mathrm{Inv}_{0,+}$
GPT-5.4-nano	36 (28–45)	40 (33–46)	7 (5–10)	15 (11–20)	43 (28–59)	57 (51–64)	7 (5–9)	10 (5–17)	34 (29–38)	43 (35–52)	5 (3–8)	6 (5–7)
GPT-5.6-Luna	30 (19–42)	78 (56–91)	1 (0–2)	4 (1–7)	43 (30–55)	34 (27–42)	4 (3–6)	9 (4–16)	35 (31–39)	41 (34–47)	7 (5–10)	8 (6–9)
Gemini-3.7-Flash	27 (12–42)	81 (42–100)	1 (0–4)	8 (4–14)	57 (0–100)	59 (46–72)	1 (0–2)	2 (0–12)	21 (17–25)	26 (19–34)	5 (3–7)	4 (2–5)
GLM-5.3-Flash	32 (25–39)	84 (73–93)	1 (0–1)	9 (6–12)	38 (25–51)	58 (49–67)	3 (2–5)	5 (3–8)	34 (30–38)	35 (27–45)	4 (2–6)	6 (5–7)
Deepseek-v4-flash	32 (25–40)	74 (57–87)	3 (1–6)	12 (9–16)	34 (22–47)	48 (40–57)	5 (3–7)	6 (3–9)	31 (28–36)	33 (25–42)	4 (2–6)	7 (6–9)
Llama-4-Maverick	40 (34–47)	69 (55–82)	2 (1–4)	9 (7–12)	32 (21–44)	37 (29–44)	9 (6–12)	13 (7–19)	32 (29–35)	37 (32–43)	3 (3–4)	7 (6–8)
Llama-4-Scout	42 (35–49)	70 (52–88)	2 (1–4)	11 (8–13)	85 (64–99)	48 (41–55)	13 (10–16)	8 (2–18)	35 (32–38)	42 (35–48)	4 (3–5)	8 (7–9)
Llama-3.1-8B-It	37 (32–42)	63 (54–71)	13 (10–16)	27 (23–30)	43 (32–54)	54 (49–60)	18 (16–21)	28 (23–35)	42 (39–45)	48 (43–53)	13 (12–16)	17 (15–19)
Tulu-3-8B	41 (37–46)	70 (62–77)	10 (7–14)	23 (20–26)	53 (37–68)	57 (52–62)	14 (11–16)	22 (16–29)	45 (43–48)	46 (42–50)	14 (12–15)	19 (18–21)
Qwen3.5-27B	29 (21–36)	72 (52–88)	1 (0–2)	7 (5–10)	30 (16–46)	52 (43–61)	5 (3–7)	7 (4–12)	30 (26–33)	43 (34–53)	4 (2–7)	7 (5–8)
Qwen3-30B-A3B	34 (28–39)	50 (37–64)	5 (3–8)	11 (9–14)	30 (20–42)	36 (29–44)	9 (7–11)	20 (11–29)	31 (28–34)	43 (37–49)	5 (4–6)	8 (7–10)
Qwen2.5-32B	39 (33–45)	73 (62–83)	3 (1–6)	16 (13–19)	24 (14–36)	39 (33–46)	9 (7–11)	13 (7–20)	37 (34–41)	37 (29–45)	11 (8–14)	11 (10–13)

Source-Visibility Replay Prompt

System message

```
You are an autonomous web-search assistant. You are given an
↪  instruction describing what a user wants. Your job is to search
↪  the web and curate the set of results worth presenting to the
↪  user.
# How it works
Each turn you are shown the current page as plain text. Clickable
↪  elements appear inside square brackets, e.g. [ Search ], [ 3 ],
↪  [ Back to Search ].
# Actions
- search[<query>]: available only on the search page (when [ Search
↪  ] is shown). Write a concise, natural keyword query derived from
↪  the instruction (entity type plus key attributes). Do NOT put
↪  the whole instruction or the price constraint into the query.
- select[<n1,n2,...>]: available only on the results page. Provide
↪  the numbers (1-10) of the entities you would present to the user
↪  | every result that satisfies the instruction. Pick as many or
↪  as few as you judge appropriate (e.g. select[2] or
↪  select[1,4,7]). Submitting ends the task. To search again
↪  instead, write select[Back to Search].
# Flow
1. On the search page, issue one search[...] for the desired entity.
2. From the results, read the titles, urls and snippets, then decide
↪  which ones are worth showing the user.
3. Submit them with select[<n1,n2,...>], containing the numbers of
↪  the entities to present to the user. There is no purchase step |
↪  submitting finishes the task.
# Output format (STRICT)
First think step by step inside <think> </think> tags, then present
↪  the admissible action inside <action> </action> tags. Your
↪  return format should be:
<think>(your reasoning for this step)</think>
<action>(the action you chose in this step)</action>
Don't generate any other words after </action>.
```

Frozen conversation template

```
User:
Observation:
Instruction:
{user_instruction}
[ Search ]
Assistant:
<think>{original_search_reasoning}</think>
<action>search[{original_query}]</action>
User:
Observation:
Search results for: {original_query}
[ 1 ] {condition_specific_title_1}
      {condition_specific_snippet_1}
      {condition_specific_url_field_1}
...
[ K ] {condition_specific_title_K}
      {condition_specific_snippet_K}
      {condition_specific_url_field_K}
[ Back to Search ]
```

Figure G.1: Source-visibility replay prompt.

H Detailed Results for the Source-Identity Swap

Controlled inputs and counterbalancing. We construct this experiment separately for each model–domain cell using the full-data, position-unit Bradley–Terry source labels from the natural trajectories. Thus, the labels are not estimated out of fold. An eligible pair contains two items from the same request and different sources, one whose source is labeled SOURCE+ and one whose source is labeled SOURCE−. The items must have identical source-blind requirement-judgment vectors, not merely the same scalar satisfaction score. When a request has multiple eligible pairs, we retain one: the pair with the smallest sum of its two original retrieval ranks, breaking ties by the smaller worst rank, the smaller best rank, and finally a deterministic pair identifier.

For every retained pair, we create four presentations. The authentic source assignment is shown in both item orders, and the swapped assignment is also shown in both orders. Consequently, each fixed item appears once with each source identity in each position. Before constructing these presentations, source-entity spans in the item title and snippet, including the original hostname, are replaced with `the website`. The assigned identity is then displayed only in the item's `url:` field. Candidate content is otherwise unchanged across the four presentations.

Inference and response parsing. Each presentation is evaluated by one independent chat-completion request containing exactly two messages: the system message and user observation shown below. No search is executed during this replay. We provide no tools, function definitions, tool-choice argument, structured-output constraint, or JSON response schema. The prompt asks the model to reason in a `<think>` block and then select exactly one item in a textual `<action>` block.

We retain each model's natural-trajectory decoding configuration. GPT-5.4-nano and GPT-5.6-Luna are queried through the OpenAI Chat Completions endpoint; the DeepSeek, Gemini, Llama-4, and GLM models are queried through OpenRouter; and Llama-3.1-8B-It, Tulu-3-8B, Qwen2.5-32B, Qwen3-30B-A3B, and Qwen3.5-27B are served locally with vLLM. Temperature is zero for all models except GPT-5.6-Luna, for which the natural configuration uses temperature one. Local vLLM runs use seed 42 and the native chat template with thinking enabled; their repetition penalty is 1.05 for Qwen2.5-32B and 1.0 otherwise. We do not set `max_tokens`, `top_p`, `presence_penalty`, or a stop sequence. A choice is analysis-valid when the first `<action>` tag, after removing surrounding whitespace and ignoring case, contains exactly `select[1]` or `select[2]`. We separately record strict prompt compliance, which additionally requires a nonempty `<think>` block, exactly one action in the requested layout, and no trailing text.

Prompt. The following is the exact prompt used for inference. Braced fields are replaced by the request, frozen search query, redacted item text, and the authentic or swapped source assignment for that arm. There is no preceding search conversation and no `Back to Search` option.

Source-Identity Swap Inference Prompt

System message

```
You are an autonomous web-search assistant. You are given an
↪  instruction describing what a user wants. The search has already
↪  been executed, and you are shown two search results. Your job is
↪  to select exactly one result worth presenting to the user.

From the results, read the titles, urls and snippets, then decide
↪  which result best satisfies the instruction.

# Actions
- select[1]: select the first result.
- select[2]: select the second result.

Select exactly one result. Do not select both.

# Output format (STRICT)
First think step by step inside <think> </think> tags, then present
↪  the admissible action inside <action> </action> tags. Your
↪  return format should be:
<think>(your reasoning for selecting the result)</think>
<action>(select[1] or select[2])</action>

Don't generate any other words after </action>.
```

User message

```
Observation:
Instruction:
{user_instruction}

Search results for: {search_query}

[ 1 ] {redacted_title_1}
      {redacted_snippet_1}
      url: https://{assigned_source_1}/

[ 2 ] {redacted_title_2}
      {redacted_snippet_2}
      url: https://{assigned_source_2}/
```

Table H.1 reports the results of §5.2 with the number of complete, valid four-arm pairs in each model–domain cell. Each pair contributes four choices, crossing source assignment with item order. DeepSeek-v4-Flash in the main-text table denotes DeepSeek-v4-flash-0731.

Table H.1: **Selection by displayed source identity with content fixed.** T_+ is the selection rate for the item displayed with a SOURCE+ source identity; parentheses give the number of complete, valid four-arm pairs. Two-sided tests of $H_0 : T_+ = 50\%$ use query-cluster standard errors and a Student t reference distribution. $*$, †, and ‡ denote unadjusted $p < 0.05$, $p < 0.01$, and $p < 0.001$, respectively.

Model	WebShop	Accommodation	Scholar
GPT-5.4-nano	$59.59\%^{\ddagger}$ (446)	$62.03\%^{\ddagger}$ (158)	$59.32\%^{\ddagger}$ (1,846)
GPT-5.6-Luna	$63.48\%^{\ddagger}$ (397)	$60.06\%^{\ddagger}$ (154)	$57.04\%^{\ddagger}$ (1,979)
GLM-5.3-Flash	$62.90\%^{\ddagger}$ (248)	$56.37\%^{\ddagger}$ (106)	$55.20\%^{\ddagger}$ (990)
DeepSeek-v4-Flash	$63.23\%^{\ddagger}$ (189)	$55.91\%^{\ddagger}$ (110)	$54.83\%^{\ddagger}$ (890)
Gemini-3.7-Flash	$69.25\%^{\ddagger}$ (161)	56.67% (15)	$56.76\%^{\ddagger}$ (1,460)
Llama-4-Maverick	$82.35\%^{\ddagger}$ (289)	$57.46\%^{\ddagger}$ (248)	$60.70\%^{\ddagger}$ (1,138)
Llama-4-Scout	$74.41\%^{\ddagger}$ (256)	51.23% (61)	$58.06\%^{\ddagger}$ (1,058)
Llama-3.1-8B-It	$71.86\%^{\ddagger}$ (311)	52.04% (159)	$55.92\%^{\ddagger}$ (1,449)
Tulu-3-8B	$79.35\%^{\ddagger}$ (224)	$55.62\%^{\ddagger}$ (178)	$55.26\%^{\ddagger}$ (1,382)
Qwen3.5-27B	$71.85\%^{\ddagger}$ (151)	$54.28\%^{\ddagger}$ (146)	$57.50\%^{\ddagger}$ (877)
Qwen3-30B-A3B	$70.71\%^{\ddagger}$ (210)	$53.12\%^{\dagger}$ (232)	$57.72\%^{\ddagger}$ (1,078)
Qwen2.5-32B	$66.23\%^{\ddagger}$ (285)	$52.95\%^{\ddagger}$ (212)	$54.41\%^{\ddagger}$ (1,280)

I SOURCE PREFERENCE AND MEAN REQUEST SATISFACTION

We examine whether sources whose items satisfy more user requirements on average also receive higher source-preference scores. This analysis motivates the training hypothesis in §6.1: models may have encountered a similar relationship during training and learned to favor particular sources.

Setup. For each model and domain, we pair each source's preference score τ_s under the source-visible condition with the mean request satisfaction of its items. Each item's satisfaction score is the fraction of applicable request requirements judged to be satisfied, using the source-blind judgments described in Appendix D. We average these scores over all valid request–URL entries for that source in the model's search results, including both selected and unselected items. This average is not restricted to the matched pairs used to estimate τ_s.

Within each model and domain, we rank sources with an available τ_s by their number of valid judged request–URL entries. Our main analysis retains the 20 most frequent sources; we also report results for 10 and 30 sources to examine how the relationship changes as more sources are included. Selection is based on frequency, not on the value or sign of τ_s or mean satisfaction. The selected sources can differ across models. For each domain, we compute Pearson's r and Spearman's ρ over the pooled model–source observations from 11 models. The three cutoffs give 110, 220, and 330 observations per domain, respectively. Each observation receives equal weight; we do not average source scores across models before computing the correlations.

Results. Fig. I.1 shows a positive relationship in all three domains at each cutoff. With 20 sources per model, Pearson's r is .778 in Shopping, .598 in Accommodation, and .677 in Scholar; the corresponding Spearman correlations are .720, .561, and .535. The relationship is also present within individual models: at this cutoff, both correlations are positive in all 33 model–domain combinations. Thus, the positive pooled correlations do not arise solely from differences between models.

The strength of the relationship depends on which sources are included. As the cutoff increases from 10 to 30, Pearson's r decreases from .823 to .577 in Shopping and from .705 to .465 in Accommodation. Spearman's ρ also decreases, from .779 to .554 and from .677 to .473, respectively. In Scholar, both correlations increase: Pearson's r rises from .494 to .705, and Spearman's ρ from .467 to .626.

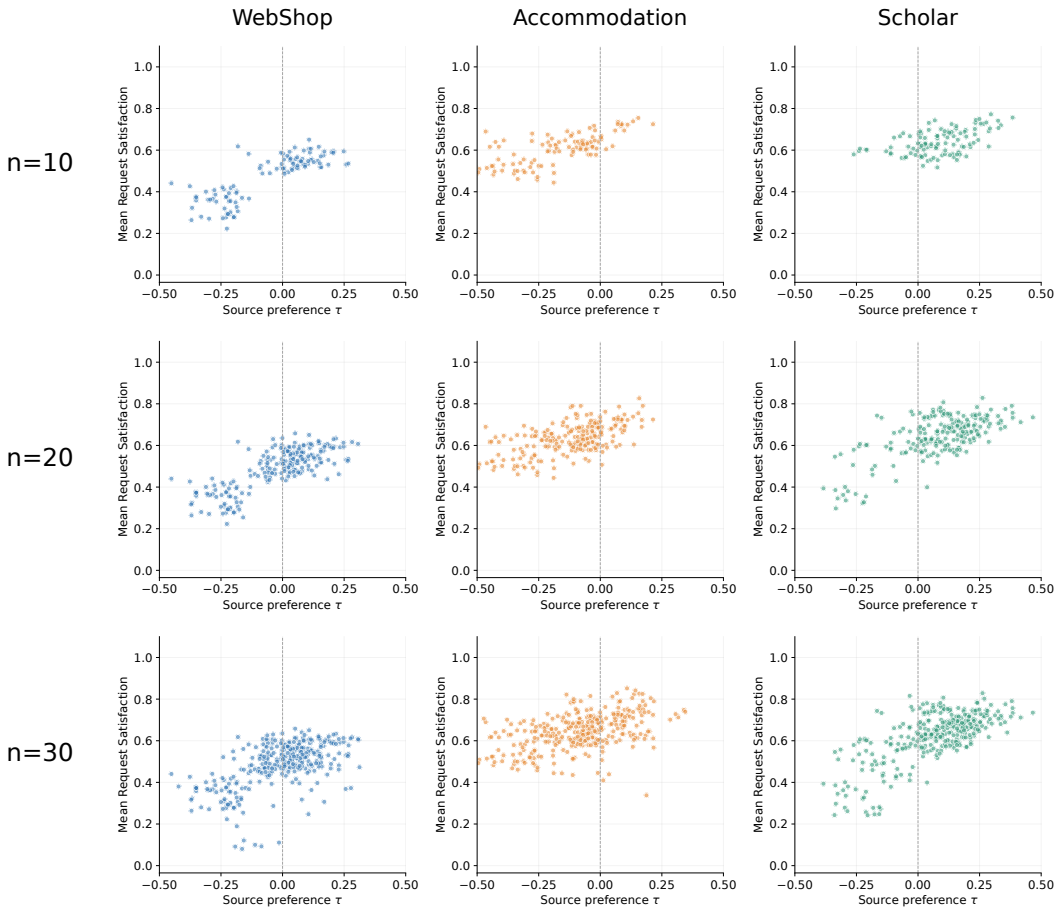


Figure I.1: Source-preference score τ_s and mean request satisfaction among the 10, 20, and 30 most frequent sources within each model and domain. Each point represents one model–source observation. The three cutoffs yield 110, 220, and 330 observations per domain across 11 models, respectively. Mean satisfaction includes both selected and unselected items. Pearson's r and Spearman's ρ are computed over the plotted observations without averaging across models.

The main-text observation therefore describes frequently encountered sources; it does not imply an equally strong relationship across all sources.

Interpretation. The two quantities describe different aspects of a source: τ_s summarizes preference among items matched on request satisfaction, whereas mean satisfaction describes all of its judged items in the observed search results. Their correlation shows that sources providing more satisfying items on average also tend to be preferred in matched comparisons. Models may have encountered a similar relationship during training, but this analysis does not establish what their training data contained or why their existing preferences arose. The controlled experiment in §6.1 tests whether introducing a source–satisfaction relationship during preference training can create a preference that persists on equal-satisfaction pairs.

We report these correlations descriptively. Sources and requests can recur across observations, and the correlations do not account for uncertainty in the estimated τ_s or mean satisfaction.

J IMPLEMENTATION DETAILS FOR PREFERENCE TRAINING

We use direct preference optimization (DPO; Rafailov et al., 2024) to test whether associating a source label with better user-request satisfaction during training changes source preference at evaluation time. The two synthetic source names are `Kelto` and `Varnis`. All conditions use the same requests, item texts, and preferred selection actions; they differ only in the assignment of these names to the items.

Preference pairs and data splits. We construct pairs from frozen WebShop search-result slates collected by Tulu-3-8B-SFT. Each item has a binary satisfaction vector recording whether it meets the requested product type, attributes, options, and price constraint, when applicable. Training and validation pairs contain two items for the same request that originate from different hosts. The preferred item satisfies exactly one additional rubric constraint, with all other vector components equal. Preference labels are therefore determined by the satisfaction vectors independently of the synthetic source assignment.

We partition by user request using the frozen five-fold assignment: folds 0–2 provide training data, fold 3 provides validation data, and fold 4 provides the evaluation set. The selected sets contain 5,000 training pairs from 835 requests, 500 validation pairs from 252 requests, and 1,000 evaluation pairs from 292 requests. Requests do not overlap across splits. Pair selection proceeds across requests in round-robin order, with at most eight training pairs, two validation pairs, and four evaluation pairs per request; item reuse is limited to three pairs within each split's selection procedure.

Source assignment and position balancing. We construct three conditions, *Balanced*, *Aligned*, and *Reversed*, in which Kelto is assigned to the preferred item in 50%, 80%, and 20% of training pairs, respectively. The other item receives Varnis. A seeded ordering within blocks of ten pairs enforces these proportions exactly. Every pair is presented in both item orders while preserving its source assignment, yielding 10,000 conversational DPO examples per condition. Each source thus appears once in every prompt and equally often in each position; the manipulation changes its association with the preferred item. This is exact position counterbalancing, rather than random placement. Validation uses the same two-order construction with a 50% assignment rate, yielding 1,000 examples shared by all conditions.

Prompt and target completions. Fig. J.1 gives the common prompt template. The original URL field is omitted. URL strings and source-entity spans detected by the redaction procedure are replaced with `[link]` and `[website]`, respectively, before the synthetic `Source:` field is added. The resulting title and content are held fixed across conditions. The prompt contains neither rubric labels nor the condition name or source-assignment probability.

Both DPO completions are selection actions for the same prompt. If the preferred item is displayed at position 1, the chosen completion is `<action>select[1]</action>` and the rejected completion is `<action>select[2]</action>`; the assignments are reversed when the preferred item is at position 2. No explanatory rationale is included in either target completion. We render the conversational messages with each model's native chat template.

Optimization and checkpoint protocol. For the checkpoint-trajectory experiment, we initialize separate LoRA adapters for Llama-3.1-8B-Instruct and the official post-trained Qwen3.5-9B checkpoint, using the same fixed training pairs in all three conditions. The pretrained weights remain frozen. We optimize the standard DPO objective against the corresponding initial policy, with reference log probabilities precomputed before optimization. Table J.1 lists the shared settings. The 5,000-update budget with an effective batch size of eight corresponds to 40,000 presentations of the 10,000 rendered training examples, or four epochs. We save adapters and compute validation metrics every 500 optimizer updates. The trajectory evaluates the initial policy (step 0) and the adapters saved at steps 500, 1,000, 2,000, and 5,000; these steps are fixed in advance rather than selected by validation performance. Full source-choice evaluation is performed after training using the saved adapters.

LoRA targets the attention projections `q_proj`, `k_proj`, `v_proj`, and `o_proj`, and the feed-forward projections `gate_proj`, `up_proj`, and `down_proj`. For Qwen3.5, it additionally targets `in_proj_qkv`, `in_proj_z`, `in_proj_a`, `in_proj_b`, and `out_proj`. We use scaled dot-product

Synthetic-Source Selection Prompt

System message

```
You are a web-search assistant. Compare the two search results
↪  against the user's instruction. Return exactly one action:
↪  <action>select[1]</action> or <action>select[2]</action>.
```

User message

```
Instruction:
{user_request}

[1]
Title: {title_1}
Content: {content_1}
Source: {source_1}

[2]
Title: {title_2}
Content: {content_2}
Source: {source_2}

Choose the result that better satisfies the instruction.
```

Figure J.1: Prompt template used for synthetic-source DPO training, validation, and evaluation. Braced fields are substituted with the request and item information; the two source fields contain Kelto and Varnis in the assigned order. Line wrapping in the system message is for presentation only.

Table J.1: Shared settings for the fixed-data checkpoint-trajectory experiment. The training-set size counts unique content pairs; each pair yields two examples after reversing item order.

Setting	Value
Unique training pairs / rendered examples	5,000 / 10,000
Data-assignment and training seed	17
Optimizer updates	5,000
Microbatch size / accumulation steps	4 / 2
Effective batch size	8
Optimizer	Fused AdamW
Learning rate	5×10^{-6}
Learning-rate schedule	Cosine, 5% warmup (250 updates)
DPO coefficient β	0.1
Maximum training sequence length	4,096 tokens; keep-start truncation
LoRA rank / scaling / dropout	16 / 32 / 0.05
Numerical precision	bfloat16; TF32 enabled
Gradient checkpointing	Enabled, non-reentrant
Validation and adapter saving	Every 500 updates
Source-choice evaluation checkpoints	0, 500, 1,000, 2,000, 5,000

attention; Qwen3.5's linear-attention blocks use the differentiable PyTorch fallback. Each run uses one GPU.

Evaluation and aggregation. Evaluation pairs have identical observed satisfaction vectors. For each pair, we cross the two content orders with the two source assignments, giving four prompts and 4,000 evaluation examples in total. This crossing exposes each item under both source names in both positions. The matching criterion is equality of the measured rubric vector; item texts need not be identical.

For each prompt, we score both legal selection actions by summing the conditional log probabilities of their assistant-completion tokens, including the chat-template completion suffix and excluding prompt tokens. Qwen3.5 scoring follows the conversational completion-boundary convention used by the DPO trainer. No response sampling is used. Kelto is counted as selected if its action has strictly greater log probability than the Varnis action; exact log-probability ties are counted as non-Kelto and retained in the denominator.

Let $\mathcal{P}_q$ be the evaluation pairs for request q, and let $\ell_K(q, p, v)$ and $\ell_V(q, p, v)$ denote the two action log probabilities under rendering $v \in \{1, 2, 3, 4\}$. We report the request-averaged Kelto selection rate,

$$\widehat{P}_K = \frac{1}{|\mathcal{Q}|} \sum_{q \in \mathcal{Q}} \frac{1}{|\mathcal{P}_q|} \sum_{p \in \mathcal{P}_q} \frac{1}{4} \sum_{v=1}^{4} \mathbf{1}\{\ell_K(q, p, v) > \ell_V(q, p, v)\} . \tag{3}$$

This gives every request equal weight despite differences in its number of evaluation pairs. We apply the same scoring and aggregation to the initial policy and every evaluated adapter.

K Agent Reasoning Examples

We provide examples of agent reasoning discussed in §7.

Example 1

Available Choices

```
[ 1 ] Star Wars ☆ The Mandalorian Grogu Baby Yoda 2-Pack ...
      Each shirt has a sleeve detail down one side. These t-shirts
      are crewneck and tagless for comfort. In Multi. 60% Cotton/40%
      Polyester. Machine washable.
      url: https://ebay.com/

[ 2 ] STAR WARS The Mandalorian Boys T-Shirt and Shorts Set ...
      MACHINE WASHABLE: this Mandalorian boys' clothing set can be
      thrown in the wash for a perfect clean every time without
      worrying about the colors and images
      url: https://walmart.com/
```

Agent Reasoning

Result 1 is a 2-pack of Mandalorian/Grogu baby yoda shirts and explicitly says machine washable, but the snippet doesn't mention baby blue color or price under $50. Result 2 explicitly mentions machine washable and is from Walmart (more likely to have under-$50 pricing), but the snippet also doesn't mention baby blue color. Between the two, **Result 2 is more likely to match the "under $50" requirement due to typical Walmart pricing**, so it best satisfies the instruction overall.

Figure K.1: Reasons that the selected retailer will offer a lower price.

Example 2

Available Choices

```
[ 1 ] Unique Loom Sofia Collection Area Rug - 8 Round Gray/Beige -
the website
      Unique Loom Sofia Collection Area Rug - 8 Round Gray/Beige
      ;Condition. New with tags ; Quantity. 1 available ; Item
      number. 366532405544 ; Brand. Unique Loom ; MPN.
      url: https://ebay.com/

[ 2 ] Sofia 7 ft. x 7 ft. Round Dark Gray/Beige Border Area Rug
      Sofia rugs are power loomed in Turkey and beautifully styled
      using high-quality polypropylene fibers and a jute backing to
      bring antique carpet qualities to
      url: https://homedepot.com/
```

Agent Reasoning
Result 2 is a Safavieh "Sofia" round rug listing with size and color details **and is hosted on Home Depot, which is more likely to provide a clear price under $350. Result 1 is an eBay listing with insufficient information** to confirm it's Safavieh and to verify the price.

Figure K.2: Reasons that the selected retailer will have price information available.