What Decides Who AI Answer Engines Name and Cite

An observational study of 68,636 answers from seven engines, with and without web search

Robert Griesmeyer
RankMonster

November 15, 2026 · Version 1.0

ABSTRACT

AI answer engines decide which brands to name in their prose and which web pages to cite as sources. We study how those decisions are made, and how they depend on whether the engine ran a web search, using 68,636 stored answers to 7,711 prompts from seven commercial engines over five months of 2026, and 4,284 answers to fixed sets of category questions in which every brand named was extracted. A language model can take a name only from its parameters or from its context, and a web search works by appending retrieved text to the context: answers that returned sources were billed a median of 4,400 to 11,000 more input tokens than answers that did not. Three natural contrasts in the data support causal or near-causal readings. When one engine's search tool was withdrawn, its unprompted naming of tracked brands fell from 1.90% to 0.70% while four unchanged engines answering the same prompts rose from 1.18% to 3.53% (difference in differences -3.55 percentage points, 95% interval -7.11 to -1.52). Across 556 prompts that ChatGPT answered both with and without a search, it named the brand only when it had searched. And one engine reports the pages it retrieved as well as those it cited: of 45,035 retrieved results it cited 42%, with the first result cited 68% of the time and the ninth 32%. When engines do search, most of who is named is still decided by a prior the engines share: how often a model without search names a brand predicts which brands five searching engines name with a pseudo-R-squared of 0.52, and what was retrieved adds 0.03. Retrieval matters most for brands with no prior, which are named 80 times more often when their own site is among the sources, and still only 7% of the time. Citations are heavy-tailed over 61,222 sites; two engines answering the same prompt share under a tenth of their combined sources; and in 61% of answers that cite a brand's own site for a prompt that does not name the brand, the brand goes unnamed. Given retrieval, citation is predicted by position, by the match between a page's title and the question, and by short self-contained passages, and not by FAQ blocks, summary boxes or question headings. A single reading is unreliable: an unprompted mention recurs on the next reading of the same prompt less than half the time. We document five measurement artefacts, three of which produced strong false findings before correction, and specify two experiments that would close the main gaps.

Keywords: answer engine optimization; generative engine optimization; AI search; retrieval-augmented generation; brand mentions; citations; test-retest reliability

1. Introduction

A growing share of the questions people once typed into a search box are now put to a language model. The model returns a paragraph or a list in place of ten links, and in doing so it makes two decisions that a search results page never had to make. It decides which organisations, products and people to **name** in its prose, and it decides which web pages to **cite** as the sources of what it wrote. For anyone whose livelihood depends on being found, those two decisions have replaced the ranking position as the thing worth understanding. A brand that is named in the answer to "what is the best applicant tracking system for a fifty-person company" has been recommended. A brand that is absent has, for that reader and that question, ceased to exist. A page that is cited may receive a visit; a page that is not cited will not, however well it ranks in a conventional index. The Pew Research Center's browsing-panel study found that when Google showed an AI summary, users clicked a traditional result on 8% of visits against 15% without one, and clicked a link inside the summary on 1% [33].

This paper is an empirical study of how those two decisions are made. It asks what determines who is named and who is cited across seven commercial answer engines, and it separates the question along the line that matters most mechanically: whether or not the engine ran a web search before it answered. That line matters because a language model has exactly two places from which the name of a company can come. One is the model's parameters, fixed when training ended, in which the regularities of a very large body of text are stored. The other is the model's context window, the sequence of tokens it is given at the moment of the request. A web search is a way of writing into the second place. When an engine searches, text from the retrieved pages is inserted into the model's input as tokens, next to the user's question, and the model then composes its answer conditioned on both. Nothing else about the model changes. The search does not consult a different model, and it does not hand back a ranked list that the model must reproduce. It lengthens the prompt. Everything that a retrieved page can do to the answer it does as input tokens, competing for the model's attention with the other retrieved pages and with what the model already believed.

That description is simple, and it is the foundation of everything that follows, so Section 2 sets it out in full: what a token is, what the context window holds, how a search tool call is issued and returned, how many tokens a search adds, and how each of the seven engines attaches a citation to a passage. The rest of the paper then measures the consequences. We use the term **mention** for a brand's name appearing as readable text in an answer and **citation** for a URL returned as a source of an answer. The two are distinct outcomes with distinct causes, and one of the main results of the paper is how weakly they are coupled: across 3,151 answers to questions that did not name the brand in which an account's own site was among the cited sources, the account's brand was named in the answer 39% of the time.

1.1 Why an observational study, and what kind

The existing academic literature on what its authors call generative engine optimisation is largely experimental and largely simulated. The founding paper of the field, Aggarwal et al. [1], built a benchmark of ten thousand queries, fetched the top five Google results for each, asked a language model to write a cited answer from those five documents, and then rewrote one document at a time to see how its share of the answer changed. That design gives clean causal estimates of how a change to a page changes its share of an answer, and it has a boundary the authors state plainly: the set of retrieved documents is held fixed. It measures what happens after a page has been retrieved and placed in the context. It says nothing about which pages are retrieved, whether the engine searches at all, or what the engine says when it does not search. Later benchmark work has the same shape [27], and the studies of manipulation by injected text [24-26] work inside a fixed context by construction.

A second line of work observes production engines from the outside. Zhang et al. [30] collected 1.4 million citation links from six LLM-based engines and two conventional ones over 55,936 queries. Chen et al. [28] compared the kinds of site that AI engines and Google cite. Liu, Zhang and Liang [13] audited whether cited sources support the sentences they are attached to. These studies establish what the engines cite in aggregate, and they treat the engine as a single black box that maps a query to a list of links. They do not observe whether a given answer was grounded in a search, they do not observe pages that were retrieved and passed over, they do not follow the same question over months, and they are about citations only. Which brands are named in the prose is outside their scope.

This paper sits between the two. It is observational, so it inherits the usual limits on causal language, and we are careful throughout to say which estimates carry a causal reading and which do not. But the data have four properties that the published observational studies lack, and each one opens a question that could not be asked before.

1. **Answers are stored whole, with their sources, per engine, per question, over time.** The panel holds 97,820 freshly generated answers to 10,802 prompts from 501 accounts between April and November 2026. A third of the prompt and engine pairs in the analysis period were read more than once, some of them ten times. That makes test-retest reliability measurable, and it supplies within-prompt comparisons in which the question, the brand and the engine are held fixed.
2. **The engines decide for themselves whether to search.** ChatGPT, Claude and Gemini are called with a search tool available and not forced. The same prompt is therefore sometimes answered from parameters alone and sometimes after a search, and the token count of the response records the difference.
3. **One engine had its search tool withdrawn.** For cost reasons the operator stopped sending Grok a search tool on 7 August 2026 while every other engine's configuration was left alone. That is a natural experiment with four contemporaneous controls.

4. **One engine reports what it read as well as what it cited.** Since 30 September 2026 the stored sources of a Claude answer are every search result the tool returned, in order, with the cited ones marked by the span of answer text they support. Each such answer is a choice set: nine or more pages were placed in the context, and the model cited some and not others. This gives the comparison group that citation studies normally lack, without any new data collection.

A fifth body of data complements the panel. On six fixed sets of about forty prompts each, covering four software categories and two service categories, six engines were asked every prompt once a month for three months, and every brand named in every answer was extracted and resolved to a roster. Here the unit of analysis is the pair of an answer and a brand, the outcome is whether that brand was named, and the predictors include a direct measure of the model's prior: how often an engine that never searches names the same brand for the same category.

1.2 Research questions

The paper is organised around five questions.

RQ1. When do answer engines search, and what does a search put into the model? We measure how often each engine returns sources, which features of a prompt predict a search, and how many tokens a search adds to the model's input.

RQ2. Does searching change who is named? We estimate the effect of search on the probability that a tracked brand is named, using the withdrawal of Grok's search tool and within-prompt contrasts, and we measure what search does to the set of brands named for a category.

RQ3. Who is named when the engine does search? We separate the contribution of the model's prior from the contribution of the retrieved text, at the level of individual brands in individual answers.

RQ4. Which sites are cited, how concentrated are citations, how much do engines agree, and how stable is one engine from one reading to the next?

RQ5. Of the pages placed in the model's context, which are cited? We model Claude's citation choices within each answer as a function of retrieval position, the page's title, the kind of site and the structure of the page.

A sixth question is methodological and runs through all the others: **how reliable is a single reading of an answer engine**, and how many readings does a claim about a change in visibility require?

1.3 Summary of findings

The findings are stated here with their main numbers; each is developed, with its uncertainty and its caveats, in Sections 7 to 11.

Searching is a choice the engines make differently. Over 68,636 analysed answers, the share carrying at least one source ranged from 35% for Grok, whose search tool was withdrawn part way through, and 53% for ChatGPT to 97% for Google AI Overview and effectively 100% for Perplexity. ChatGPT's decision to search is strongly predictable from the prompt. Against informational prompts, the odds that it searched were 5.2 times higher for comparison prompts and 4.2 times higher for requests for a recommendation, and the words "best" or "top" or a price term each more than doubled the odds. Claude shows the same pattern.

A search is a large block of input tokens. Answers that returned sources were billed a median of 11,832 tokens on Grok against 866 without, 11,317 against 2,544 on Claude, and 10,057 against 5,695 on ChatGPT. The retrieved text outweighs the user's question, which had a median length of ten words, by about three orders of magnitude.

Withdrawing search cut unbranded naming by more than half. When Grok lost its search tool, the share of its answers to unbranded prompts that named the tracked brand fell from 1.90% to 0.70%, while the same share on the four unchanged engines, answering the same prompts, rose from 1.18% to 3.53%. The difference in differences is −3.55 percentage points (95% interval −7.11 to −1.52, clustered on account), and −3.39 with prompt fixed effects. Across 556 prompts that ChatGPT answered both with and without a search, it named the brand only on readings where it had searched (10 prompts to none). For prompts that already contain the brand's name, search made little difference.

The model's prior explains most of who is named even when it searches. On the category prompt sets, how often the non-searching engine names a brand predicts whether a searching engine names it with a pseudo-R^2 of 0.52. Adding whether the brand's own site was among the sources, and whether the brand appeared in a source's title, raises that to 0.55; those two source variables alone reach 0.23. But the source variables matter most exactly where the prior is empty. A brand the non-searching engine never names was named in 0.09% of sourced answers when its own site was not among the sources and in 7.1% when it was, a difference of roughly eighty to one.

Search widens the field at the margin. Between 6% and 10% of the brand mentions in sourced answers went to brands the non-searching engine never names for that category, against 1.5% to 3% in unsourced answers from the same engines. Sourced answers named more brands, but the difference disappeared once answer length was held fixed.

Citations are heavy-tailed and engines barely agree. 476,973 citations fell on 61,222 sites. Within each engine the distribution of citations over sites is close to a power law with exponent near 2, between half and two thirds of cited sites are cited once, and the ten most-cited sites take between 4% (Gemini) and 25% (Google AI Overview) of citations. Two engines answering the same prompt in the same run shared a mean Jaccard similarity of 0.09 between their cited sites, and in 36% of such pairs they shared none. Of the sites cited for a prompt by any engine, 84% were cited by exactly one.

One engine is not stable either. The same engine reading the same prompt again, a median of ten days later, returned a set of cited sites with a mean Jaccard similarity between 0.19 (Gemini) and 0.37 (Claude, Perplexity) to its earlier set.

Being cited is not being named. For unbranded prompts, an account's own site among the sources raised the odds that the account was named by a factor of about 74 within the same prompt and engine. In absolute terms, the brand was still not named in 61% of the answers that cited its site.

Position in the context dominates what is cited. Of the pages Claude retrieved, it cited 42%. The first result was cited 68% of the time and the ninth 32%. Within an answer, each unit of log position cut the odds of citation in half. A title sharing the prompt's terms, a year in the title and a homepage URL each raised the odds; an encyclopedia page was cited about a third as often as other results at the same position. Among page-structure features, the share of short self-contained paragraphs and the density of statistics were positively associated with citation and page length negatively.

A single reading is a weak measurement. For unbranded prompts, a brand that was named in one reading was named in the next reading of the same prompt by the same engine 46% of the time. In the pool of repeatedly read prompts, where the mention rate is 0.6%, a tracking set of one hundred unbranded prompts read once has a 95% half-width of 1.5 percentage points.

1.4 Contributions

The paper makes four contributions. First, it gives an account, checked against vendor documentation and against token counts in our own data, of the mechanism by which web search affects a language model's answer, written for a reader who needs it to be exact and does not need it to be mathematical. Second, it reports the first estimates we are aware of, from production engines, of the effect of search availability on who is named, of the relative weight of prior and retrieved text in naming, and of which retrieved pages are cited conditional on retrieval. Third, it quantifies the reliability of the measurements on which the whole practice of AI visibility tracking rests, and derives what that implies for sample sizes. Fourth, it documents five measurement artefacts, three of which reversed a conclusion during the analysis, and the corrections for them, because any study of this kind will meet the same ones.

1.5 What this paper is not

It is not a controlled experiment. No page was edited to see what happened, no fictitious brand was planted, and no prompt was issued for the purpose of the study. Two experiments that would close the main gaps, a paired search-on and search-off ablation and a randomised context-injection design, are specified in Section 15 and were not run. It is not a study of a random sample of questions people ask: the prompts were written by or for paying customers of a visibility tracking product who wanted to know about their own markets, and they lean heavily towards commercial software and services. It is not a study of the consumer applications. Every answer was obtained through a vendor's API with a stated configuration, which differs from the consumer product in

system prompt, in the number of searches allowed, and in personalisation. And it is not neutral: the author builds and sells the instrument that collected the data, and the two largest accounts in the panel belong to the author's own company. Section 13 takes each of these limits in turn and says which findings they threaten.

1.6 Organisation

Section 2 explains how an answer engine produces an answer. Section 3 reviews related work. Section 4 states the hypotheses. Section 5 describes the data and the instrument, and Section 6 the statistical methods. Sections 7 to 11 report the five studies. Section 12 draws the results into a single account, Section 13 lists the threats to validity, Section 14 sets out what the results imply in practice, and Section 15 concludes with the experiments that should come next. Appendices give the engine configurations, the variable definitions, the mention-matching rule and the reproduction steps.

2. Background: how an answer engine produces an answer

This section describes the mechanism the rest of the paper measures. It is written to be exact about what happens and in what order, because several common beliefs about AI search are wrong in ways that matter for interpreting the results. The model does not "look things up" while it writes. It does not rank brands. It does not have access to a search index, a link graph or a knowledge base as it composes a sentence. It has parameters, and it has a sequence of tokens in front of it, and a web search is one way some of those tokens get there.

2.1 Tokens

A language model does not read characters or words. Its input is a sequence of integers, each of which indexes an entry in a fixed vocabulary of some tens of thousands to a few hundred thousand **tokens**. A token is a fragment of text: a common word, a piece of a rarer word, a punctuation mark, a space joined to the word that follows it. Vocabularies are built by algorithms in the family of byte-pair encoding [4], which start from single bytes and repeatedly merge the most frequent adjacent pair into a new symbol until the vocabulary reaches its target size. Frequent strings end up as single tokens and rare strings are spelled out of several. In English prose a token averages about three quarters of a word, so a thousand words is roughly 1,300 tokens, with the ratio worse for URLs, code, numbers and unusual names.

Two consequences follow for brand names. A well-known name tends to be one or two tokens, and an invented or unusual one is assembled from pieces that individually mean nothing. And the model's only representation of a brand is whatever its parameters associate with that token sequence in context. There is no entry for the company anywhere.

Tokens are also the unit of cost and of capacity. Providers bill per input token and per output token, and each model has a maximum number of tokens it can be given at once. Both facts will matter: the cost of a search is mostly the cost of the tokens it adds, and that cost is the reason the engines in this study were configured as they were.

2.2 The context window and next-token prediction

The model is a function from a sequence of tokens to a probability distribution over the next token. Write the sequence so far as $x_1, x_2, \dots, x_t$. The model computes

$$p_\theta(x_{t+1} \mid x_1, \dots, x_t), \tag{1}$$

a distribution over the whole vocabulary, where θ denotes the parameters: several billion to several hundred billion numbers fixed at the end of training. One token is drawn from that distribution, appended to the sequence, and the computation is repeated with the longer sequence. An answer of 300 words is the result of about 400 such draws. The probability of a whole answer $y = (y_1, \dots, y_m)$ given an input x factorises as

$$p_\theta(y \mid x) = \prod_{j=1}^{m} p_\theta(y_j \mid x, y_1, \dots, y_{j-1}). \tag{2}$$

The sequence the model is conditioned on, input plus what it has written so far, is the **context**, and the maximum length it may have is the **context window**. Everything the model can take into account when choosing the next token is either in the context or in θ. There is no third source.

Inside the function, the architecture is the transformer [3]. Each token is mapped to a vector, and a stack of layers repeatedly updates each position's vector by **attention**: position i computes a weighted average over the vectors at all earlier positions, with weights

$$\alpha_{ij} = \frac{\exp\left(q_i^\top k_j / \sqrt{d}\right)}{\sum_{l \le i} \exp\left(q_i^\top k_l / \sqrt{d}\right)}, \tag{3}$$

where q_i and k_j are learned projections of the vectors at positions i and j and d is their dimension. The weights at each position sum to one. That normalisation is worth holding on to. Attention is a budget: weight given to one part of the context is weight not given to another. When the context grows from a ten-word question to ten thousand tokens of retrieved pages, each page, each paragraph and each brand name inside it is competing for a share of a fixed total, at every layer and at every step of generation. Position in the context affects that competition. Models use information near the beginning and the end of a long context more reliably than information in the middle, the effect Liu et al. named "lost in the middle" [14], and Section 10 will show a steep position gradient in which retrieved pages are cited.

Because a token is drawn from a distribution and not read off as its maximum, generation is random. The randomness is usually controlled by a **temperature** τ applied to the model's unnormalised scores z before they are turned into probabilities, $p(v) \propto \exp(z_v/\tau)$. Lower temperature concentrates the distribution on its most likely tokens. It does not make the system deterministic in practice. Atil et al. ran five models at settings meant to be deterministic and found that outputs still varied from run to run, by amounts that differed across models and tasks [37]. Several of the engines in this study do not expose a temperature at all when they are used with reasoning or search. An answer is one sample from a distribution, and Section 11 measures how different two samples are.

2.3 Two sources of a name: parameters and context

Suppose a model is asked for the best project management software for a small agency and writes "Asana". Where did those tokens come from? There are two possibilities, and only two.

Parametric knowledge. During training the model saw a very large corpus and its parameters were adjusted to make the text in that corpus more probable under equation (1). Text in which questions about project management software were followed by the name Asana made that continuation more probable. Nothing about any particular document survives as a record. What survives is a tendency: given a context that looks like a request for project management software, the tokens of certain names have high probability. This is what people mean, loosely, when they say a model "knows" a brand. The tendency is a function of how often and in what company the name appeared in training text, and it is frozen at the training cutoff.

The strength of that tendency follows frequency. Kandpal et al. showed that a model's ability to answer a factual question correlates with, and is caused by, the number of training documents relevant to that question, and that performance on rare facts is poor regardless of model size [15]. Mallen et al. found the same using the popularity of the entity in question, and found that giving the model retrieved text helps most for exactly the unpopular entities the parameters handle badly [16]. For brands, the prediction is direct: a model answering from parameters will name the brands that were written about most before its cutoff, and a brand with little written about it will have a probability near zero no matter how well it fits the question.

Contextual knowledge. The other possibility is that the name was in the context. If the tokens "Asana" appear in the input, in a page that the system placed there, then attention can copy or paraphrase from that page, and the probability of producing the name rises for a different reason. The parameters still do the work of deciding what to do with the text. But the name no longer has to be recalled; it has to be selected from what is in front of the model.

The two sources interact. When the context agrees with what the parameters favour, the answer is easy. When the context contains something the parameters do not favour, the outcome depends on how persuasive the context is. Longpre et al. constructed questions in which the passage supplied contradicted what the model had memorised and found that models frequently answered from memory anyway [17]. Xie et al. revisited the question with more coherent counter-evidence

and found models far more willing to follow the context, together with a confirmation bias towards whichever pieces of evidence agreed with their memory [18]. Wu, Wu and Zou quantified the trade-off and found that the less confident the model's prior, the more readily it adopted the retrieved content [19]. For the present subject the implication is that a retrieved page naming an unfamiliar brand is working against the prior, and a retrieved page naming a familiar brand is working with it. Section 8 finds both effects.

2.4 Retrieval augmentation

Putting retrieved text into the context so that the model can use it is called retrieval-augmented generation, after Lewis et al. [2]. In its general form a retriever takes the user's query q and returns a small set of documents $D = (d_1, \dots, d_k)$ from a corpus, and the generator produces the answer conditioned on both:

$$p(y \mid q) \;=\; \sum_{D} p_\eta(D \mid q)\; p_\theta(y \mid q, D). \tag{4}$$

The first factor is the retriever, classically a learned dense retriever [5]: the probability that a given set of documents is what gets fetched. The second is the generator: the language model of equation (2) with the documents added to its input. In a production answer engine the retriever is a web search service with its own index, ranking function and freshness policy [42, 43], sometimes followed by a step that fetches and trims the pages.

Equation (4) is the structural fact on which this paper is organised. A brand or a page can be affected at two separate stages. It must first be retrieved, which is a matter of the search service's ranking for whatever query was issued, and then, given that it is in the context, it must be used, which is a matter of the language model's behaviour over a long input. The experimental literature on generative engine optimisation studies the second factor with the first held fixed [1, 27]. Conventional search engine optimisation is about the first. A study of production engines observes their product.

Early systems that connected a language model to live web search include WebGPT, which trained a model to issue search queries, follow links and quote passages before answering [6]. The general pattern of letting a model decide when to call an external tool and then continue with the tool's output in its context was formalised in ReAct [7] and Toolformer [8], and Self-RAG trained a model to decide for itself whether retrieval was needed for a given passage [9]. All the engines studied here follow that pattern in some form.

2.5 The search tool call, step by step

Figure 1 shows the sequence for an engine in which the model decides whether to search. The steps are these.

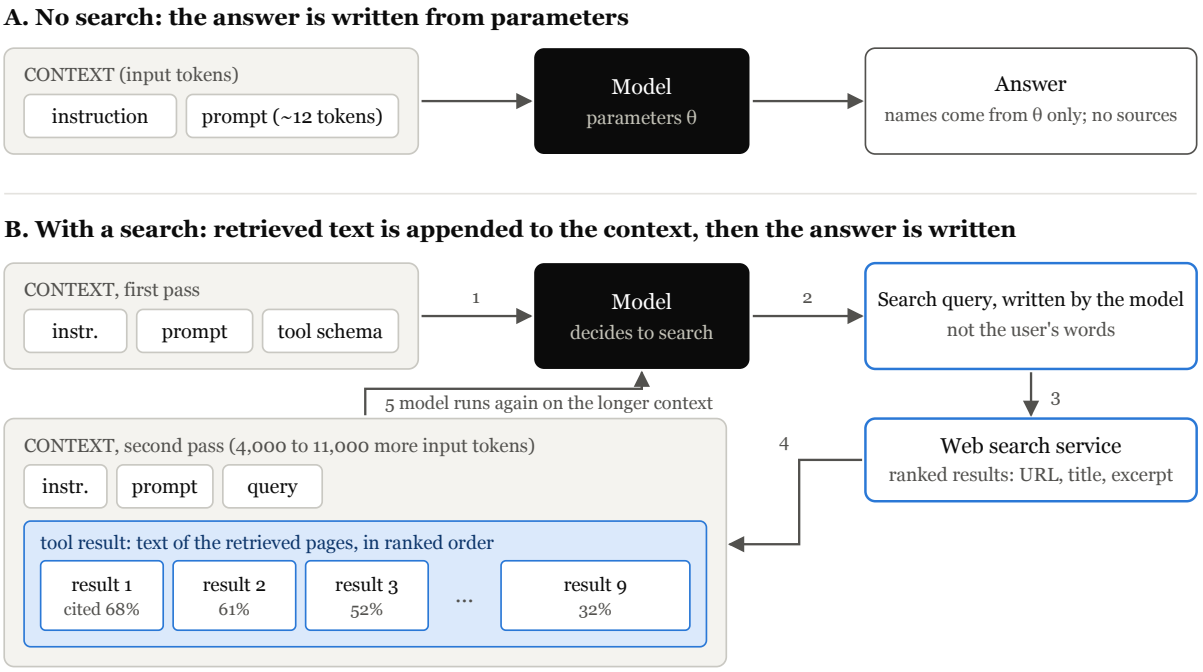


Figure 1. The path from a prompt to a cited answer. Without a search, the context holds only the system instruction and the prompt, and every name in the answer comes from the model's parameters. With a search, the tool's results are appended to the context as input tokens before the answer is written; names and claims can then come from either source, and citations are links from spans of the answer back to results.

Step 1: the request. The application sends the model a context consisting of any system instruction, the user's prompt, and a description of the tools the model may call. A tool description is itself text, a name and a schema, and occupies tokens. This is why an answer that did not search can still be billed far more input tokens than the prompt contains: in our data, Claude answers with no sources were billed a median of 2,544 tokens for prompts with a median of ten words.

Step 2: the decision. The model generates. If the continuation it finds most probable is an answer, it writes the answer and the process ends; every token came from the parameters conditioned on the prompt. If instead the most probable continuation is a tool call, it emits a structured block naming the search tool and giving a query string. The decision to search is thus an output of the same next-token distribution as everything else, shaped by training and by the tool description. It is not a rule the provider applies from outside, unless the caller forces it. When the tool is offered and not forced, whether a given prompt triggers a search is a property of the model, and Section 7 shows it varies systematically with the kind of prompt.

Step 3: the query. The query string is written by the model. It is not the user's prompt. The model may shorten the prompt to keywords, add a year, add the name of a product category, or split one question into several searches. On the one scan for which the queries were logged, ChatGPT is-

sued a mean of 5.0 queries per searched answer and Gemini 3.9, while Claude issued one query for 74% of its searched answers (Section 11.2). The pages that can reach the context are the pages that rank for those rewritten queries.

Step 4: the results. The search service executes the query and returns results. What a result contains differs by provider, but the common elements are a URL, a title, and some portion of the page's text, either a snippet, a set of extracted passages, or a longer excerpt. Anthropic's documentation states that "the API runs the searches and provides Claude with the results" and that "web search results retrieved throughout a conversation are counted as input tokens" [38]. OpenAI's exposes a parameter that "controls how much context from web search results is made available to the model" [39]. Google's states that "the model processes the search results, synthesizes the information, and formulates a response" [40]. Perplexity's interface exposes a cap on "total search context across results" and a per-page cap, both in tokens [41].

Step 5: the second pass. The results are appended to the context as a tool result, and the model is run again on the longer sequence: system instruction, prompt, its own tool call, and now several thousand tokens of page text. It may call the tool again, up to whatever limit the caller set, or it may write the answer. When it writes the answer, each token is drawn from equation (1) with the retrieved text in the conditioning sequence. This is the whole of the mechanism by which a web page influences an AI answer at request time. The page's words became input tokens. Nothing else about the page, its backlinks, its domain's reputation, its ranking in a conventional index, is visible to the model except in so far as those things determined whether and where the page appeared in the results, or are written in the text itself.

Step 6: citations. The answer is returned with source metadata, discussed in Section 2.7.

The size of step 4 can be read from billing. Table 1 gives the median tokens billed per answer in our data for the three engines where the count includes search content, split by whether the answer returned sources.

Table 1. Median tokens billed per answer, with and without sources. Interquartile range in brackets. Claude from 20 October 2026, when the instrument began recording the full usage report; ChatGPT from 16 July 2026 (gpt-5-mini, one search at most); Grok: answers with sources against answers without, the latter almost all from after its search tool was withdrawn. Source: Study 1.

Engine	No sources	Sources returned	Difference	Cliff's δ
ChatGPT	5,695 [5,297 to 6,049]	10,057 [9,779 to 10,349]	4,362	0.98
Claude	2,544 [2,443 to 2,631]	11,317 [10,786 to 14,914]	8,774	1.00
Grok	866 [745 to 1,003]	11,832 [8,354 to 18,853]	10,966	1.00

A single search adds between about 4,000 and 11,000 tokens in these configurations, the equivalent of 3,000 to 8,000 words of page text, to a context whose user-written part is around a dozen tokens. Cliff's δ, the probability that a randomly chosen sourced answer was billed more tokens

than a randomly chosen unsourced one minus the reverse probability [58], is at or within two hundredths of its maximum for all three: the two distributions barely overlap. ChatGPT's unsourced figure is high because gpt-5-mini is a reasoning model whose hidden reasoning tokens are billed as output, and because some answers searched and returned no citation annotations; the 4,362-token difference is therefore a lower bound on what its single permitted search contributes. Gemini and Perplexity do not report retrieved text in their token counts, and Google AI Overview is read through a search results API that reports none.

2.6 What the model sees of a page

It is easy to imagine that the model reads the pages a search returns. It reads what the search tool passes on, which is less and is shaped by choices the model's caller does not control. Three properties of that text matter for everything downstream.

It is an excerpt. No provider passes whole pages for every result at these token counts. Ten results in about 9,000 tokens is 900 tokens each, roughly 700 words, against a median fetched page in our corpus of 1,216 words and a mean far higher. What is excerpted, the opening of the page, the passages that best match the query, or a summary, is the provider's decision. A fact that appears only deep in a long page may never be placed in the context even though the page is "retrieved".

It is flattened. The model receives text. Layout, visual emphasis, images, the difference between a main column and a sidebar, and anything rendered by client-side code that the fetcher did not execute are absent or reduced to plain text. Structure survives only where it is expressed in the text itself: headings, list items, table rows.

It is ordered. The results arrive in a sequence, normally the search service's ranking. By equation (3) and by the empirical work on long contexts [14], order is not neutral. Section 10 measures it directly: the first result Claude was given was cited 68% of the time and the ninth 32%.

There is a further consequence for studies like this one. The quantity that a careful experiment would want to manipulate, the text of a page as it was presented to the model, is not observable from outside for any engine. Anthropic returns the content of results in encrypted form [38]. We can observe which URLs were returned, in what order, with what titles, and which were cited. We can separately fetch those URLs and describe the pages. We cannot see the excerpt. Every page-level feature in Section 10 is therefore a feature of the page as we fetched it, used as a proxy for the excerpt the model saw.

2.7 How a citation is attached

A citation, as returned by these APIs, is a record that links a source to the answer. The engines differ in what the record contains and in how it is produced, and the differences determine what can be learned from each.

Claude. The response is a sequence of blocks. A search produces a tool-result block listing every result returned, each with a URL and a title. Text blocks of the answer then carry a list of citations, each of which names a result and quotes the passage of it that supports that block of text. Anthropic's documentation states that "citations are always enabled for web search" [38]. This structure distinguishes three sets: results retrieved, results cited, and, for each cited result, the answer text it supports and the source text relied on. It is the only engine in the study for which the first two sets are separately observable, and Study 4 is built on that.

ChatGPT. The answer text carries inline annotations of type `url_citation`, each giving a URL, a title and the character positions in the answer it applies to. OpenAI's documentation describes a separate field that "returns the complete list of URLs the model consulted" [39]; the instrument used here did not request it, so for ChatGPT only cited URLs are observed. In the rendered answer each annotation appears as a bracketed link whose visible label is the source's domain. That detail produced one of the artefacts described in Section 5.6.

Gemini. With Google Search grounding, the response carries grounding metadata: the search queries issued, a list of source "chunks", and "supports" that map segments of the answer to the chunks that back them [40]. Source URLs are returned as opaque redirect links, so the publisher's domain must be recovered from the title.

Perplexity. The response includes a list of search results with titles and URLs. The answer text refers to them by bracketed index. The list is what was searched, and the instrument stores all of it; which entries are referred to in the text was not parsed, so for Perplexity "cited" in this paper means "returned as a source".

Google AI Overview. This is not a model API. The overview shown on a Google results page is read through a third-party search results API, which returns the overview's text blocks and its list of references, with some references tied to specific blocks and others listed for the overview as a whole.

Grok. During the weeks its search tool was sent, Grok returned `url_citation` annotations in the same form as ChatGPT. Afterwards it returned none, because it did not search.

Two points about citations hold across engines. First, a citation is generated or assembled alongside the answer; it is not a proof that the answer came from the source. Liu, Zhang and Liang had annotators check four generative search engines and found that 51.5% of generated sentences were fully supported by their citations and that 74.5% of citations supported the sentence they were attached to [13]. The Tow Center's test of eight tools on news excerpts found incorrect attributions in more than 60% of queries [32]. A citation in this paper means only that the engine returned the URL as a source. Second, a citation is a property of the answer, and a mention is a property of the answer's prose. A page can be cited for a claim that has nothing to do with the brand that published it, and a brand can be named in a passage whose citation points somewhere else entirely. Section 9.6 measures how often the two coincide.

2.8 Engines that always search, sometimes search, or never search

The seven engines in the panel fall into three groups by how search is invoked, and the grouping determines which comparisons each engine can support. Appendix A gives the full configuration of each.

Always search. Perplexity's Sonar models and Google AI Overview are search products. Every request is grounded in retrieved results; there is no ungrounded mode to compare against. In the analysed data virtually every Perplexity answer and 97% of AI Overview reads carried sources, the remainder being reads where the source list came back empty.

Search when the model chooses. ChatGPT (gpt-5 until 14 July 2026, gpt-5-mini from 15 July), Claude and Gemini were called with a search tool available and the decision left to the model, with a cap on the number of searches per answer: one for ChatGPT from 8 July, five for Claude. The same prompt can therefore receive a grounded answer on one occasion and an ungrounded one on another.

Never search, or not any more. Copilot, read through Azure until it was replaced by AI Overview on 15 July 2026, mostly answered from parameters. Grok was sent a search tool from 9 June to 7 August 2026 and none thereafter; from mid-August all of its answers are from parameters alone. Grok after that date is the paper's reference for what a model says without retrieval, and on the category prompt sets it supplies the measure of each brand's prior.

Before 9 June 2026 the instrument did not store sources for any engine and called ChatGPT, Claude and Grok with no search tool. That early period, 15,313 answers, is excluded from every analysis except the sample description, because the engines, their models and the instrument all changed at once on that date and no contrast across it can be attributed to any one of them.

2.9 What follows from the mechanism

The description above yields predictions that can be tested with the data in hand, and it is worth stating them before the related work, because they are consequences of the mechanism alone.

If a name can come only from parameters or from context, then removing the search tool should reduce the naming of brands the parameters do not favour and leave well-known brands largely untouched. If the decision to search is a model output, it should vary with the prompt in ways a model would plausibly have learned: more for questions whose answers change, such as prices and "best of" lists, less for definitions. If retrieved text competes for attention under a fixed budget, citation should fall with position in the context and should favour text that matches the question closely. If the prior and the context interact as the knowledge-conflict literature describes, then having one's page retrieved should matter most for brands with weak priors and should not be sufficient for any brand. And if an answer is a sample, two readings of the same prompt should differ, and should differ more where retrieval adds a second source of randomness, since the search results themselves change from request to request.

Each of these is examined in Sections 7 to 11. Most hold. Where they do not, the exceptions are informative.

3. Related work

Five bodies of work bear on the question. The first optimises content for inclusion in generated answers. The second audits what production answer engines cite. The third studies how language models use, weigh and are misled by the text in their context. The fourth concerns what models retain about entities, and brands in particular, in their parameters. The fifth is the measurement literature on the reproducibility of model outputs. We take them in turn and then say where the present study sits.

3.1 Optimising content for generative engines

The term generative engine optimisation was introduced by Aggarwal et al. [1]. Their contribution was a framing, a benchmark and a set of interventions. The framing treats a generative engine as a system that retrieves a set of sources and synthesises a cited response, and defines a source's visibility by its share of the response. Because a generated answer has no ranks, they proposed impression measures in place of position: the word count of the sentences attributed to a source, the same weighted by how early those sentences appear, and a subjective score from a language model judge. The benchmark, GEO-bench, has ten thousand queries across domains. The interventions rewrite a single source in one of nine ways and measure the change in its visibility with the other sources fixed. The reported result is that adding quotations, adding statistics and citing sources each raised visibility by up to 40% on position-adjusted word count, that improving fluency helped, and that keyword stuffing, the staple of older search optimisation, did not help and could hurt. The gains were larger for sources that ranked lower in the underlying search results, and the authors validated the leading methods on Perplexity.

Two features of that design shape how its results should be read alongside ours. The first is that retrieval is fixed. The five sources given to the model are the top five Google results for the query, and an intervention changes the text of one of them without changing whether it is among the five. In the terms of equation (4), the study estimates effects on $p_\theta(y \mid q, D)$ for a given D. The second is that the outcome is a source's share of the answer's text, which is a citation outcome. Whether the brand that published the source is named in the prose is not measured.

Puerto et al. revisited the question with C-SEO Bench [27], a benchmark covering question answering and product recommendation across six domains, with an evaluation protocol that varies how many of the competing sources apply an optimisation method at once. Their conclusion is considerably less encouraging than the original: most current methods for conversational search optimisation were "largely ineffective", traditional search optimisation that improves a document's position in the context was more effective, and whatever gains there were shrank as more actors

adopted the same method. The finding that position in the context matters more than rewriting is directly relevant to Section 10, where retrieval position is the strongest predictor of citation by a wide margin.

A parallel literature treats the same surface as an attack target. Kumar and Lakkaraju showed that appending a crafted text sequence to a product's information page could make a model far more likely to list that product as its top recommendation, using a catalogue of fictitious coffee machines [24]. Pfrommer et al. studied prompt injections that reorder the products a conversational search engine recommends and found they could reliably promote low-ranked products [25]. Nestaas, Debenedetti and Tramèr demonstrated that content crafted for the purpose could bias the selections of production systems [26]. These papers establish that the content of the context has causal force over which products are named, which is the premise of Section 8, and they do it with adversarial text that honest publishers would not write.

Kumar and Palkhouski [29] take an observational approach closer to ours on the page side, relating the citation behaviour of answer engines in business software to a sixteen-item framework of page-quality signals. Observational work on page features faces a common difficulty: cited pages have to be compared with something, and the natural comparison, pages the engine itself retrieved and declined to cite, is rarely observable. Section 10 uses exactly that comparison for the one engine that exposes it.

3.2 Audits of what answer engines cite

Liu, Zhang and Liang [13] conducted the first careful audit of citation quality in generative search. They defined citation recall, the share of generated statements fully supported by their citations, and citation precision, the share of citations that support their statement, and had human annotators evaluate four systems. Responses were fluent and appeared informative, but only 51.5% of sentences were fully supported and 74.5% of citations supported their sentence, and the systems that scored best on perceived utility scored worst on these measures. Gao et al. built ALCE, a benchmark for generating text with citations, and found that even the best models lacked complete citation support a substantial share of the time [10]. Bohnet et al. formalised attributed question answering and compared architectures for it [11], and Menick et al. trained models to back answers with verbatim quotes from sources [12], an approach that anticipates the quoted-passage citations Claude's search tool returns today.

Narayanan Venkit et al. combined a usability study of 21 participants with an automated evaluation of three answer engines and catalogued sixteen limitations, among them citations that did not support the claims they were attached to and sources retrieved but not used [31]. The Tow Center for Digital Journalism gave eight AI search tools excerpts from news articles and asked for the source; the tools answered incorrectly for more than 60% of the 1,600 queries, with error rates from 37% for Perplexity to 94% for Grok 3, and they rarely declined to answer [32].

These audits ask whether a citation is correct. A second group asks which sources are cited. Zhang et al. [30] is the largest: 55,936 queries, six LLM-based engines and two conventional ones, 1.4 million citation links. LLM-based engines returned fewer sources per response than conventional search, a mean of 4.3 URLs from 3.4 domains against 10.3 and 7.3, and, ChatGPT excepted, less popular domains. Only 38% of domains were common to both kinds of engine, and 37% appeared only in the LLM-based engines' results. Chen et al. [28] classified cited domains as brand-owned, earned or social and reported that AI engines lean heavily towards earned media, independent publications and review sites, where Google's results are more balanced, and that the engines differ from each other in domain diversity, freshness and sensitivity to phrasing. On the demand side, the Pew Research Center's panel study measured what an AI summary does to clicking, as cited in Section 1 [33].

Our Study 3 reports the same descriptive quantities for a different and narrower population of prompts, and adds three things these studies could not measure: agreement between engines on the same prompt at the same time, the stability of one engine across repeated readings of one prompt, and the relation between a site being cited and its owner being named.

3.3 How models use the text in their context

The behaviour of a language model over a long context is the proximate cause of everything a retrieved page can do, and a substantial literature now describes it.

Position. Liu et al. [14] varied where the relevant document sat among distractors in a multi-document question answering task and found a U-shaped curve: accuracy was highest when the relevant document was at the beginning or end of the context and fell, sometimes below the model's closed-book accuracy, when it was in the middle. Position effects have a long history in search itself. Users of ranked result lists click the first result far more than the second regardless of relevance, a bias documented with eye tracking by Joachims et al. [44] and modelled by Craswell et al. [45]. A language model reading ranked results inherits an analogous bias through a different mechanism, and the two compound: the search service puts what it judges best first, and the model attends most to what is first.

Relevance and style. Wan, Wallace and Klein [20] asked what makes a document convincing to a model when retrieved documents disagree. They assembled real web pages on contested yes-or-no questions and measured how often the model sided with each. Models relied heavily on how relevant the page was to the query and largely ignored stylistic features that people treat as marks of credibility, such as scientific references and a neutral tone. That finding sits awkwardly beside the GEO-bench result that adding citations and statistics raises visibility, and the two are not strictly comparable, since one measures agreement with a document's stance and the other its share of the answer. Our results in Section 10 are closer to Wan et al.: lexical match between a result's title and the prompt is among the strongest predictors of citation, and most structural features of the page are weak.

Distraction and noise. Shi et al. [21] showed that adding irrelevant sentences to a problem description lowers a model's accuracy markedly: models attend to what is in front of them whether or not it bears on the question. Cuconasu et al. [22] studied what kinds of retrieved passage help or hurt in retrieval-augmented systems and found that passages that are topically related but do not contain the answer are the most damaging kind.

Conflict with the prior. The knowledge-conflict studies introduced in Section 2.3 [17-19] establish that neither source wins unconditionally. Models hold to memory against weak or incoherent context, follow coherent context even against memory, favour context that confirms what they already held, and defer to context more when their prior is uncertain.

Source bias in the retriever. Before the model sees anything, a retriever has chosen it. Dai et al. [23] found that neural retrievers rank text generated by language models above human-written text of comparable relevance. Where answer engines use neural ranking, the pages reaching the context may be skewed towards machine-written prose before the generator's own preferences apply.

3.4 What models hold about entities and brands

The frequency dependence of parametric knowledge [15, 16] was described in Section 2.3. For brands specifically, the evidence is recent and not entirely consistent. Kamruzzaman et al. [34] probed models for associations between brand names and attributes and found a consistent pattern of global brands being associated with positive attributes and local brands with negative ones, together with country-of-origin effects. Chu and Hou [35] held product specifications identical and varied only the brand name across three commercial models; an established brand was recommended over an unknown one in every trial of their first experiment. Malthouse et al. [36], by contrast, evaluated brand retrieval and ranking in recommendations across product categories and reported limited evidence of the popularity bias familiar from conventional recommender systems, with the relation varying substantially by category, and found that prompts naming only a category often omitted large established brands altogether.

The disagreement may be one of measurement. A popularity bias measured against an external index of brand size can be weak while the model's own prior, whatever its origin, is strongly self-consistent. Study 2 measures the prior from the inside: how often an engine with no retrieval names each brand for each category. That quantity needs no external definition of popularity, and it predicts other engines' behaviour strongly. It is also the quantity that the heavy-tailed distributions of the web would lead one to expect. Text about companies is distributed very unevenly, a few names account for much of it, and processes of cumulative advantage [64] reproduce themselves in any system trained on the result.

3.5 Reproducibility of model outputs

That generation is stochastic is by design; that it is hard to make deterministic is an empirical finding. Atil et al. [37] ran five models configured for determinism on eight tasks ten times each and found that raw outputs frequently differed between runs and that even parsed final answers were not perfectly stable, with the degree of instability varying by model and task. For answer engines a second source of variance is added, since the retrieved set itself varies between requests, and a third, since providers update models and search back ends without announcement. Zhang et al. [30] and Chen et al. [28] both note sensitivity of cited sources to phrasing. We are not aware of a published measurement of how much an answer engine's named brands or cited sites vary when the identical prompt is repeated over days and weeks. Section 11.3 provides one.

3.6 Statistical methods for clustered and heavy-tailed data

Several methodological traditions are used in the analysis and are cited where applied. Observations in the panel are clustered within prompts and within accounts, so standard errors that assume independence would be far too small; we use cluster-robust inference [48], the cluster bootstrap [47], generalised estimating equations [49], and conditional likelihood methods that remove cluster-level heterogeneity altogether [50, 51]. The withdrawal of Grok's search tool is analysed as a difference in differences, with attention to the serial-correlation problem identified by Bertrand, Duflo and Mullainathan [52]. Heavy-tailed citation counts are fitted by the maximum-likelihood method of Clauset, Shalizi and Newman [54], who showed that the common practice of fitting a line to a log-log plot is unreliable, using the implementation of Alstott et al. [55]. Multiple comparisons within a family of hypotheses are controlled by the false discovery rate procedure of Benjamini and Hochberg [53].

3.7 Position of this study

Table 2 places the study against the most closely related work on four dimensions: whether production engines were observed, whether retrieval varied, whether the outcome included brands named in prose, and whether the same query was observed repeatedly.

Table 2. The present study against the closest related work.

Study	Production engines	Retrieval observed or varied	Mentions in prose	Repeated readings
Aggarwal et al. [1]	Simulated, validated on one	Held fixed	No	Five samples per query
Puerto et al. [27]	Simulated	Held fixed	Recommendation task	No
Liu, Zhang and Liang [13]	Four	No	No	No
Zhang et al. [30]	Six, plus two conventional	No	No	No
Chen et al. [28]	Several, plus Google	No	No	Across phrasings
Wan, Wallace and Klein [20]	Models with supplied documents	Held fixed	No	No
Chu and Hou [35]	Three models, no retrieval	Not applicable	Yes	Repeated trials
This study	Seven	Search on or off per answer; retrieved and cited separated for one engine	Yes	Up to ten per prompt and engine over months

The study's distinctive ground is the combination in the last row. Its distinctive weakness is that nothing was randomised by the investigator. Where the experimental literature can say that a change to a page caused a change in the answer, this study can say what production engines did, at scale, over time, and can use a small number of naturally occurring contrasts to say something about cause.

4. Hypotheses

The research questions of Section 1.2 are stated here as hypotheses, each with the reasoning that motivates it and the analysis that tests it. The study plan listed the analyses before any was run, but the hypotheses in the form given here were written with the results in hand, and none was preregistered. The study is therefore exploratory in the technical sense: the hypotheses organise the analysis, each follows from the mechanism in Section 2 or from cited prior work and not from the data, and the p-values attached to them are not those of a confirmatory trial. Where a result ran against a hypothesis, we report it as such.

Throughout, an answer is indexed by i, a prompt by p, an engine by e, an account by a and a brand by b. The indicator S_i equals one if answer i returned at least one source, M_{ib} equals one if brand b is named in answer i, and C_{iu} equals one if URL u is cited in answer i.

4.1 Search (RQ1)

H1a. The decision to search depends on the kind of prompt. For engines that choose, $\Pr(S_i = 1)$ is higher for prompts that ask for a comparison or a recommendation than for prompts that ask for an explanation, and higher for prompts containing recency or price terms. *Reasoning:* the decision is a model output (Section 2.5), and a model trained to use a search tool

sensibly will have learned to use it where its parameters are likely to be stale. *Test:* logistic regression of S_i on prompt category and lexical features, per engine, with errors clustered on account (Section 7.2).

H1b. A search adds several thousand input tokens. *Reasoning:* vendor documentation states that results are counted as input tokens [38]. *Test:* comparison of tokens billed for answers with and without sources (Section 2.5, Table 1, and Section 7.3).

4.2 The effect of search on naming (RQ2)

H2a. Removing the search tool lowers the probability that a brand is named in answers to unbranded prompts. *Reasoning:* the brands tracked in the panel are mostly small companies whose names the parameters are unlikely to favour. For such a brand the context is the main route into an answer, and a search is what fills the context. *Test:* difference in differences around the withdrawal of Grok's search tool (Section 7.4).

H2b. For the same prompt and engine, answers with sources name the brand more often than answers without. *Test:* paired comparison within prompt, sign test and conditional logistic regression (Section 7.5).

H2c. Search has little effect on naming for branded prompts. *Reasoning:* when the prompt already contains the brand's name, the name is in the context whether or not a search runs. *Test:* the same analyses restricted to branded prompts.

H2d. Answers with sources name more brands, and name brands the non-searching engine does not. *Reasoning:* retrieved pages put names in the context that the parameters would not have produced. *Test:* brands per answer and the share of named brands with a zero prior, on the category prompt sets (Sections 8.1 and 8.2).

4.3 Prior and retrieved text (RQ3)

H3a. The prior predicts naming in sourced answers. A brand that the non-searching engine names often for a category is more likely to be named by a searching engine for that category. *Reasoning:* the generator's parameters still decide what to do with the retrieved text, and the knowledge-conflict literature finds a confirmation bias towards context that agrees with memory [18].

H3b. Retrieved text predicts naming over and above the prior. A brand whose own site is among the sources, or whose name appears in a source's title, is more likely to be named, conditional on its prior.

H3c. The effect of retrieved text is larger, in relative terms, for brands with weak priors. *Reasoning:* models defer to context more when their prior is uncertain [19], and retrieval helps most for unpopular entities [16].

Test for H3a to H3c: logistic regression at the level of the answer and brand pair with errors clustered on prompt, with the prior entered at two resolutions, and a tabulation of naming rates by prior band and source status (Section 8.3).

4.4 Citations (RQ4)

H4a. Citations are heavy-tailed over sites. The distribution of citation counts per site has a power-law tail. *Reasoning:* attention on the web is heavy-tailed, search rankings concentrate it further, and each engine draws on a search service. *Test:* maximum-likelihood power-law fit with a likelihood-ratio comparison against a log-normal (Section 9.1).

H4b. Engines agree weakly on what to cite. For the same prompt in the same run, the overlap between two engines' cited sites is low. *Reasoning:* the engines use different search back ends, issue different rewritten queries, and receive different excerpts. *Test:* Jaccard similarity over engine pairs (Section 9.4).

H4c. One engine's citations for one prompt are unstable across readings, and less stable the further apart the readings are. *Test:* Jaccard similarity between consecutive readings and its rank correlation with the gap in days (Section 9.5).

H4d. An account's own site being cited is associated with the account being named, and is far from sufficient for it. *Reasoning:* mention and citation are separate outputs (Section 2.7). *Test:* conditional logistic regression within prompt and engine, and the conditional proportion (Section 9.6).

4.5 Citation conditional on retrieval (RQ5)

H5a. Earlier results are cited more. Within an answer, the probability that a retrieved result is cited decreases with its position in the context. *Reasoning:* position effects in long contexts [14] and the search service's own ordering by relevance.

H5b. Results whose titles match the prompt are cited more, conditional on position. *Reasoning:* relevance dominates what models find convincing [20].

H5c. Page structure associated with extractable passages predicts citation. Pages with a higher share of short self-contained paragraphs and a higher density of statistics are cited more, conditional on position. *Reasoning:* a citation attaches to a passage; a page that offers passages which stand alone gives the model something to attach it to. The GEO-bench interventions that worked best added statistics and quotations [1].

H5d. Markup conventions promoted for AI visibility, namely FAQ blocks, summary boxes, definition-style openings and question-form headings, predict citation. This hypothesis is included because the practices are widely recommended, not because the mechanism in Section 2

predicts them. A model sees flattened text; whether a block was marked up as an FAQ is visible only in so far as the text reads as questions and answers.

Test for H5a to H5d: conditional logistic regression with the answer as stratum, on all retrieved results for the title and site variables and on the subset with fetched pages for the structural variables, with Benjamini-Hochberg correction across terms (Section 10).

4.6 Reliability (RQ6)

H6a. A single reading is an unreliable indicator of whether a brand is named for an unbranded prompt. The probability that a brand named in one reading is named in the next reading of the same prompt by the same engine is well below one. *Test:* test-retest conditional probabilities, Cohen's κ and the intraclass correlation (Section 11.3).

H6b. Branded prompts are more reliable than unbranded ones. *Reasoning:* the name is in the prompt on every reading.

4.7 Hypotheses the data cannot test

Three hypotheses that the mechanism suggests are stated for completeness and are not tested, because the panel lacks the variation needed.

The excerpt, not the page, determines citation. Testing this requires the text the search tool passed to the model, which no engine exposes in readable form (Section 2.6).

Retrieval rank for the model's rewritten query determines whether a page is in the context. Testing this requires the search service's full ranking for each issued query. We observe only the results returned, and for one engine.

The frequency of a brand in pre-cutoff text determines its prior. We measure the prior as behaviour and do not have counts of brand occurrences in any training corpus. The link between the two rests on the literature cited in Section 2.3.

5. Data

5.1 The instrument

The data were collected by RankMonster, a commercial service that tracks how AI answer engines respond to prompts relevant to a customer's business. A customer saves a website domain, which identifies their brand, and a set of prompts, either written by the customer or generated for them from a description of their business and reviewed. A scan sends each active prompt to each engine on the customer's plan through that engine's API, stores the full text of the response, the sources

returned with it, the token count and the latency, and records how many times the customer's brand is named. Scans are started by the customer or, for some accounts, by a weekly schedule. There is no daily scan; a typical active account is scanned three or four times in four weeks.

The service was not designed as a research instrument, and three consequences of that should be kept in mind. The prompts are whatever customers wanted to track, so they over-represent commercial questions about software and professional services. The accounts differ enormously in size, so a few of them supply a large share of the answers. And the engine configurations changed over the period for operational reasons, mostly cost. Each of these is a threat to some inference and an opportunity for another, and both sides are discussed where they arise.

A second instrument on the same infrastructure supplies the category data. For a product or service category, a fixed set of about forty prompts is generated from templates covering the questions a buyer might ask, frozen with a checksum, and sent to six engines once a month. Each answer is passed to an extraction step that identifies every brand named, resolves each to a canonical brand record, and records where in the answer it first appears and whether it was recommended, merely mentioned, or mentioned negatively. The category instrument exists to publish leaderboards and has no account or customer attached to it.

5.2 Engines and configurations

Table 3 summarises how each engine was called during the analysis period. Appendix A gives the request parameters in full. All calls were made from servers in the United States, in English, with no user identity, location or conversation history.

Table 3. Engine configurations during the source era (12 June to 14 November 2026).

Engine (label in this paper)	Model or product	System instruction	Search	Limit
ChatGPT	gpt-5, then gpt-5-mini from 15 July	None	Tool offered, model decides	Two searches, then one from 8 July
Claude	claude-haiku-4-5	"Answer concisely and accurately"	Tool offered, model decides	Five searches
Gemini	gemini-2.5-flash	"Answer concisely and accurately"	Google Search grounding, model decides	Output capped at 400 tokens until 9 November
Perplexity	sonar	"Answer concisely and accurately"	Always	Low search context, 400 output tokens
Google AI Overview	Overview on the results page, read through a search results API	Not applicable	Always	One read, retried on a miss
Grok	grok-4.3	"Answer concisely and accurately"	Tool offered 9 June to 7 August, then none	2,000 output tokens
Copilot	Azure-hosted model, to 15 July	Not recorded	Occasional	Replaced by AI Overview

Three points about these configurations bear on interpretation. First, the labels name API configurations that emulate the corresponding consumer products and are not those products. The ChatGPT a person uses has a long system prompt, memory of the user, a different search budget

and a different model router. Our "ChatGPT" is gpt-5-mini with web search available, one search permitted, and nothing else in the context but the prompt. The emulation was tuned to be as bare as possible on the reasoning that every added instruction moves the answer away from what the underlying model would say, but bare is not identical. Second, the limits on searches and output length are tighter than the consumer products'. A ChatGPT limited to one search per answer cannot do the multi-step research the product does. Third, Gemini's output cap had a side effect that matters for the analysis and is described in Section 5.6.

5.3 The answer panel

The analysis file was drawn from the production database on 15 November 2026 through a read-only session. It contains every stored answer with status "completed" and non-empty text: 119,555 rows. Of these, 21,735 had been served from the instrument's response cache, which returns a stored response when the same prompt was sent to the same engine recently. A cached row is a copy of an earlier answer and not a new observation, so cached rows were dropped, leaving **97,820 freshly generated answers** from 501 accounts, 10,802 prompts and 700 scans, dated 7 April to 14 November 2026.

The panel divides at 9 June 2026. Before that date the instrument stored no sources, and ChatGPT, Claude and Grok were called without a search tool; on that date the search-enabled clients and the storage of sources were deployed together. The 15,313 answers from before 9 June are described here and excluded from all analyses. The three days from 9 to 11 June, during which old and new clients were both in service, are also excluded. The **source era** runs from 12 June to 14 November and contains 80,961 fresh answers from 253 accounts.

Three further exclusions were applied within the source era.

Onboarding audits. When a prospective customer signs up, the service runs a short audit of a few prompts and shows the result. The audit writes to the same tables as a scan but goes through its own, older clients, which do not search and store no sources. An audit is recognisable in the data as a run in which no engine returned a single source, Perplexity included: 199 runs, 5,680 answers, from 193 accounts, with a median of 23 answers each. They were removed. The remaining 207 runs are **full scans**: 75,281 answers from 116 accounts. Section 5.6 explains what the audits did to the results before they were removed.

Fragments. An answer of fewer than 40 words was treated as a failed read: 5,084 answers.

Unscorable accounts. An answer can be scored for a mention only if the account's saved domain yields a usable brand label. Accounts with a missing domain, a label shorter than four characters, or a domain on a shared platform whose label would be the platform's name were set aside for mention analyses: 1,561 further answers. They remain in the citation analyses of Study 3, which do not need a label.

The **analysis set** is the remaining 68,636 answers, from 112 accounts and 7,711 prompts. Table 4 gives its composition by engine.

Table 4. The answer panel by engine. "Fresh" excludes cached rows; "analysis" is the source-era set after removing fragments and unscorable accounts.

Engine	Fresh answers	Full scans, source era	Fragments	Analysis set	First to last answer
ChatGPT	18,836	14,810	76	14,318	13 Apr to 14 Nov
Claude	12,381	8,476	1	8,465	7 Apr to 14 Nov
Gemini	18,196	14,515	4,230	10,046	17 Apr to 14 Nov
Perplexity	18,980	14,781	51	14,289	17 Apr to 14 Nov
Google AI Overview	6,542	6,542	23	6,510	16 Jul to 14 Nov
Grok	19,627	15,346	700	14,200	17 Apr to 14 Nov
Copilot	3,258	811	3	808	17 Apr to 15 Jul
Total	**97,820**	**75,281**	**5,084**	**68,636**	

5.4 Prompts

Prompts are short. The median prompt in the analysis set has ten words. Each prompt carries a category assigned when it was created, one of five, and Table 5 shows how the categories differ on the quantities the paper studies.

Table 5. The analysis set by prompt category. "Branded" is the share of answers whose prompt contains the account's brand label; "with sources" and "brand named" are shares of answers.

Category	Typical form	Answers	Branded	With sources	Brand named
Awareness	"what is X", "how does X work"	19,388	1.2%	66.8%	1.4%
Recommendation	"best X for Y"	27,849	0.3%	75.4%	4.5%
Comparison	"A vs B", "alternatives to A"	9,668	43.3%	75.7%	41.9%
Industry	questions about a sector	6,453	0.0%	69.9%	0.9%
Use case	"how do I do X with a tool"	5,278	0.2%	72.1%	1.9%

The table makes visible the single most important structural fact about the mention data. Whether the account's brand is named depends overwhelmingly on whether the prompt already names it. Comparison prompts are branded 43% of the time and name the brand 42% of the time; every other category is branded almost never and names the brand between 1% and 5% of the time. Across the analysis set, 4,515 answers (6.6%) are to **branded prompts**, defined as prompts whose text contains the account's brand label, and the brand is named in 79% to 97% of those depending on the engine. In the remaining **unbranded prompts** the brand is named in 0.7% to 4.2% of answers.

These are different phenomena and pooling them is a mistake. A mention in the answer to "is Acme any good" tells us whether the engine recognised and discussed a name it was handed. A mention in the answer to "best tools for X" tells us whether the engine brought the name up un-

prompted, which is what a brand seeking discovery wants and what Section 2 predicts is hard. Every mention analysis in the paper is run separately for the two kinds of prompt, and the unbranded results are the ones of primary interest.

The low base rate for unbranded prompts is itself a finding and a constraint. It is the reason the paper leans on the category data for questions about which brands are named: an outcome that occurs in two or three observations in a hundred, concentrated in a few accounts, supports only coarse inference.

5.5 Accounts and the clustering problem

Answers are not independent draws. They are nested in prompts, and prompts are nested in accounts, and each account contributes one brand in one market. The 112 accounts in the analysis set are very unequal: the median account contributes 260 answers, the 90th percentile 1,065, and the largest 12,861. The two largest accounts together supply 30.2% of the analysis set and the ten largest 61.2%. Sixty-five accounts have at least a hundred answers.

The two largest accounts are operated by the author's own company, for its two products. They are the longest-running and most-scanned accounts because they are where the instrument was developed. This is a conflict of interest and a statistical problem. As a conflict, it means that the brands whose mention rates carry the most weight in pooled statistics are brands the author wishes to see mentioned. As a statistical problem, it means that a naive standard error computed over answers would treat tens of thousands of observations of two brands as if they were tens of thousands of independent brands. We handle it in three ways, detailed in Section 6: every interval for a rate or a difference of rates is computed by resampling whole accounts; regression models report errors clustered on account or on prompt; and the main contrasts are also estimated within prompt, where the account and brand are held fixed by construction. Where the data allow, we also report results with the two largest accounts removed.

5.6 Five measurement artefacts and their corrections

During the analysis five properties of the stored data distorted the results, and three of them each produced a finding that was large, statistically significant and wrong. We describe them in some detail because they are not peculiar to this instrument, and because the first drafts of several headline numbers were changed by correcting them.

Artefact 1: truncated answers masquerading as answers without search. The first within-prompt comparison of answers with and without sources showed that Gemini named the brand less than half as often when it had sources (odds ratio 0.41, $p = 0.0002$), and Claude less than a sixth as often. Taken at face value, search suppressed mentions. The cause was Gemini's output cap. With grounding enabled, Gemini's 400-token output budget was frequently consumed before the visible answer was complete, and what was stored was a fragment: among Gemini answers with no sources, the median length was 14 words, and 77% were under 40 words. A fragment has no sources and usually no brand, but occasionally consists of little more than the opening words

of a reply that restates the prompt, brand included. The fragments were not ungrounded answers. They were failed reads. Removing every answer under 40 words from every engine eliminated the effect: on unbranded prompts answered both ways, Gemini named the brand in 0.53% of sourced and 0.24% of unsourced answers, with four informative prompts, all favouring search.

The fragment filter removes 29% of Gemini's full-scan answers, 5% of Grok's and under 1% of every other engine's. A side effect is that Gemini's share of answers with sources in the analysis set (92%) overstates how often Gemini searches, because the fragments were disproportionately unsourced. We report it with that caveat. The general lesson is that "no sources returned" is not a clean indicator of "did not search". It also contains reads that failed for reasons correlated with the outcome.

Artefact 2: source chips counted as mentions. ChatGPT renders each citation inline as a bracketed link whose visible label is the source's domain. If an account's own site is cited, the account's domain therefore appears in the text of the answer. A matcher that keeps link labels and looks for the brand label will find it there. In the first pass this made an own-site citation perfectly predictive of a mention on ChatGPT: the brand was "named" in 100% of unbranded ChatGPT answers that cited the account's site, and the within-prompt effect of search on ChatGPT mentions was large and highly significant (21 prompts favouring search to none against). The production instrument had met and fixed the same problem in October 2026, and we adopted its rule: a markdown link whose label is nothing but its own target's domain is a citation marker and is removed before matching; any other link label is text a reader sees and is kept. With chips removed, and under the matching rule then in use, ChatGPT named the brand in 18% of the answers that cited its site, and the within-prompt effect shrank to five prompts against none. (Under the final matching rule described below the figures are 67% and ten prompts against none; the point here is that the chip alone had produced 100%.)

The lesson generalises beyond one vendor's rendering. Mention and citation are measured from the same response, and any leak from the citation channel into the text channel manufactures an association between them. Section 9.6, which estimates that association, depends on this correction.

Artefact 3: a changing matcher. The instrument's own mention counter was revised twice in the period, in July and in October 2026, the second time to count a brand's name written as separate words and to stop counting source chips. A comparison across either date using the stored counts would confound a change in the engines with a change in the ruler. All mention outcomes in this paper were therefore recomputed from the stored answer text with a single fixed rule, given in Appendix C.

Artefact 4: a name written two ways. The first fixed rule we wrote was strict. It looked for the brand's label, the part of the saved domain in front of the public suffix, as a whole word. For a brand whose domain is its name run together, acmewidgets.com, that rule finds "AcmeWidgets" and "acmewidgets.com" and does not find "Acme Widgets". We knew this, judged the undercount to be constant over time and therefore harmless to comparisons, and proceeded. It was not harm-

less, because how an engine writes a name depends on whether it searched. With retrieved pages in its context, an engine frequently reproduces the run-together form that appears in the domain and on the company's own site. Without them, it writes the name as a person would say it, in separate capitalised words. The strict rule therefore detected mentions far more reliably in sourced answers than in unsourced ones, and every contrast between the two was biased towards finding that search increases mentions. On branded prompts the bias was large enough to manufacture a 15-point effect out of nothing (Section 7.4).

The final rule accepts the letters of the label in order with up to two separator characters (space, hyphen, apostrophe, full stop) between any two of them, and, when separators are present, requires the match to be capitalised as a name: the first letter upper case and at least one later word capitalised. The capitalisation requirement keeps out ordinary phrases that happen to spell a label; one account's label is a common two-word phrase, and without the requirement a hundred generic uses of the phrase were counted as mentions. For the prompt, which people type in lower case, the capitalisation requirement is dropped. Under the final rule 9,011 of 95,555 scorable answers name the brand, against 6,153 under the strict rule: the strict rule missed 32% of mentions. On unbranded prompts the final rule roughly doubles to quadruples each engine's mention rate, and on branded prompts it raises the rate from between 48% and 82% to between 79% and 97%.

The final rule agrees with the instrument's stored indicator on 98.1% of answers. It finds 1,454 mentions the instrument did not record, most of them from before October when the instrument did not match names written as words, and misses 323 that the instrument recorded. It still cannot recognise a brand whose spoken name differs from its domain, such as a company trading under one name on a domain that spells another, or a name in which "and" is written as an ampersand. Absolute mention rates remain lower bounds.

The lesson of the fourth artefact is the sharpest of the four. A measurement error that is constant across time can still differ across the conditions being compared, when the conditions change the form of the thing being measured. Any study that detects brand mentions by string matching and compares grounded with ungrounded answers is exposed to it.

Artefact 5: two instruments in one table. The onboarding audits described in Section 5.3 were found last, while checking why half the accounts in the panel had never received a sourced answer from any engine. They are not scans. They use clients that predate the search-enabled ones, send a handful of prompts, and return short answers: ChatGPT's median audit answer is 109 words against 520 in a full scan. Left in, they appeared as 5,680 answers in which engines "chose not to search", including 921 from Perplexity, which always searches. They lowered every engine's apparent search rate, by three points for ChatGPT and eight for Claude, filled the "without sources" side of every comparison with answers from a different configuration, and added 137 accounts that contributed nothing but audit data. Removing them changed no conclusion's direction. It changed most of the paper's sample sizes, sharpened the model of the decision to search, and reduced the number of accounts on which everything rests from 253 to 116, of which 112 can be scored for mentions. That is the honest figure.

The audits could be identified only because one engine in the panel always searches. A run in which Perplexity returns no sources is not a scan. Without such a sentinel, two code paths writing to one table would have been indistinguishable.

5.7 Sources and their fields

Each stored source has a URL, a title where the engine supplied one, and a domain derived from the URL (or, for Gemini, from the title, since Gemini's URLs are redirects). For Claude, Gemini and AI Overview from 30 September 2026 onward, each source also carries the character spans of the answer it supports, if any. Sites are compared at the level of the registrable domain: the label in front of the public suffix together with that suffix, so that pages on a subdomain are attributed to the parent site.

Among the full-scan answers that are not fragments, scorable or not, 50,765 carry at least one source, and those answers carry 476,973 source records pointing at 245,267 distinct URLs on 61,222 sites. A source is counted as **own** when its registrable domain equals that of the account's saved domain.

For Claude in the span era, 3,760 answers carry 45,045 source records, of which 19,106 have at least one span (3,759 answers, 45,035 records and 19,105 with a span once one fragment is removed). As explained in Section 2.7, the records without a span are search results that were placed in the context and not cited. This subset is the basis of Study 4.

5.8 Fetched pages

After each scan the instrument queues the most-cited URLs of that scan for fetching, up to a fixed number per scan. A worker retrieves each page with a browser-like user agent, converts it to text, and computes a set of structural features; a language model classifies the page's format and intent. Of 16,101 URLs queued by 15 November, 13,694 were fetched, 2,109 (13.1%) were blocked by the publisher's bot protection, and 298 failed for other reasons.

The features are counts and flags derived from the page's text and markup: word count, counts of second- and third-level headings, lists, tables and paragraphs, a count of numeric statistics, a count of "quotable" paragraphs (short, self-contained paragraphs that make sense out of context), the number of headings phrased as questions, outbound link count, the schema.org types declared, and flags for an FAQ block, a summary or "TL;DR" box, an author byline, a definition-style opening sentence and a filler introduction. Appendix B defines each. The format classes are guide, listicle, landing page, comparison table, review, news and a small residual.

Two selection effects apply. Pages are queued because they were cited often within a scan, so the fetched set over-represents frequently cited pages; and pages behind aggressive bot protection are missing. For Study 4, where fetched pages are joined to Claude's retrieved results, a page was available for 38.9% of cited results and 27.9% of results not cited. The difference is expected, since being cited is what gets a page queued, and it means the page-feature model is estimated on

a subset tilted towards pages cited elsewhere in the panel. The model conditions on the answer, so this tilt would bias the estimates only if the association between a feature and citation differed between pages that are and are not popular enough to have been fetched.

5.9 The category prompt sets

The category data comprise six categories: customer relationship management software, project management software, human resources and payroll software, applicant tracking systems, digital marketing agencies and search optimisation agencies. Four are products people license and two are services people hire. Each has a frozen set of 39 or 40 prompts, 238 in all, written the way a buyer types them: "mid market crm pricing", "crm for nonprofits", "best hr and payroll software", "cheap ats that doesn't suck". Each set was sent to Claude, ChatGPT, Gemini, Perplexity, Google AI Overview and Grok in August, September and October 2026, in 19 runs, giving 4,358 completed answers, of which 4,284 have at least 200 characters and are analysed. The extraction step found 12,084 brand mentions in them.

The engines were called through the same clients as in the panel, with one difference that is central to Study 2: by August, Grok had no search tool. All 750 of Grok's category answers are from parameters alone. For the other engines the share of answers with sources was 100% for Perplexity, 99% for AI Overview, 95% for Gemini, 55% for Claude and 41% for ChatGPT.

For each category the **brand universe** is the union of the roster of brands curated for that category's leaderboard and every brand named in any answer to that category's prompts, between 89 and 487 brands depending on the category. Crossing answers with their category's universe gives 1,060,729 answer and brand pairs, in each of which the brand either was or was not named. This long table is the data for Study 2.

5.10 Search queries

Shortly before the data were drawn, the instrument began storing the search queries that ChatGPT, Gemini and Claude report issuing, where the API exposes them. The stored queries all come from one scan of one account on 14 November 2026: 1,241 queries attached to 384 answers. That is enough to describe how each engine rewrites a prompt into searches and far too narrow to say anything about what the queries contain. It is used descriptively in Section 11.2.

5.11 Ethics, privacy and disclosure

The data are responses of commercial AI systems to prompts about businesses and markets. They contain no personal data about the people who wrote the prompts beyond the prompts themselves. The paper reports aggregates only: no customer, no customer's brand and no customer's prompt is identified, and example prompts are taken from the category sets, which belong to no

customer. Third-party sites that the engines cite are named, since which public sites AI engines cite is the subject of the study. Tables of most-cited sites are restricted to sites cited in the answers of at least ten different accounts, so that no table reveals the market of an individual customer.

The author is the founder of the company that operates the instrument and has a commercial interest in the subject. No AI vendor funded, reviewed or was consulted on the study. The raw answers cannot be released, because they would identify customers. The analysis code and every aggregate reported here are kept with the source of this page and are available on request, as described in Appendix D.

6. Statistical methods

This section defines the outcomes, the estimators and the inferential procedures. The guiding constraint is the dependence structure described in Section 5.5: answers are clustered in prompts and prompts in accounts, the clusters are few in some analyses and very unequal in all of them, and the main outcome is rare. A method that ignores any of these would produce intervals that are too narrow, and most of the choices below were made to avoid that.

6.1 Outcomes

For answer i to prompt $p(i)$ from engine $e(i)$ on behalf of account $a(i)$:

- $S_i \in \{0,1\}$: the answer returned at least one source.
- $M_i \in \{0,1\}$: the account's brand is named in the answer, under the fixed rule of Appendix C.
- $O_i \in \{0,1\}$: a source with the account's registrable domain is among the answer's sources.
- B_i: the prompt contains the account's brand label (branded).
- T_i: tokens billed; W_i: words in the answer; K_i: number of sources.
- $\mathcal{D}_i$: the set of registrable domains among the answer's sources.

In the category data, for answer i and brand b in the category's universe, $M_{ib} \in \{0,1\}$ indicates that b is named, O_{ib} that a source with b's registrable domain is among the sources, and L_{ib} that b's canonical name appears as a whole word in the title of at least one source.

For Claude's retrieved results, for answer i and result r at position $\text{pos}_{ir} \in \{0,1,2,\ldots\}$ in the order returned, $C_{ir} \in \{0,1\}$ indicates that the result carries at least one span, that is, was cited.

6.2 Proportions and their intervals

A rate such as the share of an engine's answers with sources is estimated by the sample proportion $\hat{\pi} = k/n$. Two intervals are reported for the main rates.

The **Wilson score interval** [46] treats the n answers as independent:

$$\frac{\hat\pi + \dfrac{z^2}{2n} \;\pm\; z\sqrt{\dfrac{\hat\pi(1-\hat\pi)}{n} + \dfrac{z^2}{4n^2}}}{1 + \dfrac{z^2}{n}}, \qquad z = 1.96. \tag{5}$$

It behaves well for proportions near zero, where the textbook interval $\hat\pi \pm z\sqrt{\hat\pi(1-\hat\pi)/n}$ can extend below zero, and it is the right interval for a question about these particular answers. It is not the right interval for a statement about answer engines in general, because the answers are not independent.

The **cluster bootstrap interval** [47, 48] respects the dependence. Let the data consist of G clusters (accounts, unless stated otherwise), cluster g having n_g answers of which k_g are successes. One bootstrap replicate draws G clusters with replacement, $g_1^*, \dots, g_G^*$, and computes

$$\hat\pi^* = \frac{\sum_{j=1}^{G} k_{g_j^*}}{\sum_{j=1}^{G} n_{g_j^*}}. \tag{6}$$

The interval is the 2.5th to 97.5th percentile of $\hat\pi^*$ over 2,000 replicates (1,000 where noted). Because whole accounts are resampled, the interval reflects how much the estimate would move if a different set of accounts had been observed, which is the relevant uncertainty when a few accounts dominate. The same procedure gives intervals for any statistic that is a function of cell means, including a difference of two rates and a difference in differences: each cluster is reduced once to its sums and counts in each cell, and the statistic is recomputed from the resampled totals.

The gap between the two intervals is a direct reading of the clustering. For ChatGPT's share of answers with sources, the Wilson interval is 52.7% to 54.3% and the cluster interval 44.4% to 63.1%: the effective sample size is closer to a few dozen accounts than to fourteen thousand answers. Unless stated otherwise, the intervals quoted in the text are cluster intervals.

The ratio of the two variances is the **design effect**. For a cluster-sampled proportion with average cluster size $\bar n$ and intraclass correlation ρ,

$$\mathrm{deff} \;=\; \frac{\mathrm{Var}_{\text{clustered}}(\hat\pi)}{\mathrm{Var}_{\text{simple}}(\hat\pi)} \;\approx\; 1 + (\bar n - 1)\rho, \tag{7}$$

and the effective sample size is n/deff [66]. We use equation (7) in Section 11.4 to turn measured intraclass correlations into advice on how many prompts and how many readings a tracking programme needs.

6.3 Regression for binary outcomes with clustering

Three forms of logistic model are used, chosen by what the comparison needs to hold fixed.

Pooled logistic regression with cluster-robust errors. For an outcome Y_i and covariates x_i,

$$\operatorname{logit}\Pr(Y_i = 1 \mid x_i) = \log \frac{\Pr(Y_i = 1 \mid x_i)}{1 - \Pr(Y_i = 1 \mid x_i)} = \beta_0 + x_i^{\top}\beta. \tag{8}$$

Coefficients are estimated by maximum likelihood as if observations were independent, and the covariance matrix is replaced by the cluster-robust sandwich estimator

$$\widehat{\operatorname{Var}}(\hat\beta) = \hat A^{-1} \left(\sum_{g=1}^{G} \hat u_g \hat u_g^{\top} \right) \hat A^{-1}, \qquad \hat u_g = \sum_{i \in g} x_i \, (Y_i - \hat\pi_i), \tag{9}$$

where $\hat A$ is the observed information matrix and the sum runs over clusters [48]. The estimator is consistent as the number of clusters grows. With few clusters it understates the variance, and where an analysis rests on fewer than about forty clusters we say so and treat its p-values as approximate.

Generalised estimating equations. Where the question is about a population-averaged effect and the within-cluster correlation is of interest in its own right, we fit equation (8) by generalised estimating equations with an exchangeable working correlation [49], again with the robust covariance. This is used for the model of the decision to search (Section 7.2), with accounts as clusters, and for the count of brands per answer (Section 8.1), with a log link and prompts as clusters.

Conditional logistic regression. The pooled models compare across prompts and accounts and so are exposed to every difference between them. Where each cluster contains variation in both the outcome and the predictor of interest, a stronger design conditions on the cluster. Suppose the model has a cluster-specific intercept,

$$\operatorname{logit}\Pr(Y_{gi} = 1) = \alpha_g + x_{gi}^{\top}\beta. \tag{10}$$

The intercepts α_g absorb everything constant within cluster g: the prompt's wording, the brand, the market, the engine if clusters are defined within engine. They cannot be estimated consistently when clusters are small, but they need not be. Conditioning on the number of successes in each cluster, $t_g = \sum_i Y_{gi}$, removes them from the likelihood [50, 51]:

$$\mathcal{L}_c(\beta) = \prod_{g} \frac{\exp\!\left(\sum_i Y_{gi}\, x_{gi}^{\top}\beta\right)}{\displaystyle\sum_{d \in \mathcal{S}(n_g, t_g)} \exp\!\left(\sum_i d_i \, x_{gi}^{\top}\beta\right)}, \tag{11}$$

where $\mathcal{S}(n_g, t_g)$ is the set of all binary vectors of length n_g with exactly t_g ones. Only clusters in which the outcome varies contribute; a cluster in which every answer or no answer is a success has likelihood one whatever β is. We report the number of such **informative clusters** with every conditional estimate, because it, and not the number of answers, is the sample size that governs precision.

Conditional logistic regression is used in three places with three definitions of cluster. In Section 7.5 the cluster is a prompt within an engine and the predictor is S_i: among readings of the same prompt by the same engine, is the brand named more often on the readings that returned sources? In Section 9.6 the cluster is the same and the predictor is O_i. In Section 10 the cluster is a single Claude answer and the observations are its retrieved results: among the pages placed in the context of this answer, which were cited? In that last use equation (11) is McFadden's conditional logit model of choice among alternatives [50], generalised to the case where more than one alternative is chosen.

When every informative cluster points the same way, the conditional likelihood has no finite maximum, and no odds ratio can be reported. In that case we report the **sign test**: if u clusters have a higher success rate under the condition and v a lower one, then under the null hypothesis of no effect u is binomial with parameters $u+v$ and one half, and the two-sided p-value is the probability of a split at least as uneven as the one observed.

6.4 Difference in differences

The withdrawal of Grok's search tool is analysed as a two-group, two-period comparison. Let $G_i = 1$ for Grok and 0 for the four control engines (ChatGPT, Claude, Gemini, Perplexity), and let $P_i = 1$ for the period after withdrawal and 0 for the period before. With $\bar{M}_{gp}$ the mean of the mention indicator in group g and period p, the estimator is

$$\hat{\delta} = \left(\bar{M}_{11} - \bar{M}_{10}\right) - \left(\bar{M}_{01} - \bar{M}_{00}\right). \tag{12}$$

The first difference is the change in Grok's mention rate. The second is the change over the same calendar period in engines whose search configuration did not change, which estimates what would have happened to Grok's rate from everything else that changed: the mix of accounts and prompts scanned, real changes in what is written on the web about the tracked brands, and the passage of time. Under the **parallel trends** assumption, that absent the withdrawal Grok's rate would have moved by the same amount as the controls', $\hat{\delta}$ estimates the average effect of the withdrawal on Grok's mention rate.

The windows are 12 June to 9 August 2026 for the before period and 16 August to 14 November for the after period. The change was committed on 7 August and, because scans in flight and cached deployments continued to use the old client for some days, the share of Grok answers with sources fell from over 90% to zero between 8 and 16 August; that week is excluded. The control engines did change in other ways during the windows, ChatGPT moving from gpt-5 to gpt-5-mini on 15 July, and that is a limitation of the control group to which we return in Section 13.

Two versions are reported. The cell-mean estimator of equation (12) is given with a cluster bootstrap interval, resampling accounts. The regression version is a linear probability model with prompt fixed effects,

$$M_i = \alpha_{p(i)} + \gamma\, G_i + \lambda\, P_i + \delta\, G_i P_i + \varepsilon_i, \tag{13}$$

estimated by demeaning every variable within prompt and applying least squares, with errors clustered on account. The fixed effects α_p remove all differences between prompts, including the brand and the market, so δ is identified from prompts read in both periods and by both groups of engines. Clustering on account addresses the serial-correlation problem of difference-in-differences designs with many observations per unit [52]. A linear probability model is used here in place of a logit because the interaction coefficient in a linear model is the difference in differences in percentage points, the quantity of interest, and because fixed effects in a logit with a rare outcome would discard almost all the data.

The design has one treated unit, Grok. Inference about "the effect of removing search from an engine" in general would require several engines to have had search removed at different times. What the design supports is a statement about this engine, and the plausibility of generalising rests on the mechanism.

6.5 Paired and rank-based comparisons

Where the same prompt contributes observations under two conditions, the paired difference $\Delta_p = \bar{Y}_{p,1} - \bar{Y}_{p,0}$ is computed per prompt and its mean reported with a cluster bootstrap interval over accounts. For count outcomes paired by prompt, the Wilcoxon signed-rank test [61] is used. For comparisons of two unpaired distributions that are far from normal, token counts above all, the Mann-Whitney test [57] is used and the effect size reported is Cliff's δ [58],

$$\delta = \Pr(X > Y) - \Pr(X < Y) = \frac{2U}{n_X n_Y} - 1, \tag{14}$$

where U is the Mann-Whitney statistic. It ranges from -1 to 1 and equals 1 when every value in one group exceeds every value in the other. With samples of thousands, a p-value for a token difference is uninformative; δ says how completely the distributions separate.

6.6 Counts

The number of brands named in an answer is modelled by a log-linear model fitted by generalised estimating equations with a Poisson variance function,

$$\log \mathrm{E}[N_i \mid x_i] = \beta_0 + x_i^{\top}\beta, \tag{15}$$

so that $\exp(\beta_j)$ is a **rate ratio**. The robust covariance makes the inference valid whether or not the counts are Poisson-distributed.

The number of times a fetched page has been cited across the panel is heavily overdispersed, with a variance far above its mean, and is modelled by negative binomial regression [65]. Every page in the fetched set has been cited at least once, so the model is fitted to $N_i - 1$, the number of citations beyond the first, with

$$\mathrm{E}[N_i - 1 \mid x_i] = \mu_i = \exp(\beta_0 + x_i^{\top}\beta), \qquad \mathrm{Var}[N_i - 1 \mid x_i] = \mu_i + \alpha \mu_i^2. \tag{16}$$

The dispersion parameter α is estimated with the coefficients; $\exp(\beta_j)$ is an **incidence rate ratio**. Errors are clustered on site. The log of the number of weeks between a page's first and last observed citation enters as a covariate, to adjust for how long the page has been at risk of being cited again.

6.7 Concentration

Four summaries of how unevenly citations (or mentions) are spread over sites (or brands) are reported. Let $s_1 \geq s_2 \geq \cdots \geq s_n$ be the shares, summing to one.

The **top-k share** is $\sum_{j \leq k} s_j$. The **Gini coefficient** is

$$\mathrm{Gini} = \frac{\sum_{j=1}^{n} (2j - n - 1)\, s_{(j)}}{n \sum_{j} s_{(j)}}, \tag{17}$$

with $s_{(j)}$ the shares in ascending order; it is zero when all shares are equal and approaches one when a single item takes everything. The **effective number** is the exponential of the Shannon entropy,

$$N_{\mathrm{eff}} = \exp\left(- \sum_{j=1}^{n} s_j \log s_j \right), \tag{18}$$

the Hill number of order one [59]: the number of equally cited sites that would give the same entropy. An engine whose citations fall on 13,000 sites with an effective number of 3,500 is behaving, for the purposes of diversity, like an engine that cites 3,500 sites equally.

These three measures depend on sample size. A smaller sample of citations contains fewer distinct sites and looks more concentrated, so comparisons between engines with very different numbers of citations should be read with care. For the same reason we do not report bootstrap intervals for them. Resampling prompts with replacement leaves fewer distinct prompts in each replicate, which by itself raises measured concentration, so that a percentile interval sits above the point estimate it is meant to bracket. We observed exactly this in a first version of the analysis and removed the intervals.

6.8 Power-law fitting

The hypothesis that citation counts per site follow a power law in the tail is that, for counts $x \geq x_{\min}$,

$$\Pr(X = x) = \frac{x^{-\alpha}}{\zeta(\alpha, x_{\min})}, \tag{19}$$

where ζ is the Hurwitz zeta function, which normalises the discrete distribution. Following Clauset, Shalizi and Newman [54], the exponent is estimated by maximum likelihood, for which a close approximation in the discrete case is

$$\hat{\alpha} \simeq 1 + n_{\mathrm{tail}} \left[\sum_{i=1}^{n_{\mathrm{tail}}} \log \frac{x_i}{x_{\min} - \tfrac{1}{2}} \right]^{-1}, \tag{20}$$

and the lower cutoff $x_{\min}$ is chosen to minimise the Kolmogorov-Smirnov distance between the fitted and empirical distributions above the cutoff. Fits were computed with the `powerlaw` package [55], which maximises the exact discrete likelihood.

A good fit to a power law does not show that the data are power-law distributed, because other heavy-tailed distributions fit similar data. We therefore compare the power law with a log-normal by the normalised log-likelihood ratio R of Clauset et al., which is positive when the power law is favoured and negative when the log-normal is, with a p-value for whether the sign is distinguishable from zero. We do not run their bootstrap goodness-of-fit test, which at these sample sizes rejects any parametric form. The substantive claim we make is the weaker and more useful one: the distribution is heavy-tailed, with a tail exponent near 2.

For a continuous power law with exponent α, the mean is finite only for $\alpha > 2$ and the variance only for $\alpha > 3$. An exponent near 2 therefore means that sample averages of citations per site are dominated by the few largest sites and are unstable, which is one more reason the paper reports shares and medians and not means for site-level quantities.

6.9 Set similarity

The agreement between two sets of cited sites, or of named brands, is measured by the Jaccard index [60],

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}, \tag{21}$$

which is zero when the sets are disjoint and one when they are identical. It is computed between two engines' answers to the same prompt in the same scan (Section 9.4), between consecutive readings of the same prompt by the same engine (Section 9.5), and between the brand sets named for the same category prompt in different months (Section 8.5). Jaccard similarity is sensitive to set size: two sets of ten sites that share five have $J = 1/3$. We report the mean index, the share of pairs with no element in common, and, for readings over time, the Spearman rank correlation between the index and the number of days separating the readings.

6.10 Reliability

Test-retest reliability of the mention indicator is assessed three ways. For consecutive readings (M_1, M_2) of the same prompt by the same engine:

The **conditional probabilities** $\Pr(M_2 = 1 \mid M_1 = 1)$ and $\Pr(M_2 = 1 \mid M_1 = 0)$ are the most directly interpretable: given that the brand was (or was not) named this time, how likely is it to be named next time?

Cohen's κ [56] corrects raw agreement for the agreement expected by chance:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \qquad p_e = \pi_1 \pi_2 + (1 - \pi_1)(1 - \pi_2), \tag{22}$$

where p_o is the share of pairs in which the two readings agree and π_1, π_2 are the two marginal mention rates. With a rare outcome raw agreement is uninformative, since two readings that both say "not named" 99% of the time agree 98% of the time by chance alone, and κ is the appropriate summary.

The **intraclass correlation** of the indicator within prompt and engine is estimated by the one-way analysis-of-variance estimator for unequal group sizes,

$$\hat{\rho} = \frac{\mathrm{MSB} - \mathrm{MSW}}{\mathrm{MSB} + (k_0 - 1)\,\mathrm{MSW}}, \qquad k_0 = \frac{1}{G - 1}\left(n - \frac{\sum_g n_g^2}{n} \right), \tag{23}$$

with MSB and MSW the between-prompt and within-prompt mean squares. It is the share of the variance in the indicator that lies between prompts. A value near one means a prompt either names the brand or does not, and repeated readings add little; a value near zero means that knowing the prompt tells you almost nothing about a single reading.

To translate these into the precision of a tracked mention rate, Section 11.4 uses a Monte Carlo calculation in place of equation (7), because with rates near half a percent the normal approximation is poor. From the pool of prompt-and-engine pairs with at least five readings, each with its observed mention rate π_p, we draw m prompts with replacement, simulate r readings of each as binomial draws with probability π_p, compute the pooled rate, and repeat 2,000 times. The standard deviation of the simulated rates is the sampling error of a tracking set of m prompts read r times.

6.11 Multiple comparisons

Where a model contains many terms tested together, as in the page-feature models of Sections 10 and 11.1, p-values are adjusted by the Benjamini-Hochberg procedure [53]. With m p-values ordered $p_{(1)} \le \cdots \le p_{(m)}$, the adjusted value is

$$\tilde{p}_{(j)} = \min_{l \geq j} \left\{ \min\left(1, \frac{m}{l}\, p_{(l)}\right) \right\}, \tag{24}$$

and rejecting all hypotheses with $\tilde{p} \leq 0.05$ controls the expected share of false discoveries among the rejected at 5%. Across the paper as a whole no formal correction is applied. The studies address different questions with different data, and we rely instead on reporting every analysis that was run, including the ones that found nothing.

6.12 What is and is not claimed as causal

Three estimates in the paper are given a causal reading, each with a stated assumption. The difference in differences of Section 7.4 is the effect on Grok of withdrawing its search tool, under parallel trends. The within-answer position gradient of Section 10 is the effect of position on citation only under the assumption that position is unrelated to the page's inherent citability given the other covariates, which is false, since the search service ranks by relevance; we therefore describe it as the joint effect of rank and relevance and do not separate them. The token differences of Section 2.5 are mechanical and need no assumption.

Every other estimate is an association. In particular the coefficients on own-site citation in Sections 8.3 and 9.6 do not say what would happen to a brand if its site were made to appear among the sources. A page from a brand's site is retrieved because the search service judged it relevant to the model's query, and the conditions under which that happens are the conditions under which the brand is pertinent to the answer in the first place. The within-prompt version of the estimate removes differences between prompts and brands, and it cannot remove this.

6.13 Software

Analyses were run in Python 3.13 with pandas, NumPy and SciPy, with statsmodels [62] for regression models and the `powerlaw` package [55] for tail fits. Random number generators were seeded, so the bootstrap and simulation results reproduce exactly from the same data.

7. Study 1: When engines search and what searching changes

Study 1 uses the answer panel to address RQ1 and the first half of RQ2. It describes how often each engine grounds its answer in sources, models the decision to search for the three engines that make one, and then estimates what a search does to the probability that an account's brand is named, first from the withdrawal of Grok's search tool and then from prompts that the same engine answered both ways.

7.1 How often answers carry sources

Table 6 gives the basic rates over the 68,636 answers of the analysis set. Figure 2 shows the first column with its cluster interval.

Table 6. Sources and mentions by engine, source era. Intervals are 95% cluster bootstrap intervals over accounts. "Unbranded" and "branded" refer to whether the prompt contains the account's brand label.

Engine	Answers	With sources	Sources per sourced answer (median)	Words (median)	Unbranded: brand named	Branded: brand named
ChatGPT	14,318	53.5% (44.4 to 63.1)	6	522	1.71% (0.78 to 3.59)	92.7%
Claude	8,465	80.5% (71.0 to 90.0)	10	235	4.16% (1.75 to 9.24)	92.6%
Gemini	10,046	92.3% (91.2 to 93.4)	6	207	2.53% (1.07 to 5.00)	91.0%
Perplexity	14,289	99.98%	10	214	3.09% (1.47 to 5.67)	92.8%
Google AI Overview	6,510	97.4% (96.1 to 98.9)	7	231	3.01% (1.04 to 9.65)	78.6%
Grok	14,200	35.3% (24.6 to 48.9)	5	171	1.15% (0.64 to 1.94)	97.1%
Copilot	808	24.3% (13.3 to 30.5)	5	408	0.65% (0.13 to 1.08)	95.0%

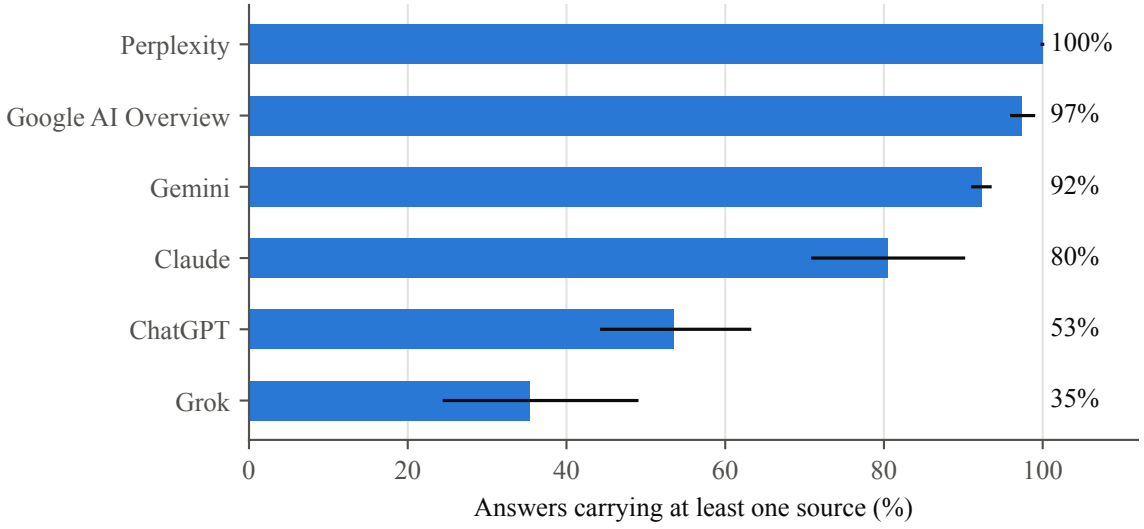


Figure 2. Share of answers that returned at least one source, by engine, 12 June to 14 November 2026. Bars are point estimates; lines are 95% cluster bootstrap intervals over accounts. Grok's figure averages a period with a search tool (93% of answers sourced) and a longer period without (0%).

Four observations.

The engines divide as their designs predict. The two search products return sources almost always. The three engines that choose return them between half and nine tenths of the time. Grok's 35% is an average over two regimes and is not a property of the model: while the tool was sent, 93% of its answers had sources, and afterwards none did.

Gemini's figure is inflated by the fragment filter. As Section 5.6 explained, removing answers under 40 words removed mostly unsourced Gemini reads. Before the filter, about 64% of Gemini's source-era answers had sources. The true rate at which Gemini grounds a complete answer lies between the two and cannot be recovered from these data.

ChatGPT writes at more than twice the length of the other current engines, a median of 522 words against 171 to 235. (Copilot, retired in July, wrote 408.) It was called with no system instruction and a generous output limit, where Claude, Gemini, Perplexity and Grok were asked to answer concisely. Length matters for what follows, because a longer answer has room to name more brands.

Unbranded mention rates are low, and the engines that search most name small brands most. The rates run from 0.7% for Copilot and 1.2% for Grok, the two engines that mostly or entirely answered from parameters, through 1.7% for ChatGPT, which searched half the time, to between 2.5% and 4.2% for the four engines that searched most. The cluster intervals are wide and overlap, so this ordering is suggestive and no more; Sections 7.4 and 7.5 test it properly. On branded prompts every engine but one names the brand in more than nine answers out of ten. The exception is AI Overview, at 79%, which summarises search results for the query as typed and often writes about the topic without repeating the name it was given.

The number of sources attached to a sourced answer is modestly related to its length for most engines (Spearman correlation between source count and word count: 0.55 for Gemini, 0.37 for Claude, 0.32 for Grok, 0.26 for ChatGPT, 0.25 for AI Overview) and unrelated for Perplexity (-0.02), which returns its search result list at a fixed size regardless of what it writes.

7.2 What predicts the decision to search

For ChatGPT, Claude and Gemini the decision to search is the model's. Table 7 reports, for each, a logistic model of whether the answer returned sources on the category of the prompt and seven lexical features of its text, fitted by generalised estimating equations with accounts as clusters. Odds ratios are relative to an awareness prompt without the feature.

Table 7. Odds ratios for an answer returning sources. GEE logistic regression, exchangeable working correlation, robust errors clustered on account; 95% intervals in parentheses. Bold: interval excludes one.

Predictor	ChatGPT (n = 14,318; 112 accounts)	Claude (n = 8,465; 58)	Gemini (n = 10,046; 109)
Category: comparison	**5.22 (3.52 to 7.73)**	**5.01 (2.56 to 9.80)**	0.70 (0.47 to 1.03)
Category: recommendation	**4.15 (3.19 to 5.40)**	1.33 (0.60 to 2.93)	1.03 (0.67 to 1.58)
Category: use case	**2.87 (2.05 to 4.01)**	1.45 (0.79 to 2.65)	0.96 (0.67 to 1.37)
Category: industry	1.27 (0.63 to 2.57)	1.12 (0.54 to 2.32)	**0.67 (0.46 to 0.97)**
Prompt names the brand	1.55 (0.75 to 3.21)	0.90 (0.22 to 3.64)	**0.25 (0.17 to 0.37)**
Contains a year	1.81 (0.81 to 4.01)	3.63 (0.12 to 108)	0.54 (0.27 to 1.06)
Contains "best", "top" or "leading"	**2.67 (1.89 to 3.78)**	**1.82 (1.50 to 2.21)**	**0.61 (0.45 to 0.84)**
Contains a comparison term	0.79 (0.61 to 1.02)	1.10 (0.44 to 2.76)	1.22 (0.88 to 1.70)
Phrased as a question	0.75 (0.53 to 1.06)	**0.40 (0.17 to 0.94)**	**1.49 (1.21 to 1.83)**
Contains a price term	**2.14 (1.40 to 3.28)**	**4.06 (2.23 to 7.40)**	1.17 (0.79 to 1.73)
Contains a locality term	**0.63 (0.41 to 0.98)**	**0.33 (0.14 to 0.80)**	0.85 (0.57 to 1.26)
Log prompt length in words	1.07 (0.77 to 1.49)	**2.23 (1.13 to 4.41)**	**0.53 (0.41 to 0.67)**
Base rate	53.5%	80.5%	92.3%

H1a is supported for ChatGPT and Claude, and not for Gemini.

ChatGPT's search behaviour is strongly structured by what is being asked. An awareness prompt, of the "what is" or "how does" kind, is the reference. Against it, a comparison prompt has 5.2 times the odds of triggering a search, a recommendation prompt 4.2 times and a use-case prompt 2.9 times. Over and above the category, the presence of "best" or "top" multiplies the odds by 2.7 and a price term by 2.1. These are exactly the features that mark a question whose answer goes out of date. The model answers "what is an applicant tracking system" from memory and looks up "best applicant tracking system pricing". This is consistent with a model that has learned, as Self-RAG was explicitly trained to [9] and as adaptive retrieval proposes [16], to retrieve when its parameters are least trustworthy.

Claude shows the same pattern on the features that matter most: comparison prompts (5.0), price terms (4.1) and superlatives (1.8) raise the odds of a search. It is also more likely to search for longer prompts and less likely to search for prompts phrased as questions or containing a locality term. Claude's estimates rest on 58 accounts and several have wide intervals; the year term, with an interval from 0.12 to 108, is not estimated in any useful sense.

Gemini searched for 92% of the prompts it answered completely, and where its coefficients differ from one they point the other way: superlatives, a named brand and longer prompts all lower the odds of a search. With a base rate that high, in a sample from which more than a quarter of its reads were removed as fragments, we do not interpret these.

Two features did not behave as expected. A comparison term in the text ("vs", "alternatives") does not raise ChatGPT's odds of searching once the comparison category is accounted for, which probably reflects collinearity between the two. And locality terms lower the odds for both ChatGPT

(0.63) and Claude (0.33), which is surprising for a query type usually regarded as needing fresh local data, and may reflect how few such prompts there are in a panel dominated by software.

The practical reading of Table 7 is that whether retrieval is in play at all is prompt-dependent in a predictable way, for the largest engine most of all. For informational questions ChatGPT answers mostly from its parameters, and no page published today can affect that answer until a future model is trained on it. For commercial comparison and recommendation questions it mostly searches, and what is retrievable today matters.

7.3 What a search adds to the context

The token evidence was presented in Section 2.5 with Table 1 and is shown in Figure 3. To restate it against hypothesis H1b: answers that returned sources were billed a median of 4,362 more tokens than answers that did not on ChatGPT, 8,774 more on Claude and 10,966 more on Grok, with almost no overlap between the two distributions on any of the three (Cliff's δ of 0.98, 1.00 and 1.00). H1b is supported.

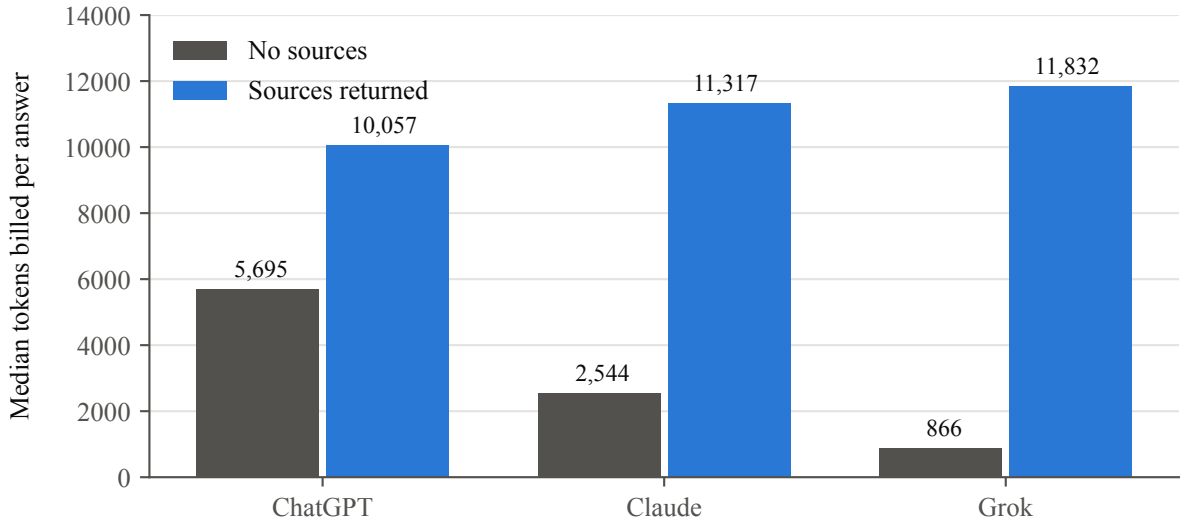


Figure 3. Median tokens billed per answer, by whether the answer returned sources, for the three engines whose token counts include retrieved content. The user's prompt has a median length of ten words.

The spread of the sourced distribution is itself informative. Claude's interquartile range runs from 10,786 to 14,914 tokens and Grok's from 8,354 to 18,853, reflecting answers that searched more than once. ChatGPT, capped at one search, has a tight range of 9,779 to 10,349. The cap on searches is therefore a cap on how much of the web can reach the model for a given question, and an engine allowed five searches can have twice the retrieved text in its context as one allowed a single search. Consumer products typically allow more than either.

7.4 The withdrawal of Grok's search tool

On 7 August 2026 the operator of the instrument stopped including a search tool in requests to Grok, because the tool's per-call charges had become the majority of that vendor's bill. No other engine's configuration was changed that week. From 16 August every Grok answer was generated from parameters alone. Table 8 gives the four cell means of the difference-in-differences design for unbranded prompts, and Figure 4 plots them.

Table 8. Unbranded prompts: share of answers naming the account's brand, before and after Grok's search tool was withdrawn. Controls are ChatGPT, Claude, Gemini and Perplexity pooled. 56,770 answers, 6,899 prompts, 110 accounts.

	Before (12 Jun to 9 Aug)	After (16 Aug to 14 Nov)	Change
Grok: answers	4,839	8,105	
Share with sources	93.1%	0.0%	
Brand named	1.90%	0.70%	−1.20 pp
Mean answer length (words)	249	134	−115
Controls: answers	14,366	29,460	
Share with sources	75.5%	82.8%	
Brand named	1.18%	3.53%	+2.35 pp
Mean answer length (words)	334	317	−17
Difference in differences, brand named			**−3.55 pp**

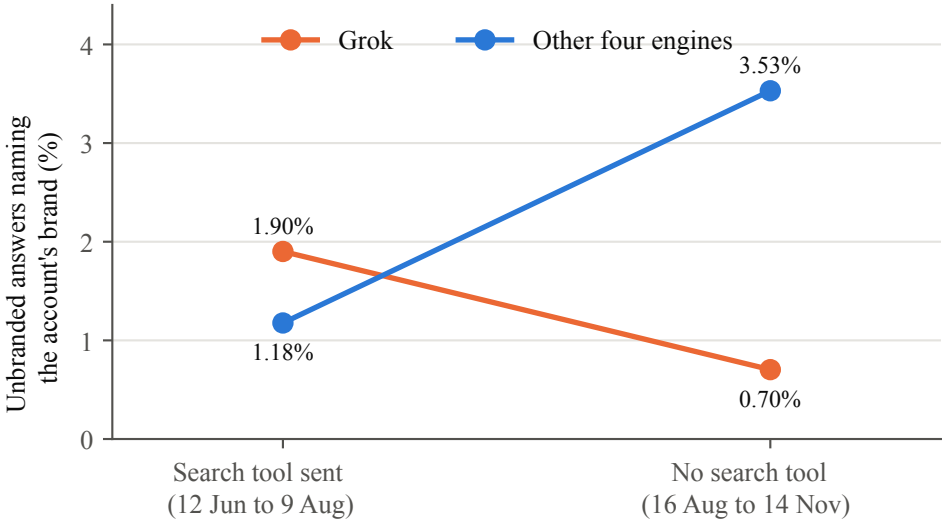


Figure 4. Share of answers to unbranded prompts that named the account's brand, for Grok and for the four engines whose search configuration did not change, before and after Grok's search tool was withdrawn.

While it could search, Grok named the tracked brand in 1.90% of its answers to unbranded prompts, well above the 1.18% of the control engines in the same weeks. After the tool was withdrawn its rate fell to 0.70%, a relative fall of 63%. Over the same interval the control engines' rate

rose threefold, to 3.53%. Before the change Grok was the engine most likely to name a small brand unprompted; afterwards it was the least. The difference in differences is **−3.55 percentage points**, with a cluster bootstrap interval over accounts of −7.11 to −1.52. The linear probability model with prompt fixed effects, equation (13), gives −3.39 points (standard error 1.20, p = 0.005, interval −5.75 to −1.03). H2a is supported.

The estimate has two parts of very different reliability, and they should be kept apart. Grok's own fall of 1.20 points is a before-and-after change in one engine. The controls' rise of 2.35 points is larger than the thing being measured, and it needs an explanation before it can be used as a counterfactual.

Why the controls rose. Each scan sends the same prompts to every engine on an account's plan, so within either period Grok and the controls were asked almost the same questions on behalf of the same accounts. That is the design's strength: the five engines are five instruments applied to one population, and one of them was altered. But the population itself changed between periods. Of the 41 accounts scanned with unbranded prompts in the before period, only three were scanned again in the after period; the other 69 accounts of the after period were new. For the three continuing accounts the controls' rate in the after period was 1.17%, the same as the 1.18% of the before period. For the 69 new accounts it was 4.80%. The controls' rise is therefore entirely a change in who was being tracked: the later accounts' brands were ones that engines with search named four times as often. On those same new accounts' prompts, Grok without search named the brand 1.06% of the time. The fixed-effects estimate rests on exactly this comparison. Within each prompt it takes the difference between Grok's mention rate and the other engines' mention rate on that prompt, and asks how that difference changed between prompts read before and prompts read after. Averaged over prompts, the within-prompt difference was +0.92 points in Grok's favour before the withdrawal and −3.81 points after.

One further caution about the after period: mentions there are concentrated. A single account supplies 34% of the control engines' unbranded mentions in that period and three accounts supply 59%. The cluster bootstrap interval, which resamples accounts, is wide for that reason, and it is the interval to rely on.

What if only Grok's own change is trusted. A reader who distrusts the controls entirely can take the 1.20-point fall as the estimate. It assumes nothing else changed for Grok, which the changing customer mix makes doubtful in the direction of *understating* the effect: the later mix was more nameable for every searching engine, and Grok's rate fell regardless.

Accounts present in both periods. Among the three accounts scanned in both windows, where the customer mix is fixed, the estimate is −1.72 points (cluster interval −2.03 to −0.99). In that subsample Grok named the brand in 1.06% of answers with search and in none of 2,742 answers without.

The two largest accounts. Removing the author's two accounts leaves 39,529 answers from 108 accounts. The estimate is −4.42 points (interval −9.37 to −1.53), and −4.17 with prompt fixed effects (p = 0.015). The effect is not produced by the author's brands.

The matching rule. Under the strict matching rule, which requires the brand's label written exactly as in its domain, the estimate is −1.35 points (interval −2.10 to −0.71). It is smaller because the strict rule misses most mentions of brands whose names are written as several words, and it has the same sign and is clearly separated from zero. Section 5.6 explains why the looser rule is the correct one.

Answer length. Grok's answers became much shorter without search, falling from 249 to 134 words on average (difference in differences −98 words, interval −117 to −80). A shorter answer names fewer things. Part of the effect of search on mentions therefore runs through length: retrieved text gives the model more to say. This is a mechanism, not a confound. But it does mean that the effect could in principle be reproduced, in part, by simply asking for longer answers, and we cannot separate the two paths with this design.

Branded prompts. H2c predicted little effect of search when the prompt already names the brand. On branded prompts Grok named the brand in 96.9% of answers while it could search and 97.3% afterwards. The controls moved from 89.5% to 93.8%. The difference in differences is −3.9 percentage points with an interval that includes zero (−12.1 to +2.1; 4,157 answers, 94 accounts), and −5.3 with prompt fixed effects ($p = 0.14$). H2c is supported: a name supplied in the prompt is repeated in the answer whether or not the engine searches.

This result deserves a note on its history, because the first version of the analysis reached the opposite conclusion with apparent confidence. Under the strict matching rule, Grok's branded mention rate appeared to fall from 97% to 81% when search was withdrawn, a difference in differences of about −15 points. Reading the answers showed why. Asked about a small company by its domain name with no search results in front of it, Grok writes the company's name the way a person would say it, as separate capitalised words. With search results in the context, the run-together form used in the domain and on the company's own pages is in front of the model and it often writes that. The strict rule recognised only the second form. The apparent effect of search on branded mentions was an effect of search on *how the name was spelled*. The correction is described as the fourth artefact in Section 5.6, and its discovery changed every mention figure in the paper.

7.5 The same prompt, with and without sources

The engines that choose whether to search sometimes answered the same prompt both ways on different days. Among such prompts the brand, the wording and the engine are fixed, and the contrast between sourced and unsourced readings is free of every difference between prompts. Table 9 reports the comparison.

Table 9. Prompts read by the same engine both with and without sources. "Prompts for / against" counts prompts whose mention rate was higher / lower on sourced readings; prompts with equal rates are not informative.

Engine	Prompts	Answers	Accounts	Named, with sources	Named, without	Prompts for / against	Sign test p
Unbranded							
ChatGPT	556	2,700	24	0.66%	0.00%	10 / 0	0.002
Claude	241	1,308	10	0.55%	0.00%	2 / 0	0.50
Gemini	332	1,555	18	0.53%	0.24%	4 / 0	0.13
Branded							
ChatGPT	55	183	13	99.0%	86.6%	9 / 1	0.021
Claude	10	33	4	100%	41.7%	6 / 0	0.031
Gemini	21	47	8	80.8%	95.2%	0 / 3	0.25

For ChatGPT the within-prompt evidence is clear. Across 556 unbranded prompts that it answered both with and without sources, it never once named the account's brand in an unsourced reading, and it named it in 0.66% of sourced readings. Ten prompts favoured search and none went the other way, for a two-sided sign-test p of 0.002. The mean paired difference is 1.06 percentage points (cluster interval 0.36 to 2.19). Because every informative prompt points the same way, the conditional likelihood has no maximum and no odds ratio can be given; the correct statement is that, in this sample, ChatGPT named a small brand unprompted only when it had searched. H2b is supported for ChatGPT.

Claude and Gemini point the same way and are too thin to stand alone: every informative prompt, two for Claude and four for Gemini, favoured search. Pooling the three engines, sixteen prompts favoured search and none opposed it.

On branded prompts ChatGPT named the brand in 99.0% of sourced and 86.6% of unsourced readings (nine prompts to one, $p = 0.021$; conditional odds ratio 11.9, interval 1.4 to 98.9). This is a modest effect on a high base. It is more precisely located than the branded Grok result and points the same way: search does a little for a name already in the prompt, mostly by removing the occasional answer that discusses the category and never returns to the name. Claude shows a strong effect on branded prompts, 100% against 42%, on ten prompts from four accounts, which is too few to lean on. Gemini's three informative branded prompts go the other way.

Two cautions apply to all of Table 9. The first is power. Even with the corrected matcher, 98% of these prompts are never named under either condition and contribute nothing; the unbranded comparison rests on sixteen prompts. The second is selection. In the Grok design search was removed from outside. Here the model chose, reading by reading, whether to search, and the readings on which it chose not to are not a random half. If the model declines to search when it is already confident of an answer, the unsourced readings are the ones where its parameters had the most to offer, which biases the contrast *against* finding a benefit of search. The ChatGPT result survives that bias.

It is worth recording what this table looked like at each stage of correction, since each version would have supported a different paper. With truncated answers included and source chips counted as mentions, the analysis gave an odds ratio of 2.5 ($p = 0.004$) for ChatGPT, 0.41 ($p = 0.0002$) for Gemini and 0.16 ($p < 0.0001$) for Claude: three strongly significant effects in two opposite directions. With those two artefacts corrected and the strict matching rule, every effect vanished: ChatGPT's count was five prompts to none ($p = 0.06$) and the others were null. With the matching rule corrected as well, and the onboarding audits of Section 5.6 removed, ChatGPT's effect returned at ten prompts to none. The final version is the only one in which every engine's result has a mechanical explanation.

7.6 Summary of Study 1

Whether an engine searches is, for ChatGPT and Claude, a predictable function of the prompt: both search for comparisons, superlatives and prices, and answer informational questions from memory more often. A search adds between four and eleven thousand tokens of retrieved text to a context that otherwise contains a few dozen. Removing the ability to search cut the rate at which one engine named small brands unprompted by 63% while the engines that kept search named them three times as often as before, and shortened its answers by 46%. ChatGPT, across 556 prompts it answered both ways, named a small brand unprompted only on readings where it had searched. When the prompt already contains the brand's name, search makes little difference: the name is repeated nine times in ten or more regardless.

8. Study 2: Which brands are named

Study 1 followed one brand per account and asked whether it appeared. Study 2 turns the question round. On the category prompt sets every brand named in every answer was extracted, so for each answer we know the whole set of brands it named, and the unit of analysis becomes the pair of an answer and a brand. This supports the second half of RQ2, what search does to the set of brands named, and all of RQ3, how the model's prior and the retrieved text combine.

The data are 4,284 answers to 238 prompts in six categories, from six engines over three monthly runs. The outcome does not depend on string matching against a domain label: brands were identified by an extraction step that resolves each surface form to a canonical brand record, so the fourth artefact of Section 5.6 does not arise here.

8.1 How many brands an answer names

Table 10 gives the number of roster-resolved brands named per answer.

Table 10. Brands named per answer on the category prompt sets. Intervals are 95% cluster bootstrap intervals over prompts. Paired difference: mean, over prompts answered both ways by the same engine in different months, of brands named with sources minus brands named without.

Engine	Answers	With sources	Brands per answer, sourced	Brands per answer, unsourced	Naming no brand, sourced / unsourced	Paired difference (prompts)
ChatGPT	754	40.8%	5.50 (5.11 to 5.88)	2.57 (2.10 to 3.08)	5.5% / 55.8%	+0.75 (79)
Claude	754	54.9%	3.71 (3.33 to 4.09)	1.39 (0.99 to 1.82)	15.0% / 72.6%	+0.61 (64)
Perplexity	754	100%	3.05 (2.76 to 3.34)		28.8% /	
Google AI Overview	743	98.8%	2.31 (2.09 to 2.51)		27.9% /	
Gemini	529	94.7%	1.32 (1.10 to 1.55)	0.18 (0.04 to 0.34)	57.3% / 82.1%	+1.36 (21)
Grok	750	0%		3.43 (3.15 to 3.69)	/ 16.9%	

Three things stand out.

Sourced answers name more brands, in the raw comparison. ChatGPT names 5.5 brands in a sourced answer and 2.6 in an unsourced one; Claude 3.7 and 1.4. But the raw comparison mixes two things. The engines search for some prompts and not others (Section 7.2), and the prompts they do not search for are the definitional ones, to which the natural answer names nobody: 56% of ChatGPT's unsourced answers and 73% of Claude's name no brand at all, against 6% and 15% of their sourced answers.

Within prompt, the difference is much smaller. Restricting to prompts that the same engine answered with sources in one month and without in another, sourced answers named 0.75 more brands on ChatGPT (95% interval 0.03 to 1.48, 79 prompts; Wilcoxon $p = 0.13$), 0.61 more on Claude (-0.11 to 1.33, 64 prompts; $p = 0.049$) and 1.36 more on Gemini (0.47 to 2.25, 21 prompts; $p = 0.006$). The direction is consistent and the size is under one brand for the two engines with enough data.

Length accounts for it. A log-linear model of the brand count for ChatGPT, Claude and Gemini, with engine and category terms and prompts as clusters, gives a rate ratio for sourced against unsourced answers of 1.27 (95% interval 1.13 to 1.44, $p = 0.0001$). Adding the log of the answer's length reduces it to 1.03 (0.93 to 1.14, $p = 0.61$). Sourced answers are longer, and at a given length they name no more brands than unsourced ones. The first half of H2d is supported only in this qualified sense: search increases the number of brands named by increasing how much the model writes.

Grok is worth a separate remark. With no search at all it named 3.4 brands per answer, more than Perplexity, AI Overview or Gemini with search, and only 17% of its answers named no brand. An engine answering from parameters is not reticent about recommending. It simply recommends from a different list, which is the subject of the next section.

8.2 Search and the long tail

If retrieval puts names in the context that the parameters would not have produced, sourced answers should name brands that the non-searching engine does not. To measure this we define each brand's **prior** for its category as the share of Grok's answers to that category's prompts in which Grok named the brand. Grok never searched on these runs, so its naming reflects parameters alone. It is one model's parameters, not a universal prior, and Section 8.3 shows it nonetheless predicts every other engine's behaviour strongly.

Table 11 reports, for each engine and condition, how concentrated naming is and what share of brand mentions went to brands with a prior of zero, that is, brands Grok never named for the category in any of its roughly 125 answers.

Table 11. Concentration of brand mentions, averaged over the six categories, and the share of mentions going to brands the non-searching engine never names. Interval on the last column is a 95% cluster bootstrap interval over prompts; concentration measures are point estimates (Section 6.7).

Engine	Sources	Answers	Distinct brands per category	Top-5 share	Effective number	Share to never-named brands
Grok	No	750	26.3	62.1%	14.8	2.6% (1.5 to 3.9)
ChatGPT	No	446	20.2	62.8%	13.5	3.1% (1.8 to 4.6)
Claude	No	340	12.5	70.7%	9.2	1.5% (0.0 to 3.5)
ChatGPT	Yes	308	22.7	56.3%	14.5	5.7% (3.8 to 8.0)
Google AI Overview	Yes	734	26.5	60.3%	14.9	5.6% (4.1 to 7.4)
Perplexity	Yes	754	27.3	57.7%	15.9	9.1% (7.0 to 11.4)
Gemini	Yes	501	18.7	61.7%	12.5	9.4% (6.6 to 12.3)
Claude	Yes	414	24.2	57.7%	15.0	9.6% (7.0 to 12.3)

(Grok's own figure is not zero because the prior for each row is computed leaving out the prompt that row belongs to, so a brand Grok named only for one prompt has a prior of zero when that prompt is the one being scored.)

The second half of H2d is supported. In unsourced answers between 1.5% and 3.1% of brand mentions went to brands outside the non-searching engine's repertoire. In sourced answers the share was 5.6% to 9.6%, and for each engine observed in both conditions it rose: from 3.1% to 5.7% for ChatGPT, and from 1.5% to 9.6% for Claude. The intervals for Claude do not overlap; those for ChatGPT overlap slightly.

The same table shows how small the opening is. Even for the engines most affected, nine in ten brand mentions in a sourced answer go to brands the parameters already favoured. The top five brands in a category take between 56% and 62% of all mentions in sourced answers, hardly different from the 62% to 71% in unsourced ones, and the effective number of brands per category is between twelve and sixteen for every searching engine, against fifteen for Grok without search. Search adds a tail. It does not flatten the head.

8.3 Prior and source at the level of the brand

We now model directly which brands are named in a sourced answer. The data are the 630,867 pairs of a sourced answer (from the five engines that searched) and a brand in that answer's category universe. The brand was named in 1.25% of pairs. Four predictors are entered:

- **Prompt prior**: the share of Grok's answers *to the same prompt* that named the brand (zero, one third, two thirds or one, since each prompt was answered three times).
- **Category prior**: the log-odds of the brand's prior for the category, computed leaving out the prompt in question.
- **Own site among sources**: a source with the brand's registrable domain is among the answer's sources.
- **Brand in a source title**: the brand's canonical name appears as a whole word in the title of at least one source.

The last two describe the retrieved material from two sides. The own-site variable says the brand's own page reached the context. The title variable says a page whose title mentions the brand reached the context, which is typically a third-party page: a listicle, a comparison, a review.

Table 12 reports the models.

Table 12. Odds ratios for a brand being named in a sourced answer. Logistic regression on answer-and-brand pairs with engine and category terms; errors clustered on prompt (238 clusters in the pooled models); 95% intervals in parentheses. Pseudo-R^2 is one minus the ratio of residual to null deviance.

Sample	Pairs	Prompt prior (0 to 1)	Category prior (per unit log-odds)	Own site among sources	Brand in a source title	Pseudo-R^2
All engines, full model	630,867	12.9 (9.4 to 17.7)	1.99 (1.88 to 2.10)	3.12 (2.35 to 4.14)	11.7 (8.9 to 15.2)	0.549
All engines, prior only	630,867	15.3 (11.1 to 21.1)	2.04 (1.92 to 2.16)			0.518
All engines, sources only	630,867			12.6 (10.2 to 15.6)	14.9 (12.4 to 17.9)	0.234
ChatGPT	53,536	29.6 (19.0 to 45.9)	2.19 (2.03 to 2.38)	8.56 (4.64 to 15.8)	14.3 (9.8 to 20.9)	0.643
Claude	83,839	12.9 (8.5 to 19.4)	1.99 (1.86 to 2.13)	7.12 (3.73 to 13.6)	6.1 (2.8 to 13.2)	0.504
Perplexity	186,527	14.3 (9.6 to 21.5)	1.95 (1.83 to 2.08)	2.61 (1.80 to 3.79)	9.2 (6.1 to 13.8)	0.540
Google AI Overview	180,042	10.9 (7.8 to 15.4)	2.00 (1.87 to 2.14)	1.61 (0.95 to 2.72)	22.6 (14.5 to 35.1)	0.552
Gemini	126,923	9.6 (5.8 to 15.9)	1.82 (1.66 to 2.00)	4.12 (2.50 to 6.80)	not estimable	0.412

(Gemini's sources carry the site name in place of a page title, so the title variable never varies for it.)

H3a: the prior predicts naming. It does, and it is the dominant term. With the prior alone the model reaches a pseudo-R^2 of 0.518. A brand that the non-searching engine names for this very prompt on every occasion has about fifteen times the odds of being named by a searching engine

as one it never names for the prompt, and each unit of log-odds in the category-level prior doubles the odds again. This holds for every engine separately, with category-prior odds ratios between 1.8 and 2.2. The engines are different models from different companies with different search back ends, and a single non-searching model's habits predict all of them. That is strong evidence that what we are calling the prior is not an idiosyncrasy of Grok but a shared property of models trained on overlapping text: they have absorbed much the same picture of which companies belong to which category.

H3b: retrieved text predicts naming beyond the prior. It does. In the full model, a brand whose own site is among the sources has 3.1 times the odds of being named, and a brand that appears in a source's title has 11.7 times the odds, both holding the prior fixed. Adding the two source variables to the prior raises the pseudo-R^2 from 0.518 to 0.549. On their own they reach 0.234.

The comparison of those three numbers is the main quantitative result of this study. Read as shares of what the full model explains, the prior alone recovers 94% of it and the source variables alone 43%; the two overlap heavily because brands with strong priors are also the brands whose pages and whose mentions in titles are retrieved. The increment from adding sources to the prior is 0.03. The increment from adding the prior to sources is 0.32. Knowing what the engine retrieved tells you a good deal about whom it will name. Knowing whom a model would have named anyway tells you much more.

Third-party titles outweigh own pages. In the pooled model and in three of the four engines where both can be estimated, the title variable has the larger odds ratio. For AI Overview the contrast is extreme: a brand in a source title has 22.6 times the odds of being named, while the brand's own site among the sources has an odds ratio of 1.6 with an interval that includes one. A page that names a brand in its title is, in these categories, almost always a page about a set of brands, a "best of" list or a head-to-head comparison, and such a page presents the brand to the model in exactly the frame the answer needs: as a candidate with reasons attached. A brand's own page presents the brand as the only candidate. ChatGPT and Claude are the engines for which the own-site effect is large (8.6 and 7.1) and comparable to the title effect.

H3c: retrieved text matters most where the prior is weak. Table 13 tabulates the naming rate by prior band and by whether the brand's own site was among the sources, and Figure 5 plots it.

Table 13. Share of answer-and-brand pairs in which the brand was named, sourced answers, by category prior and own-site status.

Category prior (share of non-searching answers naming the brand)	Own site not among sources	Pairs	Own site among sources	Pairs	Ratio
Never	0.09%	558,556	7.1%	1,611	80
Up to 2%	1.9%	19,442	21.1%	388	11
2% to 10%	3.4%	29,283	32.5%	508	9.4
10% to 30%	14.2%	10,848	51.5%	534	3.6
Over 30%	33.3%	8,278	75.6%	1,419	2.3

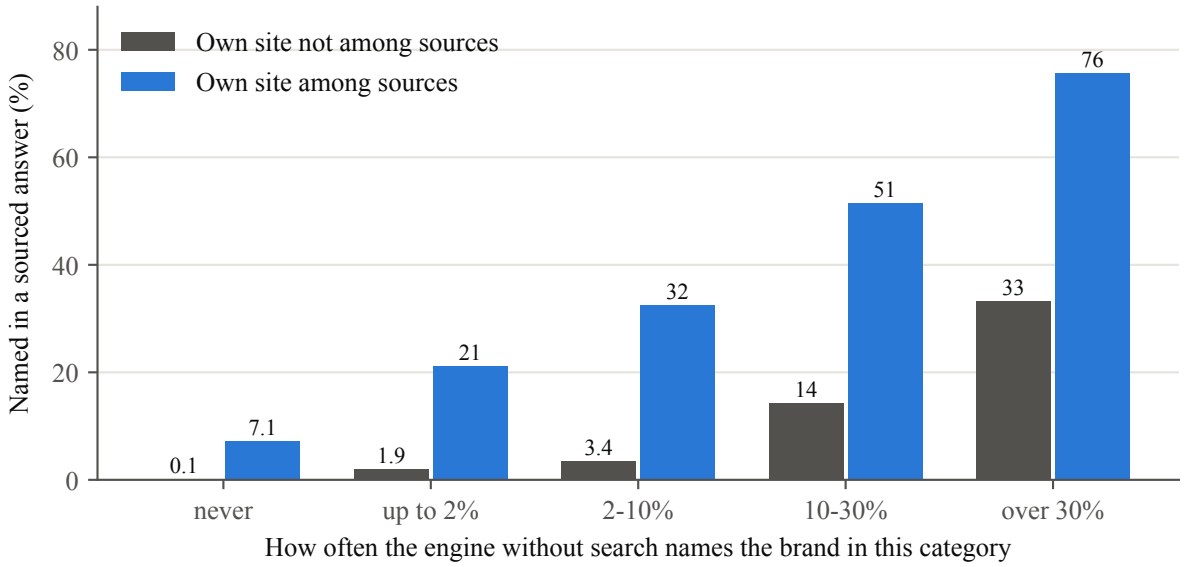


Figure 5. Share of sourced answers naming a brand, by how often the non-searching engine names that brand for the category and by whether the brand's own site was among the answer's sources.

The pattern is as H3c predicts. For a brand with no prior, having its own site among the sources is associated with a naming rate 80 times higher, 7.1% against 0.09%. For a brand with a strong prior the ratio is 2.3, 76% against 33%. In relative terms retrieval matters enormously for the unknown and modestly for the known.

In absolute terms the picture reverses, and both readings are true. Retrieval moves an unknown brand from one answer in a thousand to one in fourteen. It moves a well-known brand from one in three to three in four. The well-known brand gains 42 percentage points and the unknown one gains 7. And the unknown brand with its page in the context is still named less often (7.1%) than the well-known brand *without* its page in the context (33%).

That last comparison is the clearest statement in the paper of how prior and context combine. A page in the context is neither necessary nor sufficient. A strong prior does most of the work unaided. A page in the context without a prior behind it usually fails: 93 times in 100 the model had the unknown brand's own page in front of it and named someone else. This is the confirmation bias that Xie et al. observed in a laboratory setting [18], seen here in production: the model takes from the retrieved text what agrees with what it already holds.

Two caveats limit how far Table 13 can be pushed. First, 558,556 of the 630,867 pairs are in the never-named, not-retrieved cell. The roster universes are large, 89 to 487 brands per category, and most brands in them are obscure; the interesting cells are small by comparison, though still in the hundreds or thousands. Second, the own-site variable is not randomly assigned. A brand's site is retrieved because the search service judged it relevant to the query the model wrote, and the

prompts for which that happens are prompts for which the brand is pertinent. The prompt-level prior absorbs some of this, since it measures how apt the brand is for this particular prompt in the eyes of a model without search, but it cannot absorb all of it.

As a check on the prior measure, we fitted the prior-only model to ChatGPT's and Claude's *unsourced* answers (235,679 pairs, 786 answers). There the source variables are zero by construction and only the prior can operate. The category-prior odds ratio is 2.20 (2.05 to 2.36) and the pseudo-R^2 0.544, about the same as in sourced answers. One model's parametric habits predict another's, with or without search.

8.4 Where in the answer a brand appears

For the 7,884 named pairs in sourced answers with a recorded position, the brand's first appearance fell on average 38% of the way through the answer. A regression of that relative position on the same predictors, with engine and category terms and errors clustered on prompt, finds that brands with stronger priors appear earlier (-2.0 percentage points of the answer's length per unit of log-odds; 95% interval -2.6 to -1.4) and that brands appearing in a source title appear earlier (-6.0 points; -7.8 to -4.2). Having one's own site among the sources is not associated with position (-1.0 points; -2.9 to $+1.0$).

So the prior affects not only whether a brand is named but how early, and a third-party page that features the brand moves it up. Aggarwal et al. weighted a source's visibility by position on the reasoning that earlier text is read more [1]; on that reasoning the advantage of well-known brands is somewhat larger than the naming rates alone show.

8.5 Stability of the named set from month to month

Each prompt was answered by each engine in three successive months. Table 14 gives the mean Jaccard similarity between the sets of brands named for the same prompt by the same engine in two different months, split by whether both answers, neither, or one had sources.

Table 14. Month-to-month Jaccard similarity of the set of brands named for the same prompt by the same engine. Intervals are 95% cluster bootstrap intervals over prompts. Pairs in which neither answer named a brand are excluded.

Engine	Both with sources	Both without	One with, one without
Grok		0.62 (0.58 to 0.66), 740 pairs	
ChatGPT	0.60 (0.55 to 0.64), 265	0.27 (0.20 to 0.33), 213	0.36 (0.30 to 0.42), 156
Claude	0.49 (0.45 to 0.53), 362	0.40 (0.28 to 0.51), 110	0.15 (0.09 to 0.22), 105
Perplexity	0.48 (0.44 to 0.52), 654		
Google AI Overview	0.35 (0.30 to 0.39), 685		
Gemini	0.30 (0.24 to 0.36), 194		

No engine names the same brands for the same question from one month to the next. The most stable is Grok without search, at 0.62, followed by ChatGPT with search at 0.60. A Jaccard similarity of 0.6 between two lists of five brands means they share about four and differ in one or two. The least stable are Gemini and AI Overview, at 0.30 and 0.35: two lists of three brands that share one or two. The share of pairs with identical sets is 23% for Grok and between 6% and 10% for the searching engines.

We expected search to reduce stability, since it adds the variability of the retrieved set to the variability of generation. For Grok against the always-searching engines that is what the table shows. But ChatGPT is far *more* stable with sources (0.60) than without (0.27), and Claude somewhat more (0.49 against 0.40). The explanation is again in which prompts go unsearched. ChatGPT's unsourced answers are to prompts where naming brands is optional, and whether it happens to mention any on a given occasion is close to arbitrary, so the sets vary. When it searches, the retrieved pages anchor the answer to a list.

The mixed column is a direct, if rough, measure of what search changes: for the same prompt and engine, an answer with sources and an answer without share only 36% (ChatGPT) or 15% (Claude) of their brands. For Claude, a searched answer and an unsearched answer to the same question are nearly different answers.

8.6 Summary of Study 2

On fixed sets of category questions, the brands an engine names are predicted mainly by a prior that the engines share and that can be measured from a model with no search. Retrieved material adds to that prediction, more through third-party pages that feature the brand in their titles than through the brand's own pages, and adds the most, in relative terms, for brands with no prior at all. But the absolute naming rate for an unknown brand whose page was retrieved (7%) remains below that of a well-known brand whose page was not (33%). Search extends the list of brands that can be named by a few per cent at the tail and leaves the head as concentrated as before. And the named set for a given question changes substantially from month to month for every engine.

9. Study 3: Which sites are cited

Study 3 turns from the prose of the answer to its sources. It uses every source-era answer from a full scan that returned at least one source, whether or not the account could be scored for mentions: 50,765 answers carrying 476,973 source records, which point at 245,267 distinct URLs on 61,222 sites, from the scans of 116 accounts. It addresses RQ4: how concentrated citations are, which sites receive them, how far engines agree with one another, how stable one engine is over time, and how a site being cited relates to its owner being named.

A reminder of what "cited" means for each engine (Section 2.7) is needed before the numbers. For ChatGPT and Grok a source is a URL attached to a passage of the answer. For Perplexity it is an entry in the list of search results returned with the answer. For Claude before 30 September it is any result the search tool returned, cited or not; Study 4 separates the two for the later period. For Gemini and AI Overview it is a grounding source or reference. The engines' figures are comparable as "what the engine reported drawing on" and not as "what the engine's text was demonstrably based on".

9.1 Concentration

Table 15 summarises the distribution of citations over sites for each engine, and Figure 6 plots the rank-frequency curves on logarithmic axes.

Table 15. Concentration of citations over sites, by engine. "Sites for half" is the number of most-cited sites that together receive half of the engine's citations. α is the maximum-likelihood exponent of a discrete power law fitted above the cutoff that minimises the Kolmogorov-Smirnov distance; standard error in parentheses. R is the normalised log-likelihood ratio against a log-normal (negative favours the log-normal).

Engine	Citations	Sites	Top-10 share	Top-100 share	Sites for half	Gini	Effective sites	Cited once	α	R (p)
Perplexity	206,500	35,199	9.5%	24.5%	902	0.74	5,835	51.9%	2.00 (0.01)	−3.9 (<0.001)
Claude	81,338	12,981	9.4%	32.7%	331	0.75	2,436	52.6%	2.03 (0.02)	−1.9 (0.06)
Gemini	61,453	19,121	4.2%	18.3%	1,254	0.60	6,708	62.6%	2.07 (0.01)	−5.3 (<0.001)
Google AI Overview	51,114	9,171	24.7%	44.5%	159	0.75	1,087	59.3%	1.95 (0.01)	−2.2 (0.03)
ChatGPT	50,020	13,108	10.1%	25.9%	601	0.66	3,481	62.0%	2.02 (0.01)	−5.6 (<0.001)
Grok	25,459	7,291	8.5%	27.6%	434	0.62	2,458	60.8%	2.02 (0.01)	−3.5 (<0.001)

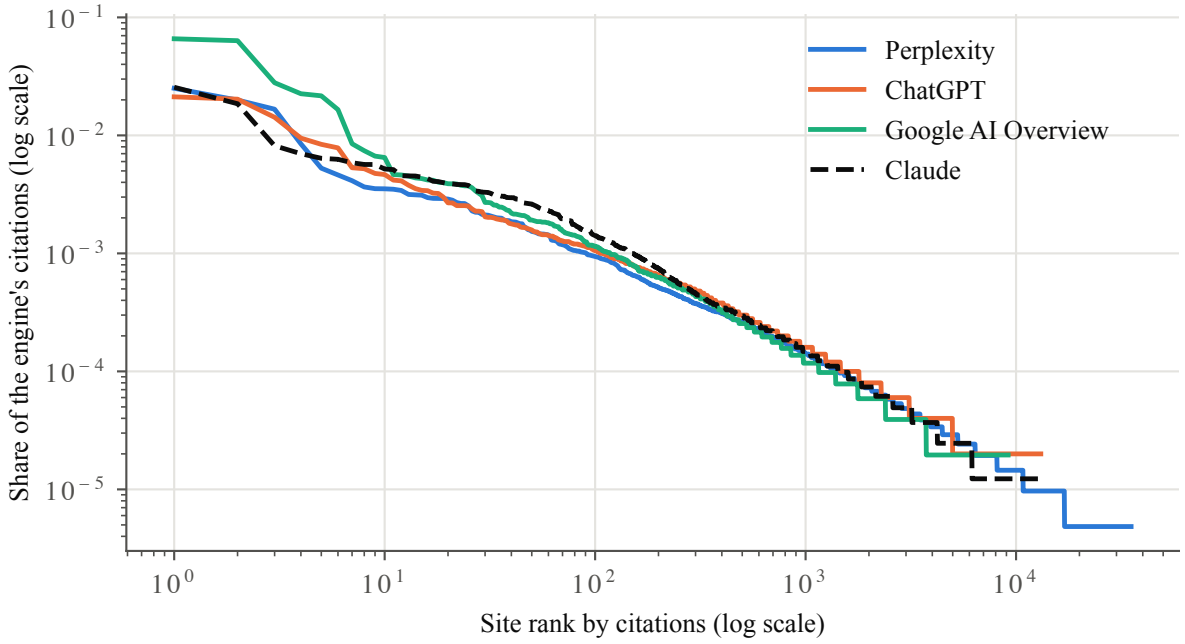


Figure 6. Share of each engine's citations received by each site, against the site's rank, on logarithmic axes. A straight line on these axes is a power law. Google AI Overview's curve sits above the others at the head: its ten most-cited sites take a quarter of its citations.

Citations are spread across a very large number of sites and are very unevenly spread. Three features hold for every engine.

A majority of cited sites are cited once. Between 52% and 63% of the sites an engine cited anywhere in the panel appear a single time. This long tail is the dominant fact about the population of cited sites, and it would be longer in a panel with more varied prompts.

A few hundred sites take half the citations. Half of Perplexity's 206,500 citations fall on 902 of its 35,199 sites, 2.6% of them. For Claude it is 331 of 12,981, for AI Overview 159 of 9,171. Gini coefficients run from 0.60 to 0.75.

The tail exponent is two. The fitted power-law exponent lies between 1.95 and 2.07 for every engine, with standard errors of 0.01 or 0.02. That is the exponent of Zipf's law for the frequency distribution, the form found for word frequencies, city sizes and many other quantities that grow by cumulative advantage [63, 64], and it means, as noted in Section 6.8, that the mean number of citations per site is dominated by the largest sites and is not a stable summary.

H4a asked specifically whether the tail is a power law, and on that the answer is a qualified no. For five of the six engines the likelihood-ratio test favours a log-normal over a power law at the 5% level, and for the sixth, Claude, it leans the same way ($p = 0.06$). The distributions are heavy-tailed, and a power law with exponent two describes them well enough for the summaries above, but the top of the distribution falls off faster than a pure power law would have it. In Figure 6 this is the slight downward bend of each curve on the left. A natural reading is that the largest sites

are capped: no single site can be cited for more than a modest share of prompts, because the prompts are about different things. This is the pattern Clauset et al. found for many empirical distributions that had been described as power laws [54].

The engines differ in how concentrated they are, and the differences are interpretable. Google AI Overview is by far the most concentrated at the head, with a quarter of its citations going to ten sites and an effective number of sites of 1,087. Gemini is the least, with 4% to its top ten and an effective number of 6,708. The two share a search back end and differ this much because of what they cite from it, as the next section shows: AI Overview leans on a handful of very large platforms.

9.2 The most widely cited sites

To say which sites are cited most without revealing any account's market, Table 16 lists sites by the share of sourced answers that cite them, restricted to third-party sites cited in the answers of at least ten different accounts, and reports the share of accounts whose answers ever cite each.

Table 16. The most widely cited third-party sites: share of the 50,765 sourced answers citing the site at least once, and share of the 116 accounts with at least one such answer. Sites cited for fewer than ten accounts are excluded.

Site	Kind	Share of sourced answers	Share of accounts	Engines (of 7)
reddit.com	Community	10.9%	95%	6
linkedin.com	Community	10.7%	78%	7
youtube.com	Community	9.3%	82%	6
wikipedia.org	Reference	4.5%	91%	7
hubspot.com	Vendor	3.6%	31%	7
g2.com	Review directory	3.5%	50%	7
facebook.com	Community	3.5%	69%	6
google.com	Google property	3.5%	75%	6
semrush.com	Vendor	3.4%	20%	7
yelp.com	Review directory	3.1%	41%	4
seranking.com	Vendor	3.0%	13%	6
medium.com	Community	2.9%	51%	6
zapier.com	Vendor	2.7%	27%	7
onelittleweb.com	Publisher	2.1%	10%	5
peoplemanagingpeople.com	Publisher	1.8%	13%	7
clutch.co	Review directory	1.8%	32%	7
gartner.com	Review directory	1.8%	44%	7
salesforce.com	Vendor	1.8%	32%	7
indeed.com	Jobs	1.7%	41%	7
arxiv.org	Research	1.6%	28%	7
forbes.com	News	1.5%	53%	7
github.com	Developer	1.4%	28%	6
nih.gov	Government	1.4%	36%	7

Three community platforms stand apart. Reddit, LinkedIn and YouTube are each cited in about one sourced answer in ten, and each for more than three quarters of all accounts. Wikipedia is cited less often per answer, 4.5%, but for 91% of accounts: it turns up a little for nearly everyone. Below these the list is a mixture of review directories (G2, Yelp, Clutch, Gartner), large vendors whose own content marketing ranks for many business queries (HubSpot, Semrush, Zapier, Salesforce), and general publications. The vendor entries reflect the panel's tilt towards marketing and sales software and would differ in a different panel. The platform entries would not.

The picture by engine is far less uniform than the pooled table suggests. Table 17 gives each engine's most-cited third-party sites.

Table 17. Each engine's most-cited third-party sites, with the share of that engine's answers that cite any third-party site which cite this one. Same restriction to sites cited for at least ten accounts.

Engine	Most-cited sites (share of the engine's sourced answers)
ChatGPT	reddit.com 11.2%, wikipedia.org 10.7%, g2.com 4.5%, arxiv.org 2.9%, techradar.com 2.5%
Claude	wikipedia.org 13.0%, yelp.com 11.2%, linkedin.com 4.6%, g2.com 3.7%, medium.com 3.6%
Gemini	medium.com 4.3%, youtube.com 2.8%, reddit.com 2.6%, zapier.com 2.5%, gartner.com 2.2%
Perplexity	linkedin.com 22.8%, reddit.com 21.2%, youtube.com 15.6%, hubspot.com 8.1%, g2.com 6.1%
Google AI Overview	youtube.com 29.9%, linkedin.com 17.5%, google.com 16.1%, reddit.com 15.9%, facebook.com 13.0%
Grok (while it searched)	linkedin.com 10.0%, reddit.com 5.0%, youtube.com 4.1%, clutch.co 2.9%, medium.com 2.5%

The engines have distinct citation profiles.

Google AI Overview cites YouTube in 30% of its overviews, LinkedIn in 17%, a Google property in 16%, Reddit in 16% and Facebook in 13%. Its five most-cited sites are all user-generated or Google-owned platforms. This is what makes it the most concentrated engine in Table 15.

Perplexity resembles it in favouring LinkedIn (23%), Reddit (21%) and YouTube (16%), with less dependence on any one.

ChatGPT and Claude are the two engines for which Wikipedia is at or near the top, in 11% and 13% of sourced answers, and for which the community platforms are much less prominent. YouTube and LinkedIn do not appear in ChatGPT's top five at all. Claude's second most-cited site is Yelp, at 11%, which probably reflects the local-service accounts among those scanned on Claude.

Gemini has no dominant source. Its most-cited site, Medium, appears in 4.3% of its sourced answers. Wikipedia is nearly absent from Gemini's citations in this panel, at 0.2% of citations, which is striking given that Zhang et al. found Wikipedia to account for 41% of Gemini's citations on their query set [30]. The discrepancy is almost certainly one of prompts. Theirs covered general knowledge and news, for which an encyclopedia is the obvious source; ours are commercial, for which it is not. It is a useful caution against reading any engine's "favourite sources" off one study.

9.3 Kinds of site

A classification of cited sites into kinds was attempted with a small set of rules for well-known platforms, directories, reference works, news publishers and government or education domains, supplemented by a stored classification of 650 domains. It covers 2.4% of sites. Together with the account's own site, which is identified separately, it places 17% of citations. The remaining 83% of citations go to sites that the rules cannot place, overwhelmingly the websites of individual companies and independent publishers, each cited a handful of times. That residual is itself the main finding of the exercise: more than four citations in five go to the long tail of ordinary business and publisher sites, and not to the platforms that head Table 16.

Within the classified portion, Table 18 gives the shares.

Table 18. Share of each engine's citations by kind of site, for the kinds that could be classified. The unclassified remainder is company and independent publisher sites.

Kind	ChatGPT	Claude	Gemini	Perplexity	AI Overview	Grok
The account's own site	2.8%	3.1%	1.9%	2.7%	2.2%	3.1%
Social and community platforms	2.7%	1.5%	2.3%	7.2%	15.5%	5.8%
Review directories	2.7%	5.4%	1.5%	3.5%	2.9%	1.9%
Government and education	4.0%	2.6%	1.6%	2.3%	0.9%	2.4%
Encyclopedias and reference	2.0%	1.9%	0.2%	0.3%	0.2%	0.5%
News and magazines	1.0%	1.0%	0.8%	1.4%	0.9%	1.6%
Google properties	0.4%	0.1%	0.2%	0.3%	6.6%	0.2%
Unclassified (company and publisher sites)	84.4%	84.4%	91.6%	82.3%	70.8%	84.6%

Two rows deserve comment. The account's own site receives between 1.9% and 3.1% of an engine's citations, with little variation across engines (cluster intervals for the six engines all lie within 1.3% to 6.2%). For a brand, that is the baseline: roughly one in forty of the sources an engine cites for the prompts the brand has chosen to track is the brand's own. And social and community platforms receive 15.5% of AI Overview's citations against 1.5% to 2.7% for Claude, ChatGPT and Gemini, a sixfold to tenfold difference between engines in how much weight user-generated content carries.

Chen et al. reported that AI engines favour earned media, independent sites writing about brands, over brand-owned and social content [28]. Our classification is too coarse to test the earned-versus-owned split directly, since both fall in the unclassified remainder. What can be said is that social content is far from absent in this panel for two of the engines, and that the engines disagree with each other on this dimension more than on any other.

9.4 Agreement between engines

When several engines answer the same prompt in the same scan, how much do their sources overlap? For each such prompt and scan, and each pair of engines that both returned sources, we computed the Jaccard similarity of the two sets of cited sites: 65,083 pairs in all. The pairwise means are shown in Figure 7.

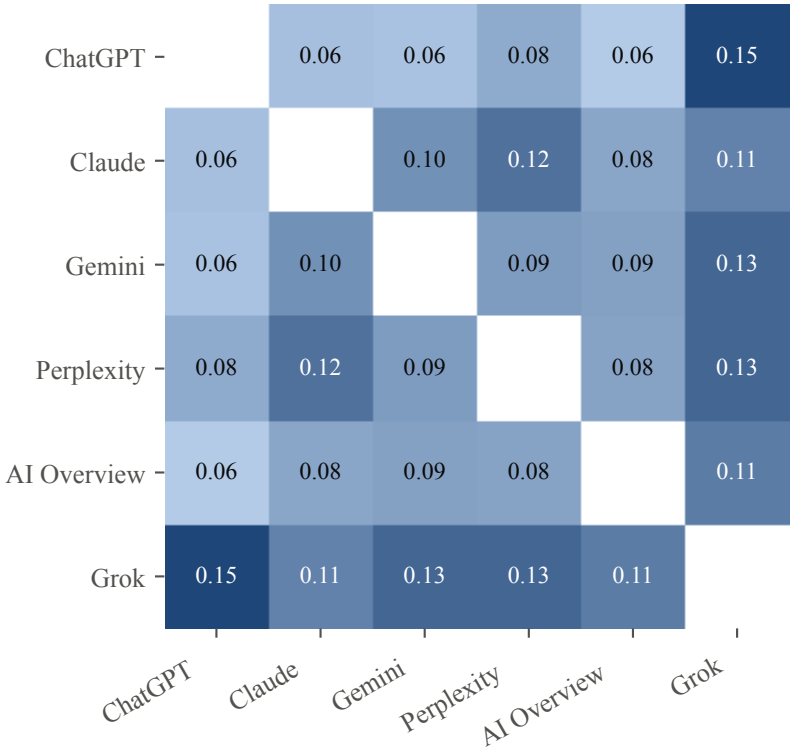


Figure 7. Mean Jaccard similarity between the sets of sites two engines cite for the same prompt in the same scan. All values are below 0.15.

The mean over all pairs is **0.093**. In 36% of pairs the two engines cite no site in common. The most similar pair, Grok and ChatGPT, has a mean of 0.15; the least similar, ChatGPT and AI Overview, 0.06. A Jaccard similarity of 0.09 between two sets of eight sites corresponds to slightly more than one shared site.

Looked at from the side of the sites: for the 10,993 prompt-and-scan combinations in which at least three engines returned sources, 83.6% of the distinct sites cited by any engine were cited by exactly one engine, 12.6% by two, 3.1% by three, and under 1% by four or more. A site cited by six engines for the same prompt was a rarity, about one in ten thousand.

H4b is strongly supported. For the same question at the same moment, the engines draw on almost entirely different parts of the web.

Three mechanisms described in Section 2 each contribute, and the data cannot apportion the disagreement among them. The engines use different search back ends with different indexes and rankings. Each engine's model rewrites the prompt into its own search queries, so even two engines on the same back end would search for different things; Section 11.2 shows how different. And each engine returns a different number of sources drawn from a different depth of the result list.

ChatGPT has the lowest overlap with every other engine (0.06 to 0.08 with all but Grok). Perplexity has the highest share of pairs with at least one site in common, 62% to 85%, simply because it returns the most sources per answer, a median of ten and a mean of fourteen.

The practical consequence is that a result established on one engine does not transfer. A site's standing as a source for one engine says little about its standing for another, and a brand's visibility cannot be characterised by measuring a single engine.

9.5 Stability of one engine over time

The same prompt read again by the same engine does not return the same sources either. Table 19 reports, for consecutive readings of the same prompt by the same engine in which both returned sources, the Jaccard similarity of the sets of cited sites and of cited URLs.

Table 19. Similarity between the sources of consecutive readings of the same prompt by the same engine. Intervals are 95% cluster bootstrap intervals over accounts.

Engine	Pairs of readings	Prompts	Median days apart	Site Jaccard	URL Jaccard	No site in common
Claude	4,389	1,076	10.9	0.37 (0.33 to 0.45)	0.33 (0.29 to 0.39)	1.0%
Perplexity	7,321	2,624	9.9	0.37 (0.34 to 0.41)	0.30 (0.27 to 0.34)	1.5%
Grok (while it searched)	1,988	1,227	9.6	0.25 (0.21 to 0.30)	0.20 (0.16 to 0.23)	20.6%
ChatGPT	3,153	1,251	9.1	0.22 (0.19 to 0.26)	0.11 (0.08 to 0.15)	16.1%
Google AI Overview	4,153	869	11.8	0.20 (0.17 to 0.25)	0.10 (0.08 to 0.12)	22.2%
Gemini	4,054	1,563	12.5	0.19 (0.17 to 0.24)	not comparable	22.1%

(Gemini returns a fresh redirect URL for each source on each request, so its URLs cannot be compared across readings; its sites are recovered from titles.)

Reading the same prompt about ten days later, an engine cites a set of sites with a mean similarity to its earlier set of between 0.19 and 0.37. Claude and Perplexity are the most stable, and are the two engines that return the most sources per answer; with ten sources apiece, a similarity of 0.37 is about five or six sites in common. For AI Overview, Gemini and Grok, about one pair of readings in five has no site in common at all, and for ChatGPT one in six. At the level of exact URLs similarity is lower still: 0.10 for AI Overview and 0.11 for ChatGPT.

Stability decays with time. Grouped by the gap between readings, the mean site similarity is 0.33 for readings on the same day, 0.38 at two to seven days, 0.28 at eight to thirty days and 0.19 beyond thirty days. The Spearman correlation between the gap in days and the similarity is -0.22 ($p < 0.001$, 25,085 pairs). H4c is supported.

The same-day figure is the most telling, because nothing about the web or the model has changed in a day. Two readings of the same prompt by the same engine on the same day share, on average, a third of their sites. This is the floor set by the randomness of generation and of the search back end's response to the model's freshly written query. It agrees with what the instrument's operators observed separately for AI Overview, where two simultaneous reads of the same prompt shared no cited domain on three prompts out of three. A change in which sources an engine cites for a prompt between two dates is, most of the time, noise.

(The slightly higher figure at two to seven days than on the same day reflects which accounts are rescanned at those intervals, and is not evidence that similarity rises.)

9.6 Cited and named

The last question of the study connects the two halves of the paper. When an account's own site is among the sources of an answer, is the account named in the answer?

On unbranded prompts with sources (46,258 answers from 112 accounts), the account's own site was among the sources in 6.8% of answers (cluster interval 3.9% to 9.5%). Table 20 gives the rate at which the brand was named, by whether its site was cited.

Table 20. Unbranded prompts with sources: share of answers naming the account's brand, by whether the account's own site was among the sources.

Engine	Answers	Own site cited	Named, site cited	Named, site not cited
ChatGPT	6,837	3.6%	66.9%	0.82%
Grok (while it searched)	4,599	3.1%	44.7%	0.67%
Claude	6,420	12.0%	39.4%	0.51%
Gemini	9,011	5.9%	37.5%	0.45%
Perplexity	13,176	7.2%	34.4%	0.66%
Google AI Overview	6,056	8.4%	33.3%	0.32%
All	**46,258**	**6.8%**	**39.1%**	**0.58%**

The association is very strong. When the account's site is among the sources, the brand is named in 39.1% of answers (Wilson interval 37.4% to 40.8%). When it is not, the brand is named in 0.58% (0.52% to 0.66%). The pooled odds ratio with engine terms and errors clustered on account is 126 (95% interval 58 to 275). Restricting to readings of the same prompt by the same engine in which the site was sometimes cited and sometimes not, which holds the prompt, the brand and the engine fixed, the conditional odds ratio is **73.5** (37.4 to 144.3; 217 informative prompt-and-engine groups). The first half of H4d is supported.

So is the second. In 61% of the answers that cited the account's own site, the answer did not name the account. On AI Overview, Perplexity and Gemini it is about two answers in three. An engine can retrieve a company's page, use a fact or a definition from it, attach the citation, and write a passage in which the company does not appear. The cited URLs show the typical case. In the answers that cited the account's site and did not name the account, the cited page was an article, guide or blog post in 58% of cases and the homepage in 10%. In the answers that did name the account, it was an article in 28% and the homepage in 29%. The engine has treated the brand's article as it would any other article on the topic, as a source of information and not as a subject. The brand published the page. The engine cited the page. The answer is about something else.

ChatGPT is the exception, naming the brand in two thirds of the answers that cite its site. ChatGPT also cites an account's own site least often (3.6% of sourced answers, against 6% to 12% for the other engines apart from Grok). The two facts fit a single account: ChatGPT cites a company's own site mainly when it is writing about that company.

The estimate of 73.5 should not be read as the effect of getting one's site cited. As Section 6.12 set out, the readings on which a brand's site is retrieved are the readings on which the search service, given the model's query, judged the brand relevant, and a model whose query led to the brand's site may already have been inclined towards the brand. The within-prompt design removes differences between prompts and cannot remove differences between readings. The defensible statements are two. Unprompted naming of a small brand almost never happens without the brand's own site in the sources: 0.58%. And the brand's own site in the sources is far from enough: 39%.

On branded prompts the relation disappears, as it should. The brand is named in 94.1% of answers that cite its site and 90.9% of answers that do not; the pooled odds ratio is 1.36 (0.65 to 2.85) and the within-prompt odds ratio 2.13 (0.85 to 5.35). When the name is in the prompt it is in the answer either way. The account's site is cited in 73% of sourced answers to branded prompts, more than ten times the rate for unbranded prompts, which is the search service doing the obvious thing with a query that contains a company's name.

9.7 Summary of Study 3

Citations are heavy-tailed, with most cited sites cited once and a few hundred taking half of each engine's citations; the tail is somewhat lighter than a power law. More than four citations in five go to ordinary company and publisher sites outside any list of well-known platforms. The engines have distinct source profiles, with AI Overview and Perplexity leaning on community platforms and ChatGPT and Claude on reference works, and the profiles depend on the kind of prompt. Two engines answering the same question at the same time share, on average, under a tenth of their combined sites, and the same engine asked again ten days later repeats between a fifth and a third of its sites. An account's own site among the sources is associated with a seventy-fold increase in the odds of the account being named, and in three answers out of five that cite the site, the account is not named.

10. Study 4: Of the pages retrieved, which are cited

Every observational study of what AI engines cite runs into the same wall. One can list the pages an engine cited and describe them, but a description of the cited says nothing about selection unless there is something to compare them with. Pages that were not cited are an unbounded set, and comparing cited pages with the web at large, or with a conventional search result list for the user's prompt, compares them with pages the model may never have seen.

Claude's search tool removes the wall. As Section 2.7 described, a Claude response lists every result its searches returned, in order, and separately marks which results it cited and for which passages. Since 30 September 2026 the instrument has stored both. Each answer is therefore a small, complete choice problem: these pages were placed in the model's context, in this order, and the model cited these ones. The comparison group is exactly right. It consists of pages that the search service judged relevant to the model's own query, that were put in front of the model in the same request, and that the model passed over.

10.1 The choice sets

The data are 3,759 Claude answers from 32 accounts between 30 September and 14 November 2026, containing 45,035 retrieved results. Claude cited 19,105 of them, 42.4%.

The median answer had nine results in its context and cited five. Answers that searched more than once had up to forty. In only 0.4% of answers was every retrieved result cited, and in almost every answer at least one was, so nearly all answers contain both outcomes and contribute to a within-answer analysis: 3,706 of the 3,759. A cited result backed a mean of 1.6 separate passages of the answer.

That 58% of what is retrieved goes uncited is the first result. Retrieval is necessary for citation and well short of sufficient. More than half the pages that win a place in the context are read, in whatever sense a model reads, and not credited.

10.2 Position

Table 21 and Figure 8 give the share of results cited by their position in the order returned.

Table 21. Share of retrieved results that Claude cited, by position in the context. Wilson intervals treat results as independent; cluster intervals resample accounts.

Position	Results	Cited	Wilson 95%	Cluster 95%
1	3,759	68.4%	66.9 to 69.8	58.9 to 75.9
2	3,759	60.9%	59.4 to 62.5	53.5 to 67.5
3	3,759	51.8%	50.2 to 53.4	46.8 to 55.9
4	3,759	50.0%	48.4 to 51.6	44.8 to 54.0
5	3,759	47.5%	45.9 to 49.1	42.7 to 51.9
6	3,759	45.4%	43.8 to 47.0	41.1 to 49.2
7	3,759	41.7%	40.1 to 43.2	37.6 to 45.6
8	3,759	39.5%	38.0 to 41.1	35.4 to 42.3
9	3,759	31.8%	30.3 to 33.3	28.8 to 35.3
10	1,856	32.1%	30.0 to 34.3	28.1 to 36.4
11 to 15	4,855	26.1%		
16 to 20	3,221	20.0%		



Figure 8. Share of retrieved results that Claude cited, by the result's position among those placed in its context. The band is the 95% cluster bootstrap interval over accounts. Positions after the ninth or tenth are results of second and later searches.

The first result is cited 68.4% of the time. The share falls at every step, to 31.8% at the ninth, the last position every answer has. The decline is monotonic across the first nine positions, and the cluster intervals at the two ends do not come close to overlapping. H5a is supported.

Positions beyond the ninth or tenth belong to answers in which Claude searched again. Results from later searches are cited less still, around a quarter of the time at positions 11 to 15 and a fifth at 16 to 20. Later searches are typically narrower follow-ups, and the model draws on them for a detail and not for the body of the answer.

Two things produce the gradient, and this design cannot separate them. The search service orders its results by its own estimate of relevance to the query, so earlier results are, on average, better matches, and a model choosing purely on merit would cite them more. And a model reading a long context gives more weight to what comes early [14], so that the same page would be cited more often at position one than at position nine. The first is a property of the pages. The second is a property of the model. What Table 21 measures is their combination, which is the quantity that matters to a publisher: the pages the search service ranks first for the model's query are cited at more than twice the rate of the pages it ranks ninth.

The gradient is also the search engine optimisation result of the paper, in the plainest sense. Ranking for the query the model writes is the largest single determinant of being cited that we can measure, larger than anything about the page's structure. This agrees with Puerto et al., who found on their benchmark that methods improving a document's position in the context beat methods that rewrite it [27].

10.3 What predicts citation within an answer

To estimate the association of other characteristics with citation at a given position, we fit the conditional logistic model of equation (11) with the answer as the stratum. The model compares results only with other results in the same answer, so everything about the prompt, the account, the date and the answer itself is held fixed. Table 22 reports it for the variables available for every result. The second column of estimates comes from an ordinary logistic model with the same terms and errors clustered on account, as a check that the stratified model's precision is not an artefact of ignoring account-level dependence.

Table 22. Odds ratios for a retrieved result being cited. Conditional logit within answer: 44,159 results in 3,706 answers. Benjamini-Hochberg adjusted p-values across the 14 terms. Pooled logit: same terms, errors clustered on 32 accounts.

Predictor	Within-answer odds ratio (95% interval)	Adjusted p	Pooled odds ratio (account-clustered interval)
Log of position	0.49 (0.48 to 0.51)	<0.001	0.50 (0.44 to 0.56)
Result of a second or later search	1.68 (1.55 to 1.81)	<0.001	0.91 (0.84 to 0.98)
Share of the prompt's terms in the title (0 to 1)	1.87 (1.57 to 2.23)	<0.001	2.47 (1.80 to 3.38)
Year 2025 or later in the title	1.21 (1.14 to 1.30)	<0.001	1.16 (1.07 to 1.26)
"Best", "top" or a leading number in the title	1.01 (0.95 to 1.07)	0.83	1.12 (1.00 to 1.26)
Comparison term in the title	1.04 (0.94 to 1.15)	0.44	0.95 (0.88 to 1.02)
The account's own site	2.13 (1.83 to 2.49)	<0.001	1.31 (0.75 to 2.29)
The account's brand in the title	1.86 (1.53 to 2.27)	<0.001	1.42 (1.06 to 1.91)
Homepage URL	1.70 (1.56 to 1.85)	<0.001	1.64 (1.31 to 2.04)
Social or community platform	1.12 (0.88 to 1.42)	0.43	0.76 (0.60 to 0.96)
Review directory	0.80 (0.69 to 0.93)	0.005	0.68 (0.44 to 1.03)
Encyclopedia	0.34 (0.26 to 0.43)	<0.001	0.30 (0.19 to 0.48)
Government or education domain	0.57 (0.49 to 0.67)	<0.001	0.71 (0.43 to 1.16)
Log of how widely the site is cited in the panel	0.85 (0.82 to 0.88)	<0.001	0.87 (0.78 to 0.97)

Position. With everything else in the model, each unit increase in the natural logarithm of position multiplies the odds of citation by 0.49. Moving from the first position to the third, or from the third to the eighth, roughly halves the odds. The estimate is the same in the pooled model with account-clustered errors (0.50, interval 0.44 to 0.56).

The term for later searches needs a word. Its within-answer odds ratio is above one (1.68), which seems to contradict the lower citation rates at positions past the ninth. It does not. The model already contains log position, which predicts very low rates for position 15 or 20; the positive coefficient says that results of a later search are cited somewhat more than their absolute position alone would predict, which is what one would expect if the model's attention restarts, in part, with each new block of results. In the pooled model, where answers with more searches are compared against answers with fewer, the term is slightly below one.

Relevance of the title. The share of the prompt's content words that appear in the result's title is the strongest title-level predictor. A title containing all of them has 1.87 times the odds of a title containing none, within the answer. In raw terms, results whose titles shared none of the prompt's terms were cited 28.7% of the time, those sharing up to a third 41.3%, up to two thirds 50.0%, and more than two thirds 64.3%. H5b is supported. It should be remembered that these are the *user*'s terms and that the search was run on the *model*'s query; a result's title matching the original question, over and above its having ranked for the rewritten one, is associated with being cited. This is the finding of Wan, Wallace and Klein [20] in a production setting: relevance to the question, as plainly signalled, is what a model responds to.

Recency in the title. A title containing 2025 or a later year has 1.21 times the odds of citation. The effect is modest and survives in the pooled model (1.16). Section 11.2 will show that the engines often add the current year to their queries; a title carrying the year both ranks for such a query and, on this evidence, is preferred among the results that do.

List and comparison titles. Titles of the "10 best" form and titles with comparison terms are not cited more than other titles at the same position (1.01 and 1.04). Such pages are very common among what is retrieved for commercial prompts, the list titles alone making up 37% of all results in this sample; the search service retrieves them readily. Once retrieved they hold no advantage.

The account's own site. Results on the account's own site were cited 54.3% of the time against 42.0% for other results, and the within-answer odds ratio is 2.13. But the pooled estimate with account-clustered errors is 1.31 with an interval from 0.75 to 2.29. The within-answer precision is illusory here: own-site results come from a few accounts, and the association differs between them. We do not regard this as established. The same caution applies, less severely, to the brand-in-title term.

Homepages. A result whose URL is a site's root is cited more (1.70; pooled 1.64, interval 1.31 to 2.04). This is stable across both models. A homepage in the results is usually there because the query named or strongly implied the organisation, and it serves as the reference for what the organisation is.

Kinds of site. Encyclopedia results are cited at about a third the rate of other results at the same position (0.34; pooled 0.30, interval 0.19 to 0.48). In raw terms 8.9% of the 1,150 Wikipedia and similar results were cited, against 43.3% of everything else. This is the most surprising coefficient in the table, given that Wikipedia is Claude's most-cited site by share of answers (Section 9.2). The two facts are consistent: Wikipedia is retrieved for a very large share of Claude's searches and cited from a small share of those retrievals. In a panel of commercial prompts, the encyclopedia article that the search returns is typically the general article on the product category, which has little to say to a question about which product to choose. Review directories are also cited less than their position predicts (0.80), as are government and education domains (0.57), though for both the account-clustered intervals include one.

Widely cited sites. The more accounts' answers a site is cited in across the whole panel, the *less* likely a given result from it is to be cited, at a given position (0.85 per unit of log breadth). Large general sites are retrieved often and cited selectively. Being a site that engines know is not, by itself, an advantage once the page is in the context.

10.4 Page structure

For 13,798 of the retrieved results, 30.6%, the instrument had fetched the page and computed its structural features. As Section 5.8 noted, coverage is higher for cited results (38.9%) than for uncited ones (27.9%), because pages are queued for fetching on the strength of being cited some-

where in the panel. The model below conditions on the answer and includes position, so the question it asks is whether, among the fetched results in one answer, those with a given feature were cited more.

Table 23 adds nineteen page-level terms to the model of Table 22, and Figure 9 plots the page-level odds ratios.

Table 23. Odds ratios for a retrieved result being cited, results with a fetched page. Conditional logit within answer: 12,008 results in 1,994 answers. The terms of Table 22 are included and not shown, except position. Adjusted p-values are Benjamini-Hochberg across all 33 terms. Pooled column as in Table 22.

Predictor	Within-answer odds ratio (95% interval)	Adjusted p	Pooled odds ratio (account-clustered interval)
Log of position	0.39 (0.36 to 0.42)	<0.001	0.40 (0.38 to 0.42)
Length and density			
Log of word count	0.83 (0.79 to 0.88)	<0.001	0.88 (0.83 to 0.93)
Statistics per 1,000 words (log)	1.11 (1.06 to 1.16)	<0.001	1.08 (1.03 to 1.13)
Share of paragraphs that are quotable	1.97 (1.60 to 2.43)	<0.001	1.79 (1.40 to 2.29)
Structure			
Log of heading count	1.14 (1.06 to 1.22)	0.002	1.08 (1.03 to 1.14)
Log of list count	0.96 (0.92 to 1.01)	0.19	0.98 (0.91 to 1.06)
Has a table	0.99 (0.88 to 1.12)	0.93	0.95 (0.76 to 1.17)
Has headings phrased as questions	1.15 (0.98 to 1.34)	0.17	1.10 (0.88 to 1.38)
Log of outbound links	0.98 (0.94 to 1.02)	0.37	0.98 (0.96 to 1.00)
Markup and conventions			
Declares structured data	1.24 (1.06 to 1.45)	0.018	1.21 (1.02 to 1.43)
FAQ block	1.07 (0.94 to 1.23)	0.37	1.09 (0.94 to 1.27)
Summary or "TL;DR" box	1.07 (0.96 to 1.20)	0.27	1.06 (1.03 to 1.09)
Definition-style opening sentence	0.94 (0.85 to 1.03)	0.24	0.95 (0.85 to 1.05)
Author byline	0.86 (0.78 to 0.94)	0.004	0.86 (0.74 to 1.00)
Filler introduction	0.70 (0.44 to 1.12)	0.20	0.60 (0.44 to 0.82)
Format (reference: guide)			
Listicle	0.83 (0.73 to 0.96)	0.026	0.83 (0.75 to 0.91)
Comparison table page	0.86 (0.70 to 1.05)	0.20	0.84 (0.60 to 1.17)
Landing page	1.26 (0.97 to 1.64)	0.16	1.00 (0.74 to 1.36)
Review	1.08 (0.70 to 1.67)	0.81	0.88 (0.50 to 1.53)
News	0.55 (0.35 to 0.86)	0.024	0.53 (0.27 to 1.06)

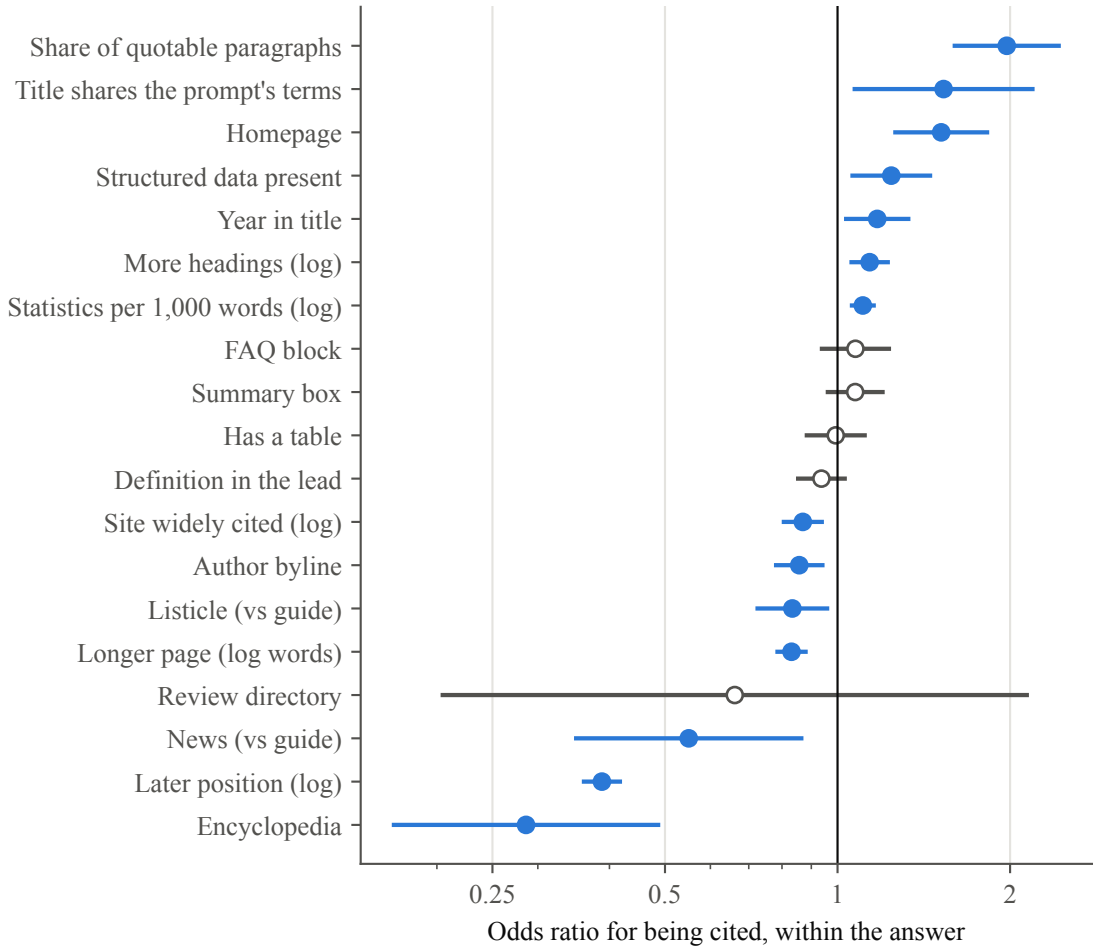


Figure 9. Odds ratios for a retrieved result being cited by Claude, within the answer, for position, title and page-structure predictors (results with a fetched page). Horizontal lines are 95% intervals; filled markers are significant after Benjamini-Hochberg correction.

Shorter pages are cited more. Each unit of log word count multiplies the odds of citation by 0.83: a page twice as long has about 12% lower odds. Cited results had a mean length of 2,749 words and uncited ones 3,459. This runs against the widespread advice that comprehensive long-form content is what AI engines want. The mechanism of Section 2.6 suggests why. The model does not receive the page; it receives an excerpt of bounded size. A long page contributes the same number of tokens to the context as a short one, and the fraction of the long page that the excerpt covers is smaller, so the chance that the excerpt contains the passage that answers the question is lower.

Dense, self-contained passages are cited more. The share of a page's paragraphs classified as quotable, meaning short paragraphs that state something complete without depending on their neighbours, has the largest page-level odds ratio: 1.97 for a page in which all paragraphs are quotable against one in which none are, and 1.79 in the pooled model with a clustered interval that excludes one. A higher density of statistics is also positively associated (1.11 per unit of the log of statistics per thousand words). H5c is supported. Both are what the mechanism predicts. A

citation attaches to a passage that supports a claim. A page built of passages that each carry a claim offers more to attach to than a page whose content is spread through long connected prose, and whatever excerpt the search tool cuts from it is more likely to stand on its own.

This is also where the present results meet those of Aggarwal et al. [1]. Their most effective interventions added statistics and quotations to a source. In our data pages denser in statistics and richer in self-contained statements are cited more. The two studies differ in everything else: theirs is experimental with retrieval fixed and a simulated engine, ours is observational on a production engine with a comparison group of retrieved-and-uncited pages. Their agreement on this point is some evidence that it is real.

More headings, slightly. Pages with more headings are cited a little more (1.14 per unit of log heading count), which fits the same story: headings mark where one self-contained passage ends and another begins.

Structured data. Pages declaring schema.org types are cited more (1.24; pooled 1.21 with an interval just excluding one). The model does not see structured data markup in a text excerpt, so a direct effect is implausible. Two indirect routes are open. The search service may use the markup in ranking or in choosing what to excerpt, and sites that maintain structured data are, in general, more carefully built. We cannot distinguish these.

The conventions that do not show an effect. Four practices that are widely recommended for AI visibility show no association with citation in this model: an FAQ block (1.07, adjusted $p = 0.37$), a summary box (1.07, $p = 0.27$), a definition-style opening (0.94, $p = 0.24$) and headings phrased as questions (1.15, $p = 0.17$). Tables and lists also show nothing (0.99 and 0.96). H5d is not supported. This does not show that these practices are useless. The intervals are wide enough to admit small effects in either direction, the detector for FAQ blocks fires on more than half of all fetched pages and so discriminates poorly, and the sample is one engine over six weeks. It does show that, among pages competing in the same context, the ones that carry these features are not cited noticeably more than the ones that do not, while the ones with denser and more self-contained prose are.

Bylines. Pages with an author byline are cited somewhat less (0.86, adjusted $p = 0.004$; the pooled interval reaches one). We have no mechanism to offer and would not act on it. Bylines mark editorial and blog content, and the finding may simply restate the format results below.

Format. Against guides, which are the most common format, listicles are cited less (0.83), and news articles much less (0.55), with comparison-table pages and reviews not distinguishable. Raw citation rates by format were 53.3% for guides, 49.4% for listicles, 49.8% for comparison-table pages, 57.8% for landing pages and 51.6% for news, among fetched results. The listicle result sits beside the title result of Section 10.3: "best of" pages are heavily retrieved and, once retrieved, cited slightly less than explanatory guides. They remain a large share of what is cited simply because they are a large share of what is retrieved.

10.5 Reading the two tables together

The variables fall into three groups by how much they matter and how firmly.

Large and firm. Position in the context (odds roughly halving with each step up the log scale) and the encyclopedia penalty. Both hold within answer and across accounts.

Moderate and firm. A title that shares the prompt's terms; a homepage URL; a higher share of quotable paragraphs; a higher density of statistics; a shorter page; a recent year in the title. Each holds in both models with intervals excluding one.

Small, fragile or absent. The account's own site (significant within answer, not across accounts) and the account's brand in the title (small, and only just clear of one across accounts); structured data and heading count (small); listicle and news formats (small, negative); FAQ blocks, summary boxes, definition openings, question headings, tables, lists and comparison titles (no detectable association).

A publisher looking at this ordering would conclude that the work is done in two places. First, in ranking for the queries the engines write, which is conventional search optimisation directed at a new set of queries. Second, in the grain of the prose: whether the page is made of statements that can each be lifted and attributed. Page furniture comes a distant third.

10.6 Limits of Study 4

The study covers one engine, one model version, six weeks and 32 accounts, and nothing in it licenses a claim about ChatGPT or Gemini. The page-level model covers 31% of results, tilted towards pages cited elsewhere in the panel. Page features describe the page as fetched by the instrument and not the excerpt the model saw, which is unobservable. The features were computed by heuristics written for another purpose; some, like the FAQ detector, are loose. Position confounds the model's attention with the search service's relevance ranking. And nothing was randomised: a page with many quotable paragraphs differs from a page with few in ways the model does not capture. The within-answer design removes every difference between answers and none of the differences between pages.

10.7 Summary of Study 4

Of the pages Claude placed in its context, it cited 42%. Position was the dominant predictor, with the first result cited more than twice as often as the ninth. Beyond position, results whose titles matched the question, homepages, shorter pages, and pages made of dense self-contained passages were cited more, and encyclopedia articles far less. The markup conventions most often recommended for AI visibility showed no detectable association with being cited.

11. Study 5: Cited pages, issued queries and the reliability of a reading

Study 5 gathers three shorter analyses. The first looks at the fetched pages across all engines and asks what distinguishes the pages that are cited again and again. The second describes the search queries the engines write. The third measures how repeatable a reading is, and what that implies for anyone tracking visibility over time.

11.1 Pages that are cited repeatedly

Study 4 compared cited pages with uncited ones for a single engine. For the other engines no such comparison is available, but a weaker question can be asked of all of them together. Among pages that have been cited at least once, some are cited once and never again, and some are cited hundreds of times. What distinguishes them?

The data are the 12,572 fetched pages with a positive word count, on 5,572 sites. For each, the instrument records how many times it has been cited across all scans. Because pages are queued for fetching on the strength of being among the most-cited in some scan, this is not a sample of all cited pages. It is a sample of pages that were prominent at least once.

The distribution of citation counts is as skewed as everything else in the study. The median page had been cited twice. Forty-five per cent had been cited once. The 90th percentile was 15 citations, the 99th was 102, and the most-cited page had 900. The top one per cent of pages account for 26.7% of all citations to fetched pages.

Table 24 describes the pages. The typical cited page is an article of about 1,200 words with seven second-level headings and eleven numeric statistics. Four in five declare structured data. Half have what the detector takes to be an FAQ block, a third carry an author byline, and about three in ten have a table or open with a definition.

Table 24. Fetched cited pages: structure and format (n = 12,572).

Feature	Value
Median word count	1,216
Median second-level headings	7
Median numeric statistics	11
Median lists	3
Median outbound links	3
Declares structured data	80.9%
FAQ block	52.8%
Author byline	33.3%
Definition-style opening	28.5%
Has a table	28.4%
Summary or "TL;DR" box	13.1%
Format, share of pages / share of citations	
Guide	58.7% / 49.8%
Listicle	20.9% / 29.9%
Landing page	7.1% / 7.8%
Comparison-table page	6.0% / 6.6%
Review	2.6% / 1.6%
News	2.5% / 3.0%
Intent, share of citations	
Informational	64.4%
Comparison	23.2%
Transactional	10.7%

Guides are the majority of cited pages, 59%, and half of citations. Listicles are a fifth of pages and three tenths of citations: a listicle that is cited at all tends to be cited more times than a guide. By the classified intent of the page, almost two thirds of citations go to informational pages and under a quarter to comparison pages. Even in a panel whose prompts are mostly commercial, the engines' sources are mostly explanatory.

To see what goes with repeated citation, we fit the negative binomial model of equation (16) to the number of citations beyond the first, with the page-structure variables of Study 4, the format, and the log of the number of weeks between the page's first and last observed citation as an exposure control. Errors are clustered on site. Table 25 reports the incidence rate ratios.

Table 25. Incidence rate ratios for citations beyond the first, fetched cited pages. Negative binomial regression, n = 12,572, errors clustered on 5,572 sites; estimated dispersion α = 2.78. Adjusted p-values are Benjamini-Hochberg across the 19 terms.

Predictor	Incidence rate ratio (95% interval)	Adjusted p
Log of weeks between first and last citation	6.07 (5.55 to 6.64)	<0.001
Format (reference: guide)		
Landing page	1.68 (1.38 to 2.05)	<0.001
Listicle	1.30 (1.15 to 1.46)	<0.001
News	1.78 (0.94 to 3.36)	0.15
Review	1.18 (0.89 to 1.56)	0.37
Comparison-table page	0.96 (0.77 to 1.19)	0.75
Structure		
Log of heading count	1.18 (1.08 to 1.28)	<0.001
Share of paragraphs that are quotable	1.36 (1.08 to 1.70)	0.026
Log of word count	0.93 (0.88 to 0.98)	0.026
Has a table	0.88 (0.77 to 0.99)	0.095
FAQ block	0.85 (0.74 to 0.98)	0.074
Summary or "TL;DR" box	0.87 (0.74 to 1.02)	0.15
Has headings phrased as questions	0.90 (0.76 to 1.07)	0.37
Author byline	0.92 (0.81 to 1.05)	0.37
Definition-style opening	0.95 (0.86 to 1.05)	0.45
Declares structured data	0.95 (0.80 to 1.13)	0.64
Log of list count	0.98 (0.93 to 1.03)	0.55
Statistics per 1,000 words (log)	1.00 (0.96 to 1.04)	0.98

The exposure term dominates, as it must: a page that has been in view for longer has had more chances. With that held fixed, four structural results survive correction, and three of them repeat what Study 4 found by an entirely different route.

Quotable paragraphs. A page whose paragraphs are all quotable is cited 36% more often than one with none (1.36, interval 1.08 to 1.70). In Study 4 the same variable had the largest page-level odds ratio for being cited by Claude given retrieval.

Length. Longer pages are cited again slightly less (0.93 per unit of log word count). Study 4 found 0.83.

Headings. Pages with more headings are cited again more (1.18). Study 4 found 1.14.

Format. Here the two studies differ, and the difference is instructive. In Study 4, listicles were cited *less* than guides once retrieved (0.83). Here listicles are cited again *more* than guides (1.30), and landing pages more still (1.68). The two are measuring different stages of equation (4). Repeated citation across the panel reflects, above all, how often a page is retrieved: a "best tools for X" page ranks for the many variants of that question that different accounts track, and a product's landing page is retrieved whenever the product is searched for. Given retrieval, a guide is slightly more likely to be cited than a listicle. A listicle's advantage is at the retrieval stage, and it is large enough to outweigh its small disadvantage at the citation stage.

The conventions that showed nothing in Study 4 show nothing or slightly less than nothing here. FAQ blocks (0.85), summary boxes (0.87), tables (0.88), question headings (0.90) and definition openings (0.95) all have point estimates below one, none significant after correction. Statistics density, which was positive in Study 4, is flat here (1.00). Structured data is flat (0.95), which weakens the case that its positive coefficient in Study 4 reflected anything about the page.

Taken with Study 4, the reading is consistent. What recurs across both analyses, on different outcomes with different comparison groups, is that pages made of short self-contained passages under clear headings, and not padded for length, are cited more. What does not recur is any benefit from the labelled blocks.

The limits are the same as before with one addition. This model has no uncited pages in it. It describes intensity among the cited and prominent, and a feature could be associated with being cited often while being negatively associated with being cited at all.

11.2 The queries engines write

On one scan of one account on 14 November 2026 the instrument stored the search queries reported by the three engines whose APIs expose them: 1,241 queries for 384 answers. The sample is a single market on a single day, so only the form of the queries is reported and nothing about their content. Table 26 summarises it.

Table 26. Search queries issued by three engines on one scan (384 answers, 1,241 queries). "Prompt terms kept" is the share of the prompt's content words that appear in the query; "new terms" is the share of the query's content words that are not in the prompt.

	ChatGPT	Gemini	Claude
Answers with logged queries	101	142	141
Queries	502	560	179
Queries per answer, mean	5.0	3.9	1.3
Queries per answer, maximum	13	28	3
Answers with a single query	0%	0%	74%
Median words in prompt	9	10	9
Median words in query	8	6	7
Prompt terms kept	40%	44%	71%
New terms in query	59%	39%	21%
Query identical to the prompt	0.0%	3.0%	0.6%
Query adds a year the prompt lacks	35.9%	0.2%	30.7%
Query adds "best" or "top"	4.8%	4.6%	6.1%
Query adds a comparison or review term	8.2%	2.7%	1.1%

Three things are worth drawing out.

The engines do not search for what the user asked. The query was identical to the prompt in 0.0%, 3.0% and 0.6% of cases. Claude stays closest, keeping 71% of the prompt's content words and adding few of its own. ChatGPT and Gemini keep only about two fifths of the prompt's words. For ChatGPT, three fifths of the words in a query were not in the prompt at all. This is the first factor of equation (4) at work: the retriever is conditioned on a query the model composed, and a page that would rank well for the user's literal words may be invisible to the searches actually run.

ChatGPT and Gemini fan out. Neither ever issued a single query. ChatGPT averaged five per answer and Gemini four, with a maximum of 28. (ChatGPT was limited to one search call, and a single call can carry several queries.) One question becomes several narrower searches, each returning its own results, which are pooled in the context. A page can enter the context by matching any one of them. This is also part of why engines' sources overlap so little (Section 9.4): even on a shared index, five queries written by one model and four written by another retrieve different things.

ChatGPT and Claude add the year. In 36% of ChatGPT's queries and 31% of Claude's, the query contained a year that the prompt did not. The interval for ChatGPT is 31.8% to 40.1%, and for Claude 24.4% to 37.8%. Gemini almost never did (0.2%), presumably because its grounding is integrated with a search engine that handles freshness itself. The models have evidently learned that appending the current year is how one asks a search engine for recent material. The consequence for publishers connects directly to Study 4, where a recent year in a result's title was associated with 21% higher odds of citation: a page whose title carries the current year is favoured twice, once in ranking for a query that contains the year and again when the model chooses among the results.

The engines seldom add "best" or "top" (5% to 6%) when the prompt lacks them, and ChatGPT adds a comparison or review term to 8% of queries.

These descriptions come from a few hundred answers in one market and should be treated as a sketch. The same analysis on a broad sample would be straightforward once the instrument has logged queries for longer.

11.3 How repeatable is a reading

All of the above concerns averages over many answers. A person tracking their own visibility looks at something much smaller: a set of prompts, read once, compared with the same set read a month ago. Whether a difference between two such readings means anything depends on how much a reading varies when nothing has changed.

For each prompt and engine with at least two readings in the analysis set, we paired each reading with the next one. There are 34,144 such pairs on unbranded prompts and 1,548 on branded prompts. Table 27 gives the probability that the brand is named in the second reading, given whether it was named in the first.

Table 27. Test-retest of the mention indicator across consecutive readings of the same prompt by the same engine. κ is Cohen's kappa with a 95% cluster bootstrap interval over accounts.

Engine	Pairs	Mention rate	Named again, given named	Named, given not named	κ
Unbranded prompts					
Claude	5,399	2.14%	59.0% (of 105)	1.21%	0.53 (0.36 to 0.63)
Gemini	4,410	1.38%	52.5% (of 61)	0.67%	0.52 (0.27 to 0.66)
Perplexity	6,762	1.79%	47.7% (of 128)	0.80%	0.50 (0.38 to 0.67)
Grok	6,606	0.38%	40.7% (of 27)	0.18%	0.44 (0.17 to 0.62)
Google AI Overview	4,198	0.92%	14.7% (of 34)	0.91%	0.12 (0.05 to 0.22)
ChatGPT	6,674	0.28%	11.8% (of 17)	0.29%	0.10 (0.00 to 0.25)
All engines	**34,144**	**1.11%**	**46.4% (of 373)**	**0.64%**	**0.45 (0.34 to 0.57)**
Branded prompts					
All engines	1,548	91.4%	95.2% (of 1,416)	50.8% (of 132)	0.44 (0.29 to 0.62)

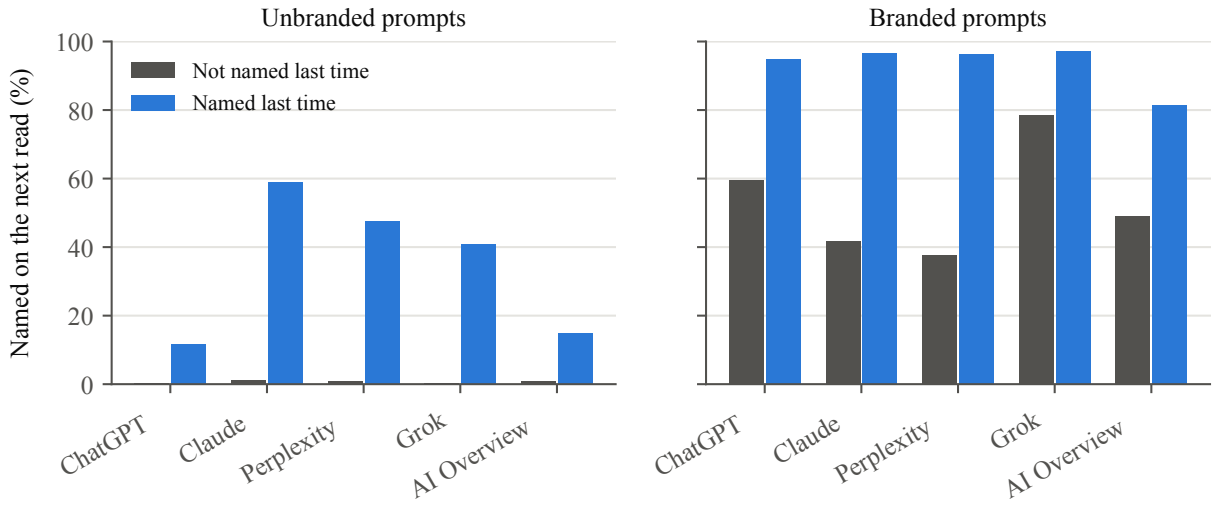


Figure 10. Probability that a brand is named on the next reading of the same prompt by the same engine, by whether it was named on the previous reading.

On unbranded prompts a mention is a coin toss on the next reading. Across all engines, a brand that was named in the answer to an unbranded prompt was named again in the next reading of that prompt by that engine 46.4% of the time. More than half the time, the mention was gone. H6a is supported.

The figure varies by engine and is lowest where it matters most. For Claude, Gemini and Perplexity, about half of mentions recur. For Google AI Overview it is 15%, and for ChatGPT 12%: a mention on those engines is almost entirely unrepeatable, barely more likely on the reading after a mention than on any other. The numbers of named-first pairs for those two engines are small, 34 and 17, so the estimates are imprecise, but the κ values of 0.12 and 0.10 with intervals that reach or approach zero tell the same story.

Raw agreement between consecutive readings is 98.8%, which sounds reassuring and is not, because with a mention rate near 1% two readings agree on "not named" almost all the time by chance. Cohen's κ, which removes chance agreement, is 0.45 overall. By the conventional verbal scale that is "moderate" agreement, and for a measurement that people use to judge whether their marketing is working, moderate agreement between two readings days or weeks apart is weak.

Branded prompts are more stable in the usual direction and no more reliable overall. A brand named in the answer to a branded prompt is named again 95.2% of the time. But when a branded prompt fails to produce the name, the next reading produces it half the time (50.8%), and κ is 0.44, no better than for unbranded prompts. H6b is supported in the sense that matters to a user, since the common outcome is highly repeatable, and not in the sense of chance-corrected agreement.

The intraclass correlations tell the same story from another angle. For unbranded prompts, the share of variance in the mention indicator that lies between prompts, as opposed to between readings of the same prompt, is 0.50 for Perplexity and Grok, 0.44 for Gemini, 0.33 for Claude, 0.29 for ChatGPT and 0.09 for Google AI Overview. Even for the most consistent engines, half the variance in whether a brand is named is variance between one reading and the next of the very same prompt.

Table 28 looks at the prompts themselves.

Table 28. Unbranded prompts read at least twice by an engine: how consistently the brand was named.

Engine	Prompts	Readings per prompt	Never named	Sometimes named	Always named
Perplexity	2,258	4.0	95.3%	3.5%	1.2%
Claude	1,375	4.9	94.4%	4.9%	0.7%
Google AI Overview	835	6.0	94.9%	5.1%	0.0%
Gemini	1,599	3.8	96.6%	2.7%	0.8%
ChatGPT	2,212	4.0	98.6%	1.3%	0.1%
Grok	2,108	4.1	98.7%	0.9%	0.3%

For every engine, the prompts on which a brand is sometimes named outnumber those on which it is always named, usually by a wide margin: 5.1% against none for AI Overview, 1.3% against 0.1% for ChatGPT. Reliable, repeatable unprompted naming of a small brand is close to non-existent in this panel. What exists is a small set of prompts on which the brand has some chance of appearing on any given reading.

This changes what the object of measurement should be taken to be. A brand does not "rank" for a prompt on an answer engine, in the sense of holding a position that can be read off. For each prompt and engine there is a probability that the brand is named, and a single reading is one draw from it. Most of those probabilities are zero. A few are somewhere between zero and one. Almost none are one.

11.4 What this implies for sample size

Suppose a brand tracks m unbranded prompts on one engine and reads each r times, and reports the share of readings in which it was named. How precise is that share?

Equation (7) gives the answer in principle: the variance is that of a simple proportion inflated by $1 + (r - 1)\rho$. With intraclass correlations of 0.3 to 0.5, three readings of a prompt are worth about 1.5 to 1.9 independent readings, and five are worth about 1.7 to 2.3. Repeating a reading helps, with quickly diminishing returns. Adding prompts helps more.

Because the rates involved are so small that the normal approximation behind such formulas is unreliable, Table 29 reports a direct simulation, described in Section 6.10, using the observed mention rates of the 3,497 unbranded prompt-and-engine pairs with at least five readings. The mean rate in that pool is 0.59%.

Table 29. Simulated sampling error of a tracked unbranded mention rate whose true value is 0.59%, by number of prompts and readings per prompt. Half-width is 1.96 standard deviations of the simulated rate.

Prompts	Readings each	Standard deviation	95% half-width	Half-width relative to the rate
25	1	1.55 pp	±3.0 pp	5.2×
25	5	0.96 pp	±1.9 pp	3.2×
50	1	1.07 pp	±2.1 pp	3.6×
50	5	0.67 pp	±1.3 pp	2.2×
100	1	0.79 pp	±1.5 pp	2.6×
100	3	0.54 pp	±1.1 pp	1.8×
100	5	0.47 pp	±0.9 pp	1.6×
200	1	0.54 pp	±1.1 pp	1.8×
200	5	0.34 pp	±0.7 pp	1.1×

The table is sobering. A set of one hundred unbranded prompts read once, which is a larger and more disciplined programme than most brands run, measures a true rate of 0.6% with a 95% half-width of 1.5 percentage points. The uncertainty is two and a half times the quantity being measured. An observed rate of 0%, 1% or 2% are all unremarkable outcomes when the truth is 0.6%, and a movement from 1% to 2% between two months, which a dashboard would display as a doubling, is well inside the noise. Even two hundred prompts read five times each, a thousand readings, leave a half-width as large as the rate.

Three practical conclusions follow.

First, **changes in an unbranded mention rate at the scale of a single account are mostly not detectable.** To distinguish a rate of 0.6% from a rate of 1.2%, a doubling, with the usual 80% power at the 5% level requires several thousand readings on each side, more once the within-prompt correlation is accounted for. That is beyond what an individual brand's tracking produces in a month.

Second, **the readings should be pooled across engines and over time before they are interpreted,** and reported with an interval. A rolling share over several scans is a far better object than a single scan's figure. The interval should come from the cluster bootstrap of Section 6.2 with prompts as the clusters, and not from the binomial formula.

Third, **quantities with higher base rates are measurable where the mention rate is not.** Whether the brand's own site is among an answer's sources occurs in about 7% of sourced answers to unbranded prompts, roughly twice the rate at which the brand is named in those answers, and the share of answers in which any one of a set of tracked competitors is named will be higher still. A programme that wants to detect change should watch the outcomes that happen often enough to estimate.

The same logic applies to the sources. Section 9.5 found that the same engine reading the same prompt on the same day shares about a third of its cited sites with its previous reading. A single reading of which sites an engine cites for a prompt is, like a single reading of whom it names, one draw.

11.5 Summary of Study 5

Among cited pages, those made of short self-contained passages under clear headings, and not padded for length, are cited more often; listicles and landing pages are cited repeatedly because they are retrieved for many queries, although, once retrieved, a guide is slightly more likely to be cited. The engines search for something other than what the user typed: ChatGPT and Gemini split a prompt into four or five queries sharing two fifths of its words, and ChatGPT and Claude add the current year to about a third of their queries. And a single reading is a poor measurement. An unprompted mention recurs on the next reading less than half the time, and for two of the engines hardly at all, so that a tracked mention rate for one brand on one hundred prompts has an uncertainty larger than the rate itself.

12. Discussion

The five studies were reported separately because they use different data and different designs. This section draws them together into a single account of how a brand comes to be named and a page comes to be cited, sets that account against the prior literature, and considers what it does and does not explain.

12.1 A two-gate, two-source account

Section 2 argued from the mechanism that a name in an answer can come only from the model's parameters or from its context, and that a page affects an answer only by passing two gates: being retrieved, and being used. The results fit that structure closely enough to write it down.

Let q be a prompt, e an engine and b a brand. Write $\sigma(q, e)$ for the probability that the engine searches, $\pi_0(b \mid q, e)$ for the probability that the brand is named in an answer written from parameters alone, $\rho(b \mid q, e)$ for the probability that material presenting the brand reaches the context given a search, and π_1 and π_1^{+} for the probabilities that the brand is named in a searched answer without and with such material. Then

$$\Pr(\text{named}) \;=\; (1-\sigma)\,\pi_0 \;+\; \sigma\Big[(1-\rho)\,\pi_1 \;+\; \rho\,\pi_1^{+}\Big]. \tag{25}$$

Each term was estimated, for some population, somewhere in the paper.

The search probability σ is a property of the engine and the prompt (Study 1). It is one for Perplexity and AI Overview, zero for Grok after August, and for ChatGPT it ranges with the kind of question, with comparison prompts having about five times the odds of a search that informational prompts have.

The parametric naming probability π₀ is, for a small brand on an unbranded prompt, close to zero. Grok without search named the tracked brand in 0.70% of answers. On the category sets, the non-searching engine's naming is confined to an effective fifteen or so brands per category, and nine in ten of the brands in a category's universe were never named by it at all.

The retrieval probability ρ is small for a brand's own site, which was among the sources in 6.8% of sourced answers to unbranded prompts, and is governed by ranking for queries the model writes, which share as little as two fifths of their words with the prompt (Study 5).

The naming probabilities given search, π₁ and π₁⁺, differ by roughly two orders of magnitude for small brands: 0.58% without the brand's own site among the sources and 39% with it (Study 3). On the category sets, for brands with no prior, the corresponding figures were 0.09% and 7.1%, and for brands with strong priors 33% and 76% (Study 2).

Equation (25) makes the structure of the problem plain. For a well-known brand every term is favourable, and the brand is named whether or not the engine searches. For an unknown brand π_0 and π_1 are both near zero, so that

$$\Pr(\text{named}) \;\approx\; \sigma \cdot \rho \cdot \pi_1^{+}, \tag{26}$$

a product of three probabilities, each of which must be non-negligible. The engine must search; material about the brand must be among what is retrieved; and the model must then choose to name the brand. With illustrative values from the panel of 0.75, 0.07 and 0.39, the product is 2%, which is the order of the unbranded mention rates in Table 6. The agreement is not a test, since the inputs come from the same data, but it shows that the pieces are of the right size to account for the whole.

The same decomposition explains the Grok result. Setting σ to zero removes the second term of equation (25) entirely and leaves π_0. Grok's rate fell from 1.90% to 0.70%. What remained is the parametric rate for the brands in the panel; what was lost is the part that flowed through retrieval.

For citations the corresponding structure is simpler, since a page cannot be cited from parameters by any of these engines:

$$\Pr(\text{page cited}) \;=\; \sigma \cdot \Pr(\text{retrieved} \mid \text{search}) \cdot \Pr(\text{cited} \mid \text{retrieved}). \tag{27}$$

Study 4 estimated the last factor for one engine at 42% on average, 68% in the first position and 32% in the ninth. The middle factor is not observed for any page we could name in advance, and is the largest unknown in the paper.

12.2 The prior is the larger part

The most consequential quantitative finding is that the parametric prior accounts for most of who is named even when the engine searches. On the category sets, one non-searching model's naming frequencies predicted the brands named by five different searching engines with a pseudo-R^2 of 0.52, and adding what was actually retrieved raised that to 0.55.

Three readings of this are possible, and they are not exclusive.

The model writes the answer it would have written and decorates it with sources. On this reading the generator's parameters largely determine the list, and retrieval supplies details and citations. The knowledge-conflict literature gives it some support: models favour the parts of the context that agree with what they already hold [18], and the brands they already hold are the ones named.

The retriever and the prior agree. The search service ranks the pages of well-known brands, and third-party pages that feature well-known brands, for category queries. What reaches the context is therefore mostly about the same brands the parameters favour, and the two sources say the same thing. On this reading the prior "explains" naming partly because it is a good proxy for what will be retrieved. The heavy overlap between the two sets of predictors in Table 12, where sources alone reach 0.23 and add only 0.03 to the prior, is what this would produce.

Both reflect the same underlying fact about the world. A model's parameters and a search engine's index are two summaries of the same body of text. A company that is written about a great deal is frequent in training data and well represented in search results for the same reason. The prior and the retrieved set are correlated because they have a common cause, and neither needs to cause the other.

The data cannot separate these, and the third is surely part of the truth. What the data do establish is the practical consequence, which is the same under all three: for a brand that the models do not already associate with its category, being retrieved is a far weaker lever than being known,

and a brand with a strong prior is named a third of the time without any of its own material in the context.

This places a boundary on the claims of the optimisation literature. Aggarwal et al. showed that rewriting a retrieved source can raise its share of the answer by up to 40% [1]. That is an effect on $\Pr(\text{cited} \mid \text{retrieved})$ and on the prominence of a source's content. Our results suggest that for the outcome a brand cares about most, being named, the room for such effects is bounded by the terms in front of them in equation (26), and above all by a prior that no change to a page can alter until the next model is trained. Puerto et al.'s finding that most such methods are ineffective in a more realistic benchmark [27] is consistent with a world in which the generator's choices are substantially fixed before the context is read.

12.3 Where retrieval does matter

It would be wrong to conclude that retrieval is unimportant. Three results show where it carries the weight.

First, for brands with no prior it is the only route. In relative terms the association between a brand's own site being among the sources and the brand being named was largest exactly for brands the non-searching engine never names: 80 to one. An unknown brand is not named without retrieval, in effect ever, and is named one time in fourteen with it.

Second, removing retrieval from an engine cut its unprompted naming of small brands by more than half and, in the three accounts observed on both sides of the change, to zero in 2,742 answers. The brands in the panel are overwhelmingly small. For them, an engine that does not search is an engine that does not name them.

Third, the kind of material retrieved matters, and the kind that matters most is not the brand's own. A brand appearing in the title of a retrieved page had odds of being named about twelve times higher at a given prior, against three times for the brand's own site, and such titles also moved the brand earlier in the answer. A third-party page that lists or compares the brand hands the model both the name and a reason to include it. Chen et al.'s observation that AI engines lean on earned media [28] has, in this, a mechanism: it is not only that such pages are cited, but that being featured in them is what gets a brand into the prose.

There is a tension here that deserves stating. Study 4 found that listicles and comparison pages are cited slightly less than guides once retrieved, and that list-style titles confer no advantage in citation. Study 2 found that a brand's presence in a retrieved page's title is the strongest source-side predictor of the brand being *named*. These are different outcomes. A "ten best" page may not be credited with a citation and may nonetheless be where the model found the ten names. Citation and mention come apart on the supply side just as they do on the brand's own side.

12.4 Cited is not named

The loosest coupling in the data is between the two outcomes that visibility tools usually report side by side. In 61% of the answers that cited an account's own site in response to an unbranded prompt, the account was not named. Conversely, on the category sets a brand with a strong prior was named in a third of the sourced answers that cited nothing of its own.

The reason is in what a citation is. It attaches a source to a claim. If a company publishes a clear explanation of a concept in its field, an engine may use that explanation and cite it, in an answer about the concept, without any occasion to mention who wrote it. The page has been useful. The brand has not been recommended. The cited URLs bear this out: where the brand went unnamed, the cited page was an article or guide in 58% of cases; where the brand was named, it was more often the homepage or another page about the company itself.

For measurement, the lesson is that the two outcomes need separate treatment and separate goals. For practice, it is that content marketing of the usual kind, informative articles on topics near one's product, is a way of being cited and not, in the main, a way of being named.

12.5 The engines are not one thing

The differences between engines recur across every study and are large enough that statements about "AI search" in general should be treated with suspicion, including some in this paper.

The engines differ in whether they search (from never to always), in how they decide (ChatGPT by the type of question, Gemini almost indiscriminately), in how many queries they issue (one for Claude, four or five for ChatGPT and Gemini), in how many sources they return (a median of five to ten), in what they cite (AI Overview gives 15.5% of its citations to social and community platforms, Claude 1.5%), in how concentrated their citations are (ten sites take a quarter of AI Overview's citations and a twenty-fifth of Gemini's), and in how repeatable they are (a mention recurs 59% of the time on Claude and 12% on ChatGPT). For the same prompt at the same moment, two engines' sources share a tenth of their combined sites.

Even the one regularity that held across engines has an asymmetry in it. The category prior predicted naming with an odds ratio near two for every engine, a striking uniformity. But the weight on a brand's own site ranged from 1.6, indistinguishable from one, for AI Overview to 8.6 for ChatGPT. The engines appear to share a picture of which brands belong to which category and to differ in how they use what they retrieve.

The obvious implication, for research as for practice, is that single-engine results do not generalise. Study 4 is about Claude. It may describe ChatGPT. Nothing here shows that it does.

12.6 An answer is a draw

The reliability results change what kind of object an AI answer should be taken to be. A search results page was, to a first approximation, a stable thing: the same query returned the same ten links to most users for days at a time, and "we rank third" was a meaningful sentence. An answer from one of these engines is a sample from a distribution that is wide in every dimension we measured. For the same prompt and the same engine: about ten days apart, between a fifth and a third of cited sites recur; from one month to the next, between a third and three fifths of named brands recur; from one reading to the next, an unprompted mention of a small brand recurs less than half the time. On the same day, with nothing in the world having changed, cited sites recur at the same rate of about a third.

Two sources of randomness compound. The generator samples its tokens. And, for engines that search, the retrieved set is itself a fresh draw on each request, from a query the model has just rewritten in slightly different words. Study 2 found a wrinkle that complicates the simple expectation that search adds noise: for ChatGPT and Claude, sets of named brands were more stable across months when both answers had searched than when neither had. Retrieval anchors an answer to the lists that are out there, and an unanchored answer to an open-ended question varies more freely. Search makes the sources unstable and the names somewhat steadier.

The implication for anyone reporting on AI visibility is that the unit of evidence is not an answer but a distribution, and that claims of change require the kind of sample sizes computed in Section 11.4. The implication for the literature is that single-shot observational studies of what engines cite, however many queries they cover, describe one draw per query, and their per-query results would not replicate if run again the following week. Their aggregates are more robust, since randomness averages out across thousands of queries, but any analysis that conditions on what happened for a particular query is conditioning on noise.

12.7 What the artefacts teach

Five times during the analysis a property of the stored data distorted the results, and three times it produced a finding that was statistically overwhelming and false. Each time the false finding had a ready interpretation. Truncated answers suggested that search suppresses mentions. Source chips suggested that ChatGPT names every brand it cites. A strict matcher suggested that search makes the difference between naming and not naming a brand the user had asked about by name. The other two, a matching rule that changed during the period and onboarding audits stored alongside scans, would have shifted every rate without announcing themselves.

The common thread is that in this domain the measurement interacts with the treatment. Whether an engine searched changes the length of its answer, the form in which it writes a name, the markup it attaches, and the chance that the read fails; a detector tuned on one condition will perform differently in the other, and the difference will appear as an effect. Constant measurement error is harmless to a comparison only if it is constant across the conditions compared, and here it rarely is.

We suspect that these artefacts are common in the grey literature on AI visibility, which is produced largely by vendors of tracking tools from data like ours, with matching rules like our first attempts, and without the opportunity that a paper provides to be embarrassed by them. A reader of any such study, including this one, should ask how mentions were detected, whether the detector was validated separately on grounded and ungrounded answers, whether failed reads were distinguished from negative results, and whether one code path produced all the data.

12.8 Relation to conventional search optimisation

It is sometimes said that optimisation for AI engines is a new discipline with little relation to the old one, and sometimes that it is the old one under a new name. The results support a more specific position.

For citation, the continuity is strong. The largest measurable determinant of being cited, given a search, is rank in the results for the model's query. That is search optimisation. What has changed is the query: it is written by a model, it is often several queries, it frequently carries the current year, and it shares as little as two fifths of its words with what the person asked. Optimising for the user's literal phrasing is optimising for a query that may never be issued.

For the page itself, the results favour properties that good writing for the web has always had, short self-contained passages, clear headings, concrete figures, economy of length, and offer no support for the newer conventions of FAQ blocks, summary boxes and question headings. Aggarwal et al. found that keyword stuffing does not help [1]; we find that the furniture recommended in its place does not obviously help either.

For naming, the continuity is weak. Nothing in conventional search optimisation corresponds to the prior. The nearest analogue is brand awareness, and the means of building it, being written about by others over a period of years, is public relations and not search optimisation. The finding that third-party pages featuring a brand outweigh the brand's own pages in predicting a mention points in the same direction.

12.9 What remains unexplained

Much does. The models in Table 12 leave almost half the deviance unexplained, and that is with a predictor, the prompt-level prior, that is itself a noisy measurement from three readings. The retrieval step is nearly invisible to us: we never observe a search service's full ranking, never observe the excerpt, and observe the queries for one scan. We do not know why ChatGPT is so much more likely than the other engines to name a brand whose site it cites, or why AI Overview is so unlikely to repeat itself. We cannot say how much of the position gradient is the model's and how much the search engine's. And we have measured a prior without being able to say where it comes from, beyond the general fact that it must reflect training text.

The mechanism of Section 2 says what kinds of answer these questions can have. The data of this paper say how large the effects are in one panel. Closing the gap between them requires experiments, and the two that would do the most are described in Section 15.

13. Threats to validity

The limits of the study have been noted as they arose. This section collects them under the conventional headings, says which findings each one threatens, and says what, if anything, was done about it.

13.1 Construct validity: are we measuring what we say?

"**With sources**" **is a proxy for** "**searched**"**.** The indicator S_i records that the API response carried at least one source. An engine can search and return no citation annotations, and a read can fail in a way that drops the sources. For ChatGPT in particular, a substantial group of unsourced answers were billed about 5,700 tokens, far more than a prompt and a tool description account for, which suggests that some of them searched without citing or spent heavily on hidden reasoning. If searched-but-uncited answers are classed as unsourced, the contrast between sourced and unsourced answers is diluted and the within-prompt effect of Section 7.5 is understated. The Grok design is immune, since there the tool was removed.

Mentions are detected by string matching against a label derived from a domain. The final rule recovers names written as words and still misses brands whose trading name differs from their domain, names with an ampersand or abbreviation, and misspellings. Absolute mention rates are lower bounds. More seriously, as the fourth artefact showed, detection can differ between conditions. We corrected the one such difference we found. There may be others. The category data of Study 2, where brands were identified by an extraction model that resolves surface forms to canonical records, are not exposed to this, and their agreement with the panel on the direction of every shared finding is the main reassurance.

The extraction step in the category data is itself a model. It can miss a brand, attribute a mention to the wrong canonical record, or merge two brands that share a domain. We did not audit it against human annotation for this paper.

"**Cited**" **means different things across engines.** For Perplexity it is the list of results retrieved; for Claude before 30 September it included results read and not cited; for ChatGPT it is annotated passages only. Cross-engine comparisons of citation counts and concentration in Study 3 compare what each API reports, and part of the difference between engines is a difference in reporting.

Page features describe the page as we fetched it, not the excerpt the model saw. This is unavoidable. It means the coefficients of Study 4 estimate the association between citation and properties of whole pages, attenuated by however weakly the page predicts its excerpt. A feature that matters through the excerpt will appear weaker than it is.

The feature detectors are heuristics. They were written for a product feature and not validated for research use. The FAQ detector fires on more than half of fetched pages. A null result for a loosely measured feature is weak evidence of no effect.

The prior is one model's behaviour. We measured each brand's prior as Grok's naming frequency without search. Its strong prediction of four other vendors' models suggests it captures something shared, but it is not a measure of training-data frequency and it carries Grok's idiosyncrasies. A prior averaged over several non-searching models would be better and was not available.

13.2 Internal validity: are the associations what they seem?

Nothing was randomised. Every estimate except the token differences is an association or rests on a stated assumption. The assumption-dependent ones are these.

The Grok difference in differences assumes parallel trends. The design has an unusually strong basis for it, since treated and control observations are the same prompts in the same scans, and an unusually weak feature, since the accounts before and after are almost entirely different. The estimate identified from the three continuing accounts (-1.7 points) is half the pooled estimate (-3.6 points), and the difference between them is the contribution of the change in account mix interacting with engine. The safest statement is the range: Grok's unprompted naming of small brands fell by between 1.2 points (its own change) and 3.6 points (relative to controls), from a base of 1.9%.

The control engines were not static. ChatGPT's model changed on 15 July, inside the before window, and all vendors update models and search back ends without notice. If a control engine became more inclined to name small brands during the after window for reasons of its own, the pooled estimate overstates the effect.

One treated unit. There is one Grok. A model-specific shock coinciding with mid-August would be indistinguishable from the effect of the tool's removal. The sharp drop of the sourced share to exactly zero over one week, and the simultaneous halving of answer length, tie the change to the tool, but the inference is about this engine at this time.

Within-prompt contrasts (Sections 7.5 and 9.6) hold the prompt, brand and engine fixed and compare readings. They are confounded by whatever differs between readings: the date, the state of the web, the query the model happened to write, and the model's own choice to search or not. The direction of the resulting bias is arguable in each case and was discussed there.

Position in Study 4 confounds the model's attention with the search service's ranking by relevance. We have reported it as their joint association with citation and have not claimed either component.

Selection into the fetched-page sample. Pages are fetched because they were cited often. The page-level model of Study 4 therefore runs on 31% of results, over-representing cited ones. Conditioning on the answer removes level differences and cannot remove an interaction between

popularity and the effect of a feature.

Reverse causation in the source variables. In Studies 2 and 3 a brand's site or name among the sources is treated as a predictor of the brand being named. The causation can run the other way within a single request: a model that is going to write about a brand may issue a query that retrieves the brand's site. The prompt-level prior in Study 2 was included partly to absorb this.

Analytic flexibility. The analysis was not preregistered, the hypotheses were written with results in hand, and five rounds of data cleaning each changed the numbers. Each cleaning step was motivated by a defect found by inspection and not by the direction of a result, and two of them weakened or removed findings we would have been glad to report. But a reader is entitled to discount accordingly, and the fact that the conclusions moved as much as they did under cleaning is itself a caution about how much they might move under further scrutiny.

Multiple comparisons. Within the page-feature models p-values are corrected. Across the paper they are not. Several dozen hypotheses were examined, and a few nominally significant results among the secondary findings are to be expected by chance. The findings we rest weight on have p-values or interval separations that leave a wide margin.

13.3 External validity: to what do the results generalise?

The prompts are not a sample of what people ask. They were chosen by paying customers of one tracking product to monitor their own markets. They are commercial, in English, predominantly about business software and professional or local services, and short. The category sets are four software categories and two agency categories. Nothing here speaks to health, news, politics, coding, or casual questions, and the profile of cited sources in particular (Section 9.2) would look very different elsewhere, as the comparison with Zhang et al. on Gemini and Wikipedia showed.

The brands are small. The accounts are those of companies that felt the need of a visibility tool, which selects against the well known. The headline rates of unprompted naming, around 1% to 4%, are rates for small brands on prompts they chose. The category data cover the full range of brand size and were used for every claim about how prior and retrieval combine.

The engines were accessed by API in particular configurations. Our ChatGPT is a model with one search call allowed and no system prompt. The consumer product differs in ways that plausibly matter for every outcome studied: more searches, a long system prompt, memory, location. The direction of the difference is unknown. We would expect the consumer products to search more and to retrieve more, which by the argument of Section 12.1 would give retrieval more weight than we observed.

One country, one language, no personalisation. All requests originated from United States servers in English with no user context.

Five months, and the systems are changing. The data run from June to November 2026. During that time one engine's model was replaced, one engine was dropped and another added, one engine's search tool was withdrawn, and the instrument's own clients were revised several times. Any numerical result here has a short shelf life. The structural results, that there are two sources and two gates and that the prior dominates, depend on how these systems are built and should outlast the specific models.

Study 4 is one engine. Claude is the only engine for which retrieved and cited results are separately observable in our data. Its citation behaviour may or may not resemble the others'.

The panel is concentrated. Two accounts supply 30% of answers and ten supply 61%; in the after period of the Grok analysis one account supplies a third of all control-engine mentions. Cluster-robust intervals reflect this and are wide. Results were re-estimated without the two largest accounts where the design allowed, and held. A panel of several hundred comparably sized accounts would support much firmer conclusions.

13.4 Statistical conclusion validity

Few clusters in places. Study 4 rests on 32 accounts, the Claude search-decision model on 58, the continuing-accounts estimate on three. Cluster-robust standard errors are biased downward with few clusters. Where an estimate was significant within strata and not with account-clustered errors, as for the own-site term in Table 22, we have said so and not relied on it.

Rare outcomes. Unprompted mentions occur in 1% to 4% of answers. Models of them are driven by a few hundred events, and within-prompt designs by a dozen or so informative prompts. The sign tests in Table 9 are exact, and their power is low.

Model form. The logistic models assume effects that are linear in the log-odds and, apart from Table 13, additive. Table 13 shows directly that the association of the own-site variable with naming varies sixfold in absolute terms and more than thirtyfold in relative terms across prior bands, so the single coefficient in Table 12 is an average over a strong interaction.

Concentration measures are sample-size dependent, and the engines differ eightfold in citation counts. Differences in Gini or effective number between engines partly reflect that.

13.5 Conflict of interest

The author founded and runs the company whose product collected the data, and that company sells visibility tracking and content services. Three findings cut against the commercial interest of such a company and are reported without softening: that a single reading is too noisy to detect most changes at the scale of one account; that the page conventions commonly sold as AI optimisation show no detectable association with citation; and that a brand's own content is cited far

more readily than the brand is named. The two largest accounts in the panel are the author's own, and the main result was re-estimated without them. The raw data cannot be shared, which prevents independent re-analysis; the code and the aggregates are available on request.

13.6 What would most change our conclusions

If an independent panel with broad prompts and many comparably sized accounts found that the prior explained much less than half of naming in sourced answers, the central claim of Section 12.2 would fall. If a randomised context-injection experiment found that planting a page about an unknown brand in the context produced a naming rate far above the 7% we observe, it would show that our observational estimate is depressed by confounding with relevance, and that retrieval is a stronger lever than this paper suggests. If the consumer products were shown to search for nearly every commercial query, the practical weight of σ in equation (25) would shrink to nothing. Each of these is checkable, and Section 15 says how.

14. Practical implications

The paper's purpose is explanatory, but the questions it addresses are ones on which people are spending money, and it would be evasive not to say what the results suggest for practice. The statements below are graded by how firmly the evidence supports them. None is a guarantee, and all are subject to the limits of Section 13: one panel, small brands, commercial prompts, API configurations, five months.

14.1 For a brand that wants to be named

Establish first whether the engines search for the questions that matter to you. The route from a web page to an answer exists only when the engine searches. For ChatGPT that depends on the question: comparison and recommendation questions, superlatives and price terms mostly trigger a search, and definitional questions mostly do not. For a question the engine answers from memory, nothing published today will change the answer until a later model is trained on text that includes it. A brand's list of target questions should be divided accordingly, because the two halves call for different work on different timescales. (Firm, for the engines and configurations studied.)

Know which case you are in. The evidence divides brands sharply by whether the models already associate them with their category. A brand with a strong prior is named in a third of searched answers with nothing of its own retrieved, and in three quarters when its site is among the sources. A brand with no prior is named in one searched answer in a thousand without its site among the sources and one in fourteen with it. The first kind of brand is protecting a position. The second is trying to enter one, and for it retrieval is the only route and an inefficient one. A brand can find out which it is by asking an engine with search disabled, repeatedly, the category questions it cares about, and counting. (Firm as description; the measurement is cheap to make.)

Expect third-party coverage to matter more than your own pages. At a given prior, a brand appearing in the title of a retrieved page was associated with about twelve times the odds of being named, against about three times for the brand's own site being retrieved, and for one engine the brand's own site showed no association at all. Pages that list, compare or review a set of products present a brand to the model as a candidate with reasons attached. The implication is that effort spent on being included in the independent comparisons, reviews and round-ups that rank for a category's questions is better directed, for the purpose of being named, than effort spent on one's own blog. (Moderately firm: a strong association, observational, in six categories.)

Do not read a citation of your content as a recommendation. In three out of five answers that cited an account's own site for an unbranded question, the account was not named. Informative articles earn citations as sources of information. Pages about the company itself, the homepage above all, are what is cited when the answer is about the company. (Firm.)

The prior is slow. Whatever determines a model's prior, it is fixed when the model is trained and changes only with the next model. Being written about widely, by others, in connection with one's category, over the period that will become a future model's training data, is the only lever on it that the mechanism suggests. This is an inference from how the systems work and from the literature on frequency and recall [15, 16], not something this study measured.

14.2 For a publisher that wants a page to be cited

Rank for the query the model writes, not the one the person typed. Position among the retrieved results is the strongest measured predictor of citation: the first result was cited at more than twice the rate of the ninth. And the queries are not the prompts. ChatGPT and Gemini split a prompt into four or five searches that keep about two fifths of its words; ChatGPT and Claude add the current year to roughly a third of their queries. Where an engine exposes the queries it issued, they are the best available guide to what a page needs to rank for. (Firm for Claude on position; the query description rests on one scan.)

Make the title say what the question asks. A title containing the terms of the user's question was cited more, at a given position, than one that did not. A recent year in the title helped modestly. (Moderately firm, one engine.)

Write in passages that stand alone. In two analyses with different outcomes and comparison groups, pages with a higher share of short, self-contained paragraphs were cited more: about twice the odds of citation by Claude given retrieval, and a third more repeat citations across all engines. Pages with more headings did slightly better in both. A citation attaches to a passage, and the model sees an excerpt and not the page. A page built from statements that each make sense alone gives every possible excerpt something citable. (Moderately firm; observational; consistent with the experimental literature [1].)

Do not pad. Longer pages were cited less in both analyses. The excerpt the model receives is of bounded size whatever the page's length. (Moderately firm.)

Include concrete figures. A higher density of statistics was associated with citation given retrieval by Claude, in line with the experimental finding of Aggarwal et al. [1], though it showed nothing in the repeat-citation analysis. (Suggestive.)

Do not expect much from labelled blocks. FAQ sections, summary boxes, definition-style openings, question-form headings, tables and lists showed no detectable association with citation in either analysis. The evidence cannot rule out small effects, and there may be other good reasons to include them. It gives no support to the idea that they are what makes a page citable. (A null result on loosely measured features: weak evidence, but evidence.)

Understand what a listicle does. "Best of" pages are retrieved for many variants of a question and so accumulate citations, even though, once retrieved, an explanatory guide is slightly more likely to be cited. Their main effect may not be on citation at all: they are where engines appear to find the names they list. (Suggestive.)

14.3 For anyone measuring AI visibility

Treat an answer as a draw. An unprompted mention recurred on the next reading less than half the time overall, and about one time in seven or eight on AI Overview and ChatGPT. Cited sites recurred at a rate of about a third on the same day. No conclusion should rest on one reading of one prompt. (Firm.)

Size the programme to the question. One hundred unbranded prompts read once estimate a mention rate of 0.6% to within plus or minus 1.5 percentage points. Detecting a doubling of such a rate needs thousands of readings. Either the programme must be that large, or it must track outcomes with higher base rates, such as whether the brand's site is among the sources or whether any of a set of competitors is named, or it must pool across engines and across time before interpreting. Adding prompts buys more precision than repeating readings, though both help. (Firm.)

Report intervals, computed by resampling prompts. A binomial interval treats readings as independent and is too narrow by a factor that depends on how many readings share a prompt. (Firm.)

Separate branded from unbranded prompts. They measure different things and differ about fortyfold in base rate. A blended "visibility score" is dominated by the branded prompts and can sit unchanged while unprompted discovery, the thing most brands want, is zero. (Firm.)

Separate mentions from citations, for the reasons of Section 12.4. (Firm.)

Measure each engine. Two engines share a tenth of their combined sources for the same question, and they differ in whether they search, what they cite and how repeatable they are. A result on one says little about another. (Firm.)

Validate the detector on both kinds of answer. If mentions are found by matching text, check the matcher's recall separately on answers with and without sources. An engine without retrieved pages in front of it writes names differently. And confirm that everything in the table came through the same code path. (Firm, and learned the hard way.)

Record whether the engine searched, and if the API exposes them, the queries it issued and the full list of what it retrieved as well as what it cited. Without the first, a large part of the variation in every outcome is unexplained by construction. Without the last, there is no comparison group for any statement about what makes a page citable. (Firm.)

14.4 For researchers

Observational panels of the kind used here are cheap, large and ecologically valid, and they can support causal claims only where a natural contrast happens to exist. This paper found three: an engine's search tool withdrawn, engines that choose whether to search, and one API that reports what it read. Others will exist in other instruments, and looking for them is worthwhile. But the questions left open in Section 12.9 will not be closed by more observation. They need retrieval to be manipulated, and the next section describes two experiments, neither expensive, that would do it.

For work in this area generally, three habits would raise the standard. Report the retrieved set and not only the cited set wherever an API allows it. Repeat each query several times and report the between-run variance alongside the result, since a per-query finding from a single run is one draw. And say exactly how mentions were detected and how the detector was checked.

15. Conclusion and future work

15.1 Conclusion

This paper asked what decides who is named and who is cited by AI answer engines, and whether the answer depends on the engine having searched the web.

It does, and the reason is mechanical. A language model has its parameters and its context and nothing else. A web search adds several thousand tokens of retrieved text to the context, some ten thousand in the configurations we measured, against a dozen for the question, and the model then writes its answer conditioned on both. A page influences an answer only by being retrieved and then used. A brand is named either because the parameters favour it or because the retrieved text presents it and the model takes it up.

Measured on 68,636 answers from seven engines over five months, and on a further 4,284 answers to fixed sets of category questions, the consequences of that mechanism were these.

The engines differ in whether they search at all, from always to never, and ChatGPT searches for commercial and time-sensitive questions and answers explanatory ones from memory more often than not.

When an engine that had been searching stopped, its unprompted naming of small brands fell by 63% while the engines that kept searching, asked the same questions, named them three times as often as before. One engine, across 556 questions it answered both ways, named a small brand unprompted only when it had searched.

Yet when engines do search, most of who is named is still decided by a prior the engines share. How often a model with no search names a brand for a category predicted which brands five searching engines named with a pseudo-R^2 of 0.52; what those engines actually retrieved added three hundredths. Retrieval matters most for brands with no prior, which are named eighty times more often when their own site is among the sources, and even then it mostly fails: seven times in a hundred. A well-known brand is named a third of the time with nothing of its own in the context.

A brand's presence in third-party pages that list and compare was more strongly associated with being named than the retrieval of the brand's own site.

Citations are spread over tens of thousands of sites with a heavy tail, most of them ordinary company and publisher sites cited once. The engines scarcely agree: two engines answering the same question at the same moment share a tenth of their combined sources.

Being cited and being named are different outcomes. In three of five answers that cited a brand's own site for a question that did not name the brand, the brand went unnamed.

Of the pages placed in one engine's context, 42% were cited. Position was decisive, with the first result cited at twice the rate of the ninth. After position came the match between the page's title and the question, and after that the grain of the page's prose: shorter pages built from dense, self-contained passages were cited more. The labelled blocks often recommended for the purpose showed no association with being cited.

And an answer is a draw from a distribution, not a fact about the engine. An unprompted mention recurred on the next reading of the same question by the same engine less than half the time. A brand tracking a hundred questions cannot tell a doubling of its mention rate from noise.

Along the way, three artefacts of measurement each produced a strong and false result before being found, and two more would have quietly shifted every rate. In this domain, whether the engine searched changes how answers are written and stored, and a detector that behaves differently across that divide will manufacture an effect of search.

Taken together the results describe a system in which visibility has two layers on two timescales. There is a slow layer, the prior, laid down in training from everything written about a brand before the cutoff, shared across vendors, strongly favouring the established, and not movable by

anything a brand does to its own pages this month. And there is a fast layer, retrieval, which operates only when the engine searches, admits the unknown at the margin, turns over substantially from one reading to the next, and is governed by ranking for queries the model writes and by whether a page offers passages that can be lifted and credited. Most of the advice in circulation addresses the fast layer. Most of the outcome, on this evidence, is decided in the slow one.

15.2 Two experiments

Two experiments were designed for this study and not run, because the study was confined to data already collected. They are specified here in enough detail to be carried out, because each addresses a gap that further observation cannot close.

Experiment A: a paired search ablation. *Question:* what is the causal effect of search on which brands are named, per engine, free of the model's own selection of when to search? *Design:* take a fixed set of about 300 unbranded prompts from frozen category sets. For each of the three engines that accept the setting (ChatGPT, Claude, Gemini), issue every prompt under two conditions, search forced on and search disabled, with three replicates of each, in randomised order within a single day: 5,400 calls. Request the full list of retrieved results as well as citations wherever the API provides it. Extract every brand named. *Analysis:* the unit is the prompt; the outcome is the set of brands named under each condition. The primary estimand is the within-prompt difference in the probability that each roster brand is named, estimated by conditional logistic regression with the prompt as stratum and the interaction of condition with the brand's prior as the term of interest. *What it adds:* the Grok design of this paper with three treated engines instead of one, randomised assignment in place of a calendar date, the same prompts on both sides, and replication to separate the effect of search from run-to-run noise. It would also yield retrieved-but-uncited comparison sets for ChatGPT and Gemini, extending Study 4 beyond a single engine. *Cost:* modest, on the order of a hundred US dollars in API charges at current prices.

Experiment B: randomised context injection. *Question:* what properties of a retrieved page cause the brand it presents to be named, and the page to be cited? *Design:* follow the logic of Aggarwal et al. [1] and of Kumar and Lakkaraju [24], with the difference that the outcome is naming as well as citation and the brand is unknown to the model by construction. For each of about fifty category questions, assemble a fixed set of nine real retrieved pages. Add a tenth page for a fictitious brand, a name with no occurrence in any corpus, and supply all ten to the model as search results in its context, with search itself disabled. Vary the fictitious brand's page in a fractional factorial design over factors suggested by this paper: its position among the ten (first, fifth, last); whether it is the brand's own page or a third-party comparison that includes the brand; the share of self-contained paragraphs; the presence of statistics; its length; whether the title matches the question. *Analysis:* logistic regression of the naming indicator on the factors with the question as a random effect; the same for citation. *What it adds:* because the brand is fictitious its prior is exactly zero, and because the page's properties are assigned at random, each coefficient is a causal effect on π_1^+ and on $\Pr(\text{cited} \mid \text{retrieved})$ in the notation of Section 12.1. It would settle whether the position gradient of Study 4 is the model's or the search engine's, since position

would be randomised. It would give the true ceiling on what a page in the context can do for an unknown brand, which we estimate observationally at 7% and suspect is confounded downwards. And it would test the page-feature associations of Section 10.4 experimentally. *Limitation:* supplying pages directly is not identical to the vendor's search tool, which controls the excerpt; the experiment measures the generator's behaviour and not the retriever's. *Cost:* a few thousand calls to small models, tens of dollars.

15.3 Further directions

Beyond those two, several extensions follow directly from the limits of the present data.

A broader panel. The most valuable single improvement would be a panel of prompts drawn from what people ask and not from what a set of customers chose to track, with brands spanning the full range of size, read on a fixed schedule. The category sets used here are a step in that direction and cover six categories for three months.

Measuring the prior properly. A brand's prior should be measured from several models without search and related to an external measure of how much has been written about the brand. That would test the link, assumed here on the strength of the literature, between frequency in text and frequency in answers, and would show how quickly the prior moves from one model generation to the next.

The consumer products. Everything here was measured through APIs. Whether the consumer applications search more often, retrieve more, and cite differently is an open and important question that requires a different method of collection.

The retrieval stage. For engines that report their queries, running those queries against the underlying search services and comparing the full rankings with what was retrieved and cited would open up the factor of equation (27) that this paper could not observe.

Excerpts. An engine that exposed the retrieved text in readable form would allow the page-feature analysis to be redone on what the model saw. Failing that, a careful simulation of how search tools excerpt pages would be informative.

Time. The panel will continue to grow. With a year or more of readings on stable prompts it will become possible to ask the question that matters most to practitioners and that this paper could not: when something is published about a brand, how long does it take to affect what the engines say, by which route, and for how long.

15.4 Data and code availability

The analysis code, the figure code and every aggregate statistic reported in this paper are kept in the repository from which this page is built, under `scripts/research` and `lib/research/results`, and are available from the author on request. Each number in the text can be traced to a

field in one of six results files, each produced by one script. The raw answers cannot be released, because they would identify the customers whose prompts they answer. Researchers who wish to replicate the analysis on their own collections can run the same scripts against any table of stored answers with the fields listed in Appendix B.

Acknowledgements

The author thanks the customers of RankMonster, whose scans are the data, and the colleagues who built and maintained the instrument. Portions of the analysis code and of the text were prepared with the assistance of an AI model, Anthropic's Claude; the author reviewed the analysis and is responsible for its content and for any errors.

References

1. Aggarwal, P., Murahari, V., Rajpurohit, T., Kalyan, A., Narasimhan, K., and Deshpande, A. (2024). GEO: Generative Engine Optimization. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. arXiv:2311.09735. doi:10.1145/3637528.3671900.
2. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., and Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*. arXiv:2005.11401.
3. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems 30*. arXiv:1706.03762.
4. Sennrich, R., Haddow, B., and Birch, A. (2016). Neural Machine Translation of Rare Words with Subword Units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*. arXiv:1508.07909.
5. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W. (2020). Dense Passage Retrieval for Open-Domain Question Answering. *Proceedings of EMNLP 2020*. arXiv:2004.04906.
6. Nakano, R., Hilton, J., Balaji, S., Wu, J., Ouyang, L., Kim, C., Hesse, C., Jain, S., Kosaraju, V., Saunders, W., Jiang, X., Cobbe, K., Eloundou, T., Krueger, G., Button, K., Knight, M., Chess, B., and Schulman, J. (2021). WebGPT: Browser-assisted question-answering with human feedback. arXiv:2112.09332.
7. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. (2023). ReAct: Synergizing Reasoning and Acting in Language Models. *International Conference on Learning Representations (ICLR 2023)*. arXiv:2210.03629.

8. Schick, T., Dwivedi-Yu, J., Dessì, R., Raileanu, R., Lomeli, M., Zettlemoyer, L., Cancedda, N., and Scialom, T. (2023). Toolformer: Language Models Can Teach Themselves to Use Tools. arXiv:2302.04761.
9. Asai, A., Wu, Z., Wang, Y., Sil, A., and Hajishirzi, H. (2023). Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. arXiv:2310.11511.
10. Gao, T., Yen, H., Yu, J., and Chen, D. (2023). Enabling Large Language Models to Generate Text with Citations. *Proceedings of EMNLP 2023*. arXiv:2305.14627.
11. Bohnet, B., Tran, V. Q., Verga, P., Aharoni, R., Andor, D., Baldini Soares, L., Ciaramita, M., Eisenstein, J., Ganchev, K., Herzig, J., Hui, K., Kwiatkowski, T., Ma, J., Ni, J., Sestorain Saralegui, L., Schuster, T., Cohen, W. W., Collins, M., Das, D., Metzler, D., Petrov, S., and Webster, K. (2022). Attributed Question Answering: Evaluation and Modeling for Attributed Large Language Models. arXiv:2212.08037.
12. Menick, J., Trebacz, M., Mikulik, V., Aslanides, J., Song, F., Chadwick, M., Glaese, M., Young, S., Campbell-Gillingham, L., Irving, G., and McAleese, N. (2022). Teaching language models to support answers with verified quotes. arXiv:2203.11147.
13. Liu, N. F., Zhang, T., and Liang, P. (2023). Evaluating Verifiability in Generative Search Engines. *Findings of the Association for Computational Linguistics: EMNLP 2023*, 7001–7025. arXiv:2304.09848.
14. Liu, N. F., Lin, K., Hewitt, J., Paranjape, A., Bevilacqua, M., Petroni, F., and Liang, P. (2024). Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12, 157–173. doi:10.1162/tacl_a_00638.
15. Kandpal, N., Deng, H., Roberts, A., Wallace, E., and Raffel, C. (2023). Large Language Models Struggle to Learn Long-Tail Knowledge. *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*. arXiv:2211.08411.
16. Mallen, A., Asai, A., Zhong, V., Das, R., Khashabi, D., and Hajishirzi, H. (2023). When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023)*, 9802–9822. arXiv:2212.10511.
17. Longpre, S., Perisetla, K., Chen, A., Ramesh, N., DuBois, C., and Singh, S. (2021). Entity-Based Knowledge Conflicts in Question Answering. *Proceedings of EMNLP 2021*. arXiv:2109.05052.
18. Xie, J., Zhang, K., Chen, J., Lou, R., and Su, Y. (2024). Adaptive Chameleon or Stubborn Sloth: Revealing the Behavior of Large Language Models in Knowledge Conflicts. *International Conference on Learning Representations (ICLR 2024)*. arXiv:2305.13300.
19. Wu, K., Wu, E., and Zou, J. (2024). ClashEval: Quantifying the tug-of-war between an LLM's internal prior and external evidence. arXiv:2404.10198.

20. Wan, A., Wallace, E., and Klein, D. (2024). What Evidence Do Language Models Find Convincing? *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024)*, 7468–7484. arXiv:2402.11782.

21. Shi, F., Chen, X., Misra, K., Scales, N., Dohan, D., Chi, E., Schärli, N., and Zhou, D. (2023). Large Language Models Can Be Easily Distracted by Irrelevant Context. *Proceedings of the 40th International Conference on Machine Learning (ICML 2023)*. arXiv:2302.00093.

22. Cuconasu, F., Trappolini, G., Siciliano, F., Filice, S., Campagnano, C., Maarek, Y., Tonellotto, N., and Silvestri, F. (2024). The Power of Noise: Redefining Retrieval for RAG Systems. arXiv:2401.14887. doi:10.1145/3626772.3657834.

23. Dai, S., Zhou, Y., Pang, L., Liu, W., Hu, X., Liu, Y., Zhang, X., Wang, G., and Xu, J. (2024). Neural Retrievers are Biased Towards LLM-Generated Content. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24)*. arXiv:2310.20501.

24. Kumar, A., and Lakkaraju, H. (2024). Manipulating Large Language Models to Increase Product Visibility. arXiv:2404.07981.

25. Pfrommer, S., Bai, Y., Gautam, T., and Sojoudi, S. (2024). Ranking Manipulation for Conversational Search Engines. arXiv:2406.03589.

26. Nestaas, F., Debenedetti, E., and Tramèr, F. (2024). Adversarial Search Engine Optimization for Large Language Models. arXiv:2406.18382.

27. Puerto, H., Gubri, M., Green, T., Oh, S. J., and Yun, S. (2025). C-SEO Bench: Does Conversational SEO Work? *NeurIPS 2025 Datasets and Benchmarks Track*. arXiv:2506.11097.

28. Chen, M., Wang, X., Chen, K., and Koudas, N. (2025). Generative Engine Optimization: How to Dominate AI Search. arXiv:2509.08919.

29. Kumar, A., and Palkhouski, L. (2025). AI Answer Engine Citation Behavior: An Empirical Analysis of the GEO16 Framework. arXiv:2509.10762.

30. Zhang, P., Ye, Q., Peng, Z., Garimella, K., and Tyson, G. (2025). Source Coverage and Citation Bias in LLM-based vs. Traditional Search Engines. arXiv:2512.09483.

31. Narayanan Venkit, P., Laban, P., Zhou, Y., Mao, Y., and Wu, C.-S. (2024). Search Engines in an AI Era: The False Promise of Factual and Verifiable Source-Cited Responses. arXiv:2410.22349.

32. Jaźwińska, K., and Chandrasekar, A. (2025). AI Search Has a Citation Problem. *Columbia Journalism Review*, Tow Center for Digital Journalism, 6 March 2025.

33. Chapekis, A., and Lieb, A. (2025). Google users are less likely to click on links when an AI summary appears in the results. Pew Research Center, 22 July 2025.

34. Kamruzzaman, M., et al. (2024). "Global is Good, Local is Bad?": Understanding Brand Bias in LLMs. *Proceedings of EMNLP 2024*. arXiv:2406.13997.

35. Chu, X., and Hou, Y. (2026). Incumbent Advantage: Brand Bias and Cognitive Manipulation Dynamics in LLM Recommendation Systems. arXiv:2606.17443.

36. Malthouse, E., Lee, K.-Y., Yang, J., Pal, S., and Feng, X. (2026). Evaluating Brand Retrieval and Ranking in Large Language Model Recommendations. arXiv:2609.16304.
37. Atil, B., Aykent, S., Chittams, A., Fu, L., Passonneau, R. J., Radcliffe, E., Rajagopal, G. R., Sloan, A., Tudrej, T., Ture, F., Wu, Z., Xu, L., and Baldwin, B. (2024). Non-Determinism of "Deterministic" LLM Settings. arXiv:2408.04667.
38. Anthropic. Web search tool. Claude API documentation. platform.claude.com/docs/en/agents-and-tools/tool-use/web-search-tool. Retrieved 15 November 2026.
39. OpenAI. Web search. API documentation. developers.openai.com/api/docs/guides/tools-web-search. Retrieved 15 November 2026.
40. Google. Grounding with Google Search. Gemini API documentation. ai.google.dev/gemini-api/docs/google-search. Retrieved 15 November 2026.
41. Perplexity. Web Search. Agent API documentation. docs.perplexity.ai/docs/agent-api/tools/web-search. Retrieved 15 November 2026.
42. Brin, S., and Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1–7), 107–117.
43. Robertson, S., and Zaragoza, H. (2009). The Probabilistic Relevance Framework: BM25 and Beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389.
44. Joachims, T., Granka, L., Pan, B., Hembrooke, H., and Gay, G. (2005). Accurately interpreting clickthrough data as implicit feedback. *Proceedings of the 28th Annual International ACM SIGIR Conference*, 154–161.
45. Craswell, N., Zoeter, O., Taylor, M., and Ramsey, B. (2008). An experimental comparison of click position-bias models. *Proceedings of the 2008 International Conference on Web Search and Data Mining (WSDM)*, 87–94.
46. Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158), 209–212.
47. Efron, B. (1979). Bootstrap methods: another look at the jackknife. *The Annals of Statistics*, 7(1), 1–26.
48. Cameron, A. C., and Miller, D. L. (2015). A practitioner's guide to cluster-robust inference. *Journal of Human Resources*, 50(2), 317–372.
49. Liang, K.-Y., and Zeger, S. L. (1986). Longitudinal data analysis using generalized linear models. *Biometrika*, 73(1), 13–22.
50. McFadden, D. (1974). Conditional logit analysis of qualitative choice behavior. In P. Zarembka (ed.), *Frontiers in Econometrics*, 105–142. Academic Press.
51. Chamberlain, G. (1980). Analysis of covariance with qualitative data. *The Review of Economic Studies*, 47(1), 225–238.

52. Bertrand, M., Duflo, E., and Mullainathan, S. (2004). How much should we trust differences-in-differences estimates? *The Quarterly Journal of Economics*, 119(1), 249–275.

53. Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B*, 57(1), 289–300.

54. Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Review*, 51(4), 661–703. arXiv:0706.1062.

55. Alstott, J., Bullmore, E., and Plenz, D. (2014). powerlaw: a Python package for analysis of heavy-tailed distributions. *PLoS ONE*, 9(1), e85777.

56. Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1), 37–46.

57. Mann, H. B., and Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18(1), 50–60.

58. Cliff, N. (1993). Dominance statistics: ordinal analyses to answer ordinal questions. *Psychological Bulletin*, 114(3), 494–509.

59. Hill, M. O. (1973). Diversity and evenness: a unifying notation and its consequences. *Ecology*, 54(2), 427–432.

60. Jaccard, P. (1912). The distribution of the flora in the alpine zone. *New Phytologist*, 11(2), 37–50.

61. Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6), 80–83.

62. Seabold, S., and Perktold, J. (2010). Statsmodels: econometric and statistical modeling with Python. *Proceedings of the 9th Python in Science Conference*, 92–96.

63. Zipf, G. K. (1949). *Human Behavior and the Principle of Least Effort*. Addison-Wesley.

64. Barabási, A.-L., and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439), 509–512.

65. Cameron, A. C., and Trivedi, P. K. (2013). *Regression Analysis of Count Data* (2nd ed.). Cambridge University Press.

66. Kish, L. (1965). *Survey Sampling*. John Wiley & Sons.

Appendix A. Engine configurations

The request parameters below are those of the instrument's clients as of 15 November 2026, with the dates on which each changed during the study period. Every engine was called from servers in the United States with no conversation history, user identity or location.

ChatGPT. OpenAI Responses API. Model `gpt-5` from 9 June to 14 July 2026 and `gpt-5-mini` from 15 July. No system instruction: the request contains the prompt and nothing else, on the reasoning that any added instruction moves the answer away from what the model would otherwise say. Tools: `web_search`, with `tool_choice` set to `auto` so that the model decides whether to search. `max_tool_calls` was 2 from 11 June and 1 from 8 July. Reasoning effort `low`, text verbosity `medium`, `max_output_tokens` 2,000, responses not stored. Sources are the `url_citation` annotations on the output text. Requests are retried up to twice on rate-limit and server errors.

Claude. Anthropic Messages API. Model `claude-haiku-4-5`. System instruction: "You are a helpful assistant. Answer the following question concisely and accurately." Tools: the web search tool with `max_uses` 5; the model decides whether and how often to search. `max_tokens` 1,024. Sources are read from two places in the response: the `citations` attached to text blocks, which give the cited URL, its title and the quoted passage, and the `web_search_tool_result` blocks, which list every result returned. From 30 September 2026 each source is stored with the character spans of the answer it supports, so a source without a span is a result that was retrieved and not cited. From 20 October 2026 the stored token count is the full usage report including cached tokens.

Gemini. Google Generative AI SDK. Model `gemini-2.5-flash`. System instruction as for Claude. Tools: Google Search grounding; the model decides whether to search. `maxOutputTokens` was 400 until 9 November 2026 and 8,192 afterwards. The model's internal reasoning tokens count against that cap, and under the earlier setting a large share of grounded answers were returned truncated (Section 5.6). Sources are the grounding chunks, whose URLs are redirects; the publisher's domain is taken from the chunk's title.

Perplexity. Chat completions endpoint. Model `sonar`. System instruction as for Claude. `max_tokens` 400, `search_context_size` `low`. Every request is searched. Sources are the `search_results` list returned with the answer.

Google AI Overview. Not a model API. The AI Overview on the Google results page for the prompt, with country United States and language English, is read through a third-party search results API: SerpApi from 15 July to 6 October 2026 and SearchApi from 7 October. The overview is usually returned by a second request using a token from the first. Each request is abandoned after 20 seconds and retried. Sources are the overview's references; where a reference is tied to a text block of the overview, the block's span is stored with it. If no overview is shown for the query, no answer is stored.

Grok. xAI Responses API. Model `grok-4.3`. System instruction as for Claude. `max_output_tokens` 2,000. From 9 June to 7 August 2026 the built-in `web_search` tool was offered and the model decided whether to search. On 7 August the tool was removed from the request to reduce cost; because of scans in flight and staged deployment the last sourced Grok answer is dated 13 August. From 16 August no Grok answer has sources.

Copilot. An Azure-hosted model, in service until 15 July 2026, when it was replaced in the product by Google AI Overview. Its client is no longer in the codebase and its parameters were not recovered for this paper. It contributes 808 answers to the analysis set and is reported for completeness only.

Before 9 June 2026. ChatGPT (an earlier model through the chat completions endpoint), Claude and Grok were called without search tools, and no engine's sources were stored. Those 15,313 answers are excluded from all analyses.

Onboarding audits. A separate route, used when a prospective customer signs up, calls the engines through older clients without search and writes its results to the same tables. It is identified in the data as a run in which no engine returned a source, and is excluded (Sections 5.3 and 5.6).

Appendix B. Variables

Table B1. Variables in the answer panel.

Variable	Definition
Fresh	The answer was generated for this request and not served from the instrument's response cache.
Full scan	The answer belongs to a run in which at least one engine returned at least one source.
Fragment	The answer has fewer than 40 whitespace-separated words.
Scorable	The account's saved domain yields a brand label of at least four characters that is not the name of a shared hosting or social platform.
Brand label	The label immediately in front of the public suffix of the account's saved domain, by the public suffix list.
Mentioned	The brand label occurs in the readable text of the answer under the rule of Appendix C.
Branded	The brand label occurs in the prompt under the rule of Appendix C without the capitalisation requirement.
With sources	The answer was returned with at least one source record.
Site	The registrable domain of a source: public suffix plus one label.
Own site cited	At least one source has the same registrable domain as the account's saved domain.
Tokens	Total tokens billed for the request as reported by the provider.
Prompt category	One of awareness, recommendation, comparison, industry or use case, assigned when the prompt was created.

Table B2. Variables for retrieved results (Study 4).

Variable	Definition
Position	Zero-based index of the result in the stored source list, which preserves the order in which the search tool returned results.
Cited	The result carries at least one answer span.
Later search	Position is 9 or greater.
Title overlap	Share of the prompt's content words (lower-cased alphanumeric tokens, stop words removed) that occur in the result's title.
Year in title	The title contains a year from 2025 to 2029.
List title	The title begins with a number, or with "top" or "best" followed by a number, or contains "best" or "top".
Comparison title	The title contains "vs", "versus", "alternative" or a form of "compare".
Homepage	The URL path is empty.
Breadth	Number of accounts in whose sourced answers the result's site appears, anywhere in the source era.

Table B3. Page features (Studies 4 and 5), computed from the fetched page.

Feature	Definition
Word count	Words in the main text after conversion to markdown.
Heading count	Second- and third-level headings.
List count, table count	Lists and tables in the main text.
Statistics	Count of numeric expressions with a unit, percentage, currency or multiplier.
Quotable paragraphs	Paragraphs that are short, contain a complete statement, and do not open with a reference to preceding text; reported as a share of all paragraphs.
Question headings	Headings ending in a question mark or opening with an interrogative.
FAQ block	A section headed as frequently asked questions, or FAQ structured data.
Summary box	A section labelled as a summary, key takeaways or "TL;DR" near the top of the page.
Definition opening	The lead paragraph has the form "X is a ...".
Author byline	An author name in the page's metadata or near its title.
Filler introduction	The lead paragraph matches a list of generic opening phrases.
Structured data	At least one schema.org type declared.
Outbound links	Links to other registrable domains.
Format	One of guide, listicle, landing page, comparison table, review, news; assigned by a language model from the page's text.
Intent	Informational, comparison, transactional or navigational; assigned by the same classifier.

The page features were implemented for a product feature and are heuristics. Their exact definitions are in the instrument's source code and are summarised here in the terms the code uses.

Appendix C. The mention-matching rule

The rule takes an answer's text and an account's brand label and returns whether the brand is named. It was written to be simple enough to state completely.

1. **Derive the label.** Take the account's saved domain. Remove any scheme and path. The label is the part immediately in front of the public suffix: `acme` for `acme.com`, `acme` for `shop.acme.co.uk`. Discard everything but letters and digits. If the label has fewer than four characters, or is the name of a shared platform (a site builder, a code host, a social network), the account is unscorable.
2. **Reduce the answer to what a reader sees.** Replace every markdown link whose visible label is nothing but the host name of its own target with a space; this removes source chips. Replace every other markdown link with its visible label. Remove bare URLs.
3. **Search.** Look for the label's characters in order, ignoring case, allowing up to two separator characters (white space, hyphen, apostrophe or full stop) between any two consecutive characters, and requiring that the match is not preceded or followed by a letter or digit.
4. **Check capitalisation.** If the match contains no separators it counts. If it contains separators it counts only if its first character is an upper-case letter and at least one character following a separator is upper case.

5. **Result.** The brand is mentioned if any match counts. The position of the first counted match, divided by the length of the reduced text, is its relative position.

For prompts, step 4 is omitted.

The strict rule used as a check in Section 7.4 is the same without separators in step 3.

Examples for the label `acmewidgets`. Counted: "AcmeWidgets", "acmewidgets", "Acme Widgets", "Acme-Widgets", "ACME WIDGETS". Not counted: "acme widgets" in lower case in running text; "the `acmewidgets.com` page" where the name appears only as a source chip or bare URL; "Acme" alone; "Acme and Widgets".

Known failures. A brand whose trading name is not its domain label is never matched. A label that is an ordinary capitalised phrase, or that begins a sentence as ordinary words, can be matched falsely. A brand written with an ampersand or an abbreviation where the domain spells the word out is missed. Possessives are handled ("Acme's Widgets" matches); plurals of the final word are not, unless the label already contains them.

Agreement. Against the indicator stored by the instrument, whose own rule changed twice during the period, the rule agrees on 98.1% of 95,555 scorable answers. It finds 9,011 mentions; the strict rule finds 6,153.

Appendix D. Reproduction

The analysis is an export script, six analysis scripts, a figure script and one shared module, run in order against a local export of the stored tables.

Step	Script	Produces
Export	`export.py`	Local copies of stored answers, category-scan answers, fetched pages, logged queries, brand rosters. Read-only database session.
Sample	`s0_sample.py`	The sample flow of Section 5: counts at each exclusion, matcher agreement.
Study 1	`s1_search.py`	Tables 1 and 6 to 9.
Study 2	`s2_brands.py`	Tables 10 to 14.
Study 3	`s3_citations.py`	Tables 15 to 20.
Study 4	`s4_retrieved_vs_cited.py`	Tables 21 to 23.
Study 5	`s5_pages_queries_stability.py`	Tables 24 to 29.
Figures	`figures.py`	Figures 2 to 10.

The shared module holds the mention rule of Appendix C, the definition of a full scan, the Wilson interval, the cluster bootstrap and the Benjamini-Hochberg adjustment. Each study script writes one results file of aggregates; no results file contains an answer, a prompt, or an account identifi-

er. All random number generators are seeded with a fixed value, so the bootstrap intervals and the simulation of Table 29 reproduce exactly from the same export.

The data were drawn on 15 November 2026. The database is live, and a later export will contain later scans; the scripts apply the date windows given in the text, so results for those windows should change only if stored rows are deleted, which happens when a customer deletes a prompt or an account.

To apply the scripts to another collection, the answers table needs, per answer: an identifier, a run identifier, an account identifier, a prompt identifier and the prompt's text, the engine, a timestamp, the answer text, the list of sources with URL, title and domain, the token count if available, and the domain that identifies the brand being tracked.

Software versions. Python 3.13; pandas 3.1, NumPy 2.5, SciPy 1.18, statsmodels 0.15, powerlaw 2.0; matplotlib 3.11 for figures.

Cite this paper

Robert Griesmeyer (2026). What Decides Who AI Answer Engines Name and Cite: An observational study of 68,636 answers from seven engines, with and without web search (Version 1.0). RankMonster Research. https://www.rankmonster.ai/research/who-gets-named-and-cited

```
@techreport{griesmeyer2026what,
  title       = {What Decides Who AI Answer Engines Name and Cite: An
observational study of 68,636 answers from seven engines, with and without web
search},
  author      = {Robert Griesmeyer},
  institution = {RankMonster},
  year        = {2026},
  note        = {Version 1.0},
  url         = {https://www.rankmonster.ai/research/who-gets-named-and-cited}
}
```