TRUE CONTEXT CORRUPTS KNOWN FACTS IN LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study the behaviour of LLMs on structured puzzles: each puzzle consists of a candidate answer, and a list of categories, and the LLM is asked the yes/no question of if the candidate lies in the intersection of these categories. By exploring the case wherein the candidate answer fails membership in a single 'disqualifier' category, but lies in the rest, we establish that LLMs are 'distractible': even when in isolation the models are able to correctly reject membership in the disqualifier, as we add more correct distractor categories, they progressive become more likely to incorrectly say '`Yes`,' to the extent that with a few distractors, the majority of puzzles are answered incorrectly. This distraction is also shown to arise in free-form generation. We further establish that 'fact corruption' plays a non-trivial role: in certain settings, distraction arises not because the LLMs cannot conjunct category-memberships, but because they begin to endorse membership in the disqualifier. Together, this describes a new failure mode of LLMs on factual problems, and suggests a plausible mode by which hallucinations may occur.

1 INTRODUCTION

Hallucination in large language models (LLMs) is a difficult problem, in part because the term is used for a range of different failures (Ji et al., 2023; Huang et al., 2025; Kalai & Vempala, 2024; Kalai et al., 2025). Indeed, LLMs are used in a variety of ways and domains, with each showing some form of 'hallucination', but manifesting it in different ways, and for different causes (Kim et al., 2025; Dahl et al., 2024; Lee et al., 2025). A natural step towards getting a handle on this concept, then, is to study it in restricted domains. Factual questions are a particularly nice such domain, since a broad array of questions are available, and broadly accepted accompanying answers.

Yet hallucination in factual question-answering is not just a matter of right or wrong, e.g., a person answering a question incorrectly is not blanket-deemed to be hallucinating—they may not know the answer, or may be confusing some aspect of it. To even decide what is a hallucination, and a fortiori to gain insight on hallucinations, it is important to understand the incorrect answers that LLMs give in a more nuanced way. However, typical factuality evaluation often treats responses in a binary wrong/right way. This is true for both 'single-hop' and 'multi-hop' question datasets. (Joshi et al., 2017; Mallen et al., 2023; Yang et al., 2018).The case of multi-hop databases is interesting, in that these questions are require retrieval of atomic information and its composition, which in principle one can evaluate separately, but the catch being that getting to these underlying atomic information pieces and the required logic from a question posed in natural language is itself a hard task.

The Catfishing Game. To address this limitation, we study the behaviour of LLMs on structured factual puzzles: each puzzle consists of a list of Wikipedia categories, and the goal is to answer with a Wikipedia page that lies in the intersection of these categories. In fact, this is a popular daily internet game called 'catfishing', available on `catfishing.net` (cat, 2026). We extensively use data made available by the owner of this website, which gives us not only a curated set of puzzles, but also a distribution over wrong answers given by human players. Structurally, these puzzles lie intermediate to single- and multi-hop questions: they need retrieval of many category memberships, but for known categories, and the logic is just an AND.

Our main protocol further simplifies the structure into yes/no puzzles: we pick a subset of categories for a catfishing puzzle, and a single human answer (right or wrong), and ask the LLM if this answer lies in the intersection of the given categories. The output is scored as $P(\texttt{Yes}) := \pi(\texttt{Yes})/(\pi(\texttt{Yes}) + \pi(\texttt{No}))$, where π is the LLMs output distribution.

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



Figure 1: **Overview of the protocol and main finding. (1)** Each item pairs a target article with a wrong guess given by human players; the guess shares some of the target's categories but violates at least one. **(2)** Asked about each category on its own, the model accepts the shared ones and rejects the disqualifying one. **(3)** When the disqualifier is listed together with n shared categories and the model is asked whether the guess belongs to all of them, $P(\text{YES})$ rises with n, although the correct answer is NO at every length.

Distracting LLMs from Known Disqualifiers. Consider the above experiment, run with a **wrong candidate answer**, wherein we show the LLM one disqualifying category, and n satisfied categories. We find that even though LLMs can reliably disqualify the candidate on it's own (i.e., when $n = 0$), as n increases, the LLM's score P smoothly increases with it, and for moderate n, the majority of puzzles resolve to `Yes` (i.e., $P > 0.5$). Thus, even if LLMs can atomically disqualify the answer on the basis of one category, adding other true categories leads the LLM to incorrectly accept it. This effect holds across two families of open-source LLMs, and across model sizes, and is only associated with adding true categories as 'distractors': adding false categories diminishes P. Further, the strength of distraction is modulated by informativeness and placement of categories.

In a refinement of this experiment to a 'free-form' generation setting, we further find that a similar effect manifests: when comparing the mass placed on a true and a human-generated wrong answer, the model increases the mass placed on the wrong answer as n increases, and the 'hard-error' rate grows. For instance, for Qwen-32B, the error rate grows from 21% at $n = 1$ to 28% at $n = 7$.

Fact Corruptions and Logic Errors. One natural way this failure can occur is through a failure of logic, i.e., that the LLMs may, in some internal sense, continue to correctly judge each category membership, but fail to compute the AND of all these memberships. A contrasting way is fact corruption, whereby as n grows, the isolated disqualifying fact may itself be corrupted by the presence of other positive categories in the context. By refining the above protocol, we find that factual corruptions indeed play a role in these compositional puzzles. In particular, in a version of the experiment wherein the LLM is forced to explicitly resolve the each category membership before answering all, we find that logic errors mostly vanish, and fact corruption dominates the effect.

Wider Implications. The above experiments establish that the presence of truthful distractors in the model's context can lead it to endorse compositional facts which a known disqualifier, and thus can cause the model to endorse statements that in isolation it can reliably detect to be false. This observation has significant bearing on the wider phenomenon of hallucination, in that such a mechanism may affect more general modes of generation than our restricted setting of factual intersection-of-category questions. As an example, when pursuing high level composition of proofs or arguments, it is common for humans to develop several lines of thinking, and evaluate them on many criterion, with even a single failure eliminating this line of thinking. An LLM carrying out such a process as a subsidiary step in task can instead mistakenly accept this line of thinking (even if guarded against it in isolation), and thus introduce high-level errors that produce hallucinations. We believe that, in such ways, this distractibility of LLMs may shed light on more generic situations, and gives a concrete design target for future experimentation on this important problem.

Summary of Contributions. To summarise, our contributions are thus:

- We propose using catfishing-type puzzle to elicit nuance in factual question answering.
- We uncover that LLMs are distractible, in that adding true categories in the puzzles can lead to the LLM accepting an answer despite having a disqualifier it atomically recognized.
- We show that this effect also increasing the rate of false responses to such puzzles, and establish the effect of structural properties of categories and prompts on distractibility.
- We establish that logic errors do not capture all of this distraction, and fact corruption plays a role. Further, we show a version of puzzles wherein fact corruption dominates the phenomenon.

Related Work

Parametric knowledge. Language models store factual knowledge in their weights during pretraining, and this is referred to as parametric knowledge, as opposed to contextual knowledge supplied at inference time (Petroni et al., 2019; Roberts et al., 2020). Parametric knowledge is not always employed: probing studies have identified that representations can carry correct answers even if the generated answer is wrong (Orgad et al., 2025). It is also known that behavioural elicitation of individual queries and their composition are somewhat different question: a model can recall two facts, and still fail to joint them in a single forward pass (Biran et al., 2024). Our distraction phenomenon is similar on the surface, but it is driven by the presence of true facts that are relevant to the task. A further distinction is that we observe the corruption of facts the model knows: in all our experiments, we first verify each category on its own and keep only those the model already answers correctly, so any later error on them is caused by the context.

Contextual override. Added text can displace what the model already holds. Retrieval work reports large drops on questions the model had answered correctly once a misleading passage is supplied (Choi et al., 2025), and the balance between contextual and parametric knowledge shifts with the volume of context (Tao et al., 2024). A recent survey organises these cases into two kinds: context that is untruthful, and context that is irrelevant (Li et al., 2025). In our study, however, the added information in the context is neither false nor irrelevant: the truth of the distractor classes is certainly pertinent to the task of computing the conjunctive Yes/No answer. Nevertheless, the fact corruption we see is related to how irrelevant context can contextually affect parametric knowledge.

Factors and mechanisms. Individual properties of the added text have been identified as drivers of errors, including semantic proximity to the target (Yang et al., 2025; Wu et al., 2024), position within the prompt (Liu et al., 2023), and plausibility of competing content (Bhuiya et al., 2024). Yuksekgonul et al. (2024) model factual queries as constraint satisfaction problems and find that the attention a model pays to constraint tokens predicts whether its answer is correct. These studies concern irrelevant or misleading text, or observe correlations in the model's own answers. We instead control the context directly: we add true, relevant categories next to a disqualifier the model knows, and vary their number, their informativeness, and the position of the disqualifier. This lets us measure how each factor changes the error, and show that part of the error comes from corruption of the known fact itself.

2 SETUP

We begin by describing the setup of the catfishing puzzles, and the ensuing dataset, along with establishing some key definitions.

Catfishing Puzzle Structure. A catfishing puzzle arises from a Wikipedia page, which we will refer to as a *target*. Each Wikipedia page belongs to a set of Wikipedia categories (curated by the editors), which are presented at the bottom of the page. The catfishing puzzle presents the player with this set of categories, and the players guess the target, i.e., the title of the page. Fig. 1, Left shows an example of categories for the target *Rhine*, as well as a plausible wrong guess.

`catfishing.net` data. The website `catfishing.net` has run a daily catfishing game since July 2024. Each day, 10 curated puzzles are presented to the player, and since July 2024, most puzzles have had $\geq$ 4,000 players. Most puzzles have between 6 and 37 categories, and 94% of questions have ≥ 5 unique wrong answers. We scraped 7750 puzzles along with wrong answers and their frequencies from this website. Scraping was done with permission from the site owner, who also permitted us to use, but not to share, this data. The puzzle archive remains publicly accessible.

Terminology. For a page T, we denote the *set of categories T lies in* as $\mathcal{C}(T)$. We will use the set inclusion symbol to denote this, i.e., for any category $c \in \mathcal{C}(T), T \in c$ The catfishing puzzle thus reveals the members of $\mathcal{C}(T)$, and demands the player to produce a page in $\bigcap_{c\in\mathcal{C}(T)} c$.

For two pages T, W, we define the set of **shared categories** as $\mathcal{C}_{\text{sh}}(T, W) := \mathcal{C}(T) \cap \mathcal{C}(W)$, and the set of **$T$-categories missed by** W as $\mathcal{C}_{\text{mis}}(W; T) := \mathcal{C}(T) \setminus \mathcal{C}(W)$.

As described in the introduction, our main experiments involve analysing the behaviour of LLMs on player-generated wrong answers. For a page W that is an incorrect answer for a target T, we call any member of $\mathcal{C}_{\text{mis}}(W; T)$ a **disqualifier**, in that if $d \in \mathcal{C}_{\text{mis}}(W; T)$, then $W \notin d, T \in d$. Further, we

will call members of $\mathcal{C}_{\text{sh}}(W, T)$ **distractors**. The rationale is that a disqualfier immediately reveals membership status of W in the intersection, and distractor-membership is irrelevant for this.

Induced Yes/No Scores. In the bulk of our experiments, we turn catfishing-like puzzles into yes/no problems. For these setups, we measure the relative mass of the LLM answer on a `Yes` response against a `No` response. Following Kadavath et al. (2022), we score the response on a prompt p as

$$P(\texttt{Yes}; p) = \sigma\left(\text{logsumexp}_{v\in\mathcal{Y}}\, z_v - \text{logsumexp}_{v\in\mathcal{N}}\, z_v\right), \tag{1}$$

where z_v are logits at the first answer position, $\sigma(x) := 1/(1+e^{-x})$, and $\mathcal{Y}$ contains the string `Yes` in lower, title, and upper cases, with and without a leading space, and $\mathcal{N}$ analogously with `No`.

3 Distractors Lead to False Acceptance

We proceed to discussing our main result. We first detail the experimental design and protocol, and then present the main evidence for the distractibility phenomenon. A number of ablations analysing the effect of structural aspects of categories and the prompt on distraction follows.

3.1 Design

Structure. We investigate LLM performance on a verification version of a catfishing puzzle. Each puzzle comprises a list of $n+1$ categories $c_0, c_1, \cdots, c_n$, for varying number n, and a candidate answer A. The LLM receives the following prompt, which we denote as $\text{Prompt}(A; c_0^n)$

```
Groups:
- <category 0>
- ...
- <category n>
Does "<candidate answer>" belong to all of these groups? Answer Yes or No.
```

Scoring. We score the LLM response according to $\pi(\texttt{Yes})$ as described in (1). Accordingly, for each list of categories and answer A, we obtain a score

$$P(A; c_0^{n+1}) := P(\texttt{Yes}|\,\text{Prompt}(A; c_0^n)).$$

We will often abbreviate this to just $P(\texttt{Yes})$ or P, suppressing A, c_0^{n+1}. Notice P has an intrinsic threshold: if $P > 0.5$, then the LLM prefers a `Yes` answer to `No`, i.e., it judges that $A \in \bigcap_{i=0}^{n} c_i$.

Protocol. We investigate the behaviour of P for *wrong* answers, with varying lists of categories. The main controlling variable is n, the number of categories. Given n, we pick puzzles thus:

- Pick a target T, and a human-given wrong answer W such that $|\mathcal{C}_{\text{sh}}(W, T)| \geq n$.
- Randomly pick a *disqualifier* $d \in \mathcal{C}_{\text{mis}}(W; T)$. This passes if $P(W; d) < 0.5$.
- Sample n *distractors* out of $\mathcal{C}_{\text{sh}}(W, T)$ uniformly without replacement.
- Permute d and the above categories randomly to get $c_0, \cdots, c_n$.
- If d passed, evaluate and record $P(W; c_0^n)$. Else repeat.

In words, this sets up a puzzle with $n+1$ categories, of which one is a disqualifier, and n are distractors. Below, n ranges from 0 to 7, and we restrict attention to pairs (W, T) such that $|\mathcal{C}_{\text{sh}}(W, T)| \geq 3, |\mathcal{C}_{\text{mis}}(W; T)| \geq 1$. Finally, we restrict attention to T such that the number of human-generated wrong answers is at least 5, and restrict W to lie in the top-5 wrong answers for the puzzle $\mathcal{C}(T)$. Note that this gives a large variation of possible prompts between the wrong answers, puzzles, and choices and permutation of categories.

Atomically Known Disqualifier. Importantly, before evaluating the final prompt, we ensure that $P(W; d) < 0.5$, i.e., that the model correctly determines that $W \notin d$ when shown d alone. Thus, **we only study false facts that the LLM can detect atomically** (i.e., in isolation).

Notice that our puzzles amount to asking the AND of the truth values of $\{\ W \in c_i?\}_{i=0}^{n}$. Since the disqualifier d is present in the prompt, the answer to this puzzle should be `No`, i.e., P should be small. Since, by design, we restrict ourselves to puzzles where the disqualfier is atomically verified, any increase in P as n grows is evidence of this known fact failing to be employed in richer puzzles.

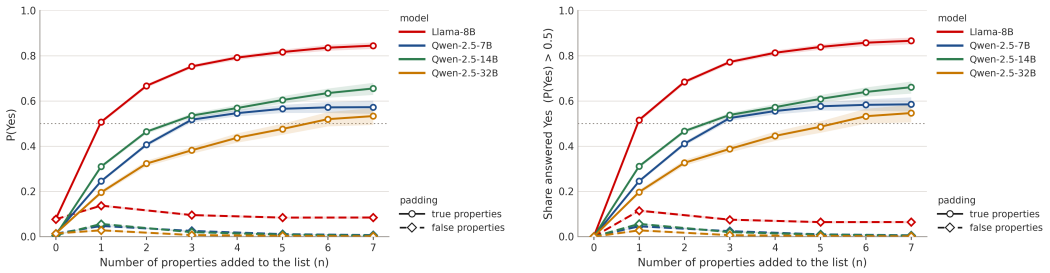


Figure 2: **Adding true properties drives false acceptance.** *Left:* Mean $P(\texttt{Yes})$ for a candidate ruled out by one disqualifier, as n categories that are true of it (solid) or n further false ones (dashed) are added to the list. *Right:* Fraction of candidates where $P(\texttt{Yes}) > 0.5$. Four models are presented, and the dashed lines indicate the control. Number of prompts used per point is at least 5K (see §B); shaded regions represent 95% CIs.

Control Experiment. We control for potential effects of increased context length by padding the context with random categories instead of members of $\mathcal{C}_{\text{sh}}(W;T)$. Explicitly, the same protocol is used, but random categories c such that $W \notin c$ are added.

3.2 Results

Growth of P. Fig. 2 presents the main finding underlying this paper: on a variety of model sizes across two LLM families, we find that the average of P over puzzles drawn as described above steadily increases as the list c_0^n grows in size. For all the models studied, this increase is strong: although the average P lies well below 0.1 at $n = 0$ for all models (and close to 0 for all but Llama-8B), it crosses above 0.5 by $n = 6$.

Importantly, the control curves in the same figure do not show such growth. Indeed, barring a rise at $n = 1$, they all exhibit decay with increasing n. Further, even the rise at $n = 1$ is much smaller than the corresponding rise with true categories. Thus, the observed effect is not obtained by adding arbitrary categories to the list, and relies upon adding categories that W indeed lies in.

Growth of Error Rate. Fig. 2 also presents the hard-error rate of the LLMs, i.e., the fraction of puzzles for which $P > 0.5$. The main point is the same: the error rate grows smoothly with n, and the control error rate decays. Observe further that this hard-error plot is nearly identical to the soft $P(\texttt{Yes})$ plot, indicating that the law of P is concentrated near $\{0, 1\}$, as we plot in §B.

Conclusion and Significance. Together, these results establish the following fact: even when an LLM can disqualify a candidate answer in isolation, when asked to 'AND' this disqualifying fact with other true facts, it instead falsely begins to accept the candidate answer, which the rate of acceptance itself growing smoothly with the number of distractor classes. More generically, when asked to evaluate a given candidate on a given set of properties, an LLM can fail to employ a known disqualifying property, instead getting distracted by properties in its context that support acceptance.

As discussed in §1, this basic structure holds significance because it embeds quite generically in inference and reasoning tasks. Such distractions, when introduced into high-level reasoning, can cause LLMs to ignore disqualifying criteria for lines of argument (even if they would detect them in, say, unit tests), and lead to downstream faulty output. Thus, this distractability points at a mode by which hallucination may occur, and invites future investigation into distractions in richer tasks.

3.3 Effects of Qualitative Aspects of Categories and Prompts

The setup above draws disqualifiers and distractors, and permutes them arbitrarily. However, it qualitative properties of both categoreis and placement can affect distraction, as show below. For the sake of space, we only present plots for Qwen-14B in the main text, and leave plots for the remaining models to Appendix.C.2. The conclusions remain the similar for all models.

Category Informativeness. Not all categories are equally informative. For instance, membership in a very niche category is a more informative predicate than membership in a broad category (e.g., *French River* v/s *European River*). In other words, the smaller the *size* of a cateogry, the more informative it is. We study how category informativeness affects the distraction of LLMs, and find that more informative categories have systematically greater importance to LLM decisions.

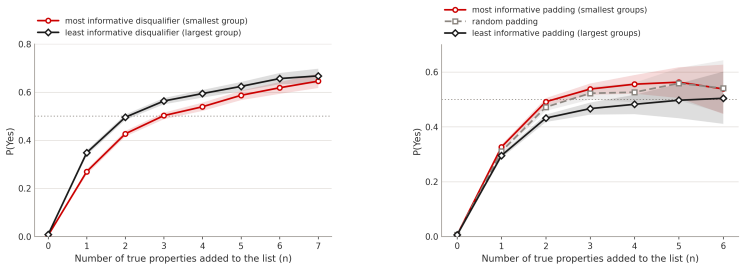


Figure 3: **Effect of category informativeness on distraction for Qwen-14B.** Informative disqualifiers are associated with less acceptance (*Left*), while informative distractors lead to greater acceptance (*Right*).

Informative disqualifiers attenuate distraction. First, we ask: *is negative evidence from more informative disqualifiers 'worth more' to the LLMs?* To study this, we restrict to (T, W) pairs where $|\mathcal{C}_{\text{mis}}(W;T)| \geq 3$, and separately evaluate performance with the smallest disqualfier in this set (most informative), and the largest (least), presenting the same distractors to each. Fig. 3 answers us affirmatively: low-information disqualfiers see about a 15% larger P than high-information ones.

Informative distractors boost distraction. Similarly, we ask: *is positive evidence from more informative distractors 'worth more' to the LLMs?*. To this end, we restrict attention to (T, W) and n where $|\mathcal{C}_{\text{sh}}(T;W)| \geq 2n$, and present either the n smallest distractors, the n largest distractors, or a random choice (as control). Again, the answer is yes: Fig. 3 shows that the most-informative distractors increase P by about a $5-10\%$ factor beyond least-informative distractors. Interestingly, the random distractor curve lies very close to the most-informative curve, suggesting that even a few highly informative distractors can elicit this boost.

Disqualifier Placement. Due to the causal attention structure of the LLMs studied, the evolution of states corresponding to categories presented earlier in the list c_0^n can affect those following, but not vice versa. Thus, it is possible that an 'internal evaluation' of the disqualifier changes due to its state attending to those of true categories preceding it, which may explain the observed distraction effect.

We control for this by simply placing the disqualifier first in the list, which ensures that the disqualifier-state evolves independently of all other categories. Thus, we conduct three experiments where we place the disqualifier (i) first (i.e., $c_0 = d$), (ii) last (i.e., $c_n = d$), and (iii) in the middle (i.e., $c_{\lfloor (n+1)/2 \rfloor} = d$). Fig. 4 shows the P curves for these scenarios. There are two main observations:

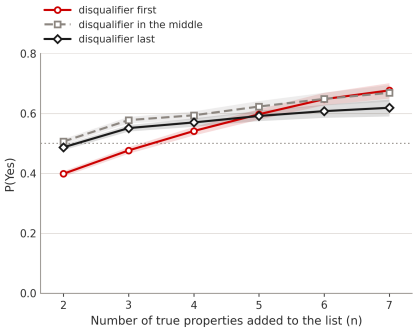


Figure 4: **Variation of distraction with disqualifier placement.** For small n, placing the disqualifier after the distractors leads to an increase in both P and hard errors.

Firstly, disqualifier position indeed changes the degree of failure, in that for small n, P is largest when the disqualifier is at the end. Secondly, even with the disqualifier first, the failure rate remains high, and again grows with n. In fact, the two curves saturate to the same level (up to noise) at large n, and the separation between these curves is at most 0.1 Thus, while distractors early in the context do affect the score P, this alone cannot explain the bulk of the distraction effect.

4 Distraction in Freeform Generation

§3 studied a verification version of catfishing puzzles by revealing a candidate answer, and measuring the yes/no response to the prompts. However, LLMs are often used in free-form generation mode. This section aims at eliciting whether LLMs are distractible in this mode as well: we ask how the LLMs' preference between wrong and correct answers shifts as the number of distractors grows.

Design. The main challenge with studying free-form answers is the large output space. To handle this, we pose the problem in free-form mode, but evaluate only the relative probabilities of a target T, and a wrong answer W. Specifically, we draw a triple (T, W, c_0^n) as in §3.1, give only the categories to the LLM, and ask `Which Wikipedia article belongs to all of these`

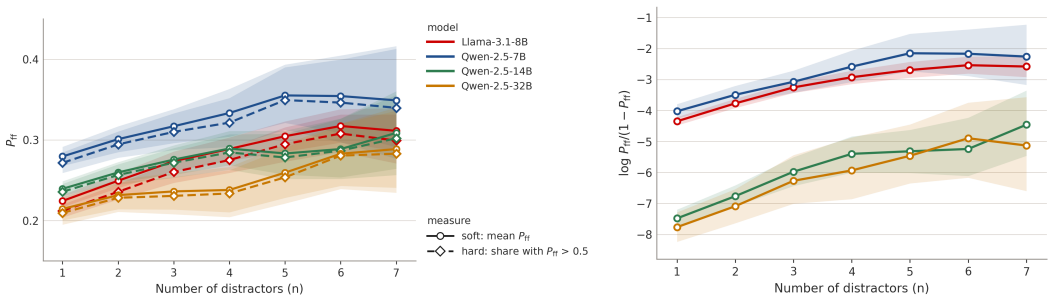


Figure 5: **Distractions in Free-Form Generation.** *Left.* Both soft preference of W over T (P_{ff}) and hard errors (instances wherein $P_{\text{ff}} > 0.5$) grow as the number of distractors increases. *Right.* The growth of the average log-odds ratio $\log P_{\text{ff}}/(1-P_{\text{ff}})$ demonstrates the same distraction effect in detail.

`groups?` Responses are scored by $P_{\text{ff}}(W;T,c_0^n) = \pi(W)/(\pi(T)+\pi(W))$, where π is the LLM output distribution. We leave an explicit description of the prompt to §D for the sake of space.

Notice that since the generation mode is free-form, and the wrong answer is still atomically disqualified, this experiment is closer to the usual manifestation of a hallucination, although we note that we are still measuring only the changes in a restricted answer space.

Observations. Fig. 5, *Left*, presents the results for the same four models as in §3. At a high level, we see that while the true target answer is, on average, preferred to the wrong answers, the degree of preference sees a marked decay with n. This apprears also in the hard-error, where we measure the fraction of puzzles where $P > 0.5$, i.e., W is preferred to T—the rate of this rises from 21% at $n = 1$ to 28% at $n = 7$ for Qwen-32B. Thus, we draw the conclusion that the distractibility we saw in the verification version of the puzzles manifests also in this free-form generation mode, and that the LLMs suffer an increasing rate of distraction has the size of the distracting cohort grows.

Controlling for Answer Length. One possible confounder for these observations is the token lengths of T and W: if true answers were systematically longer than wrong answers, we would expect an artificial boost in P_{ff}. However, for a fixed pair (T, W), the effect of their length difference does not depend on n. Thus the *change* of the log-odds $\log P_{\text{ff}}/(1-P_{\text{ff}})$ with n, measured within each pair relative to $n = 1$, is a controlled measure of the distraction effect. Fig. 5, *Right*, shows the average log-odds, which rises with n for every model; the per-pair change relative to $n = 1$ (App. D, Fig. 13) shows the same growth, confirming that it is not an effect of answer length.

5 Analysis

Having established that true properties in the context can distract an LLM from using a known disqualifier, we now explore the modes by this can happen. A catfishing puzzle in the Yes/No form of §3.1 demands the answering the AND of individual cateogry membership predicates:

$$(W \in c_0?) \wedge (W \in c_1?) \wedge \cdots \wedge (W \in c_n?),$$

There are two natural ways in which the evaluation of this formula can fail:

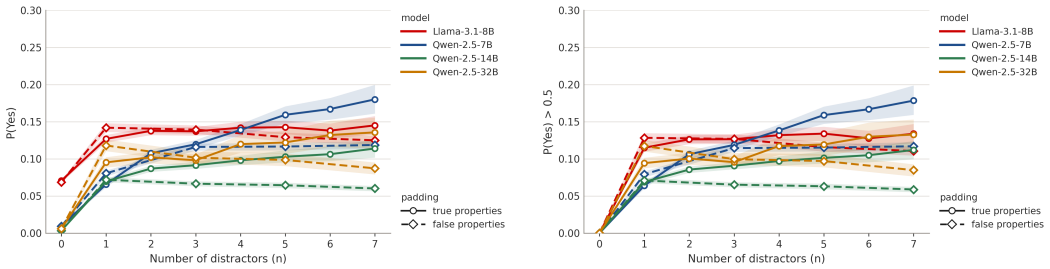
- **Logic Errors**: the predicates are resolved correctly, but their composition via AND fails.
- **Fact Corruption**: in the presence of true distractors, the disqualifier evaluation is corrupted, i.e., the disqualifier predicate $(W \in d?)$ is computed as `True`.

While logic errors have been previously observed multi-hop QA (e.g. Biran et al., 2024), atomic fact corruption is directly linked to our setting, and so our study focuses on these, in two experiments.

5.1 Explicit Disqualifier Resolution

We take the design of §3.1, but ask the LLM to resolve *only the disqualifier-membership for* W by changing the last line of the prompt to "`Does <candidate> belong to the group <disqualifier>?`." This allows us to check how the resolution of $(W \in d?)$ changes as positive distractors are added to the context. Fig. 6 presents the main observations.

Fact Corruption. The first observation is that progressive fact corruption, in the sense of an increase in P with n, does indeed occur for the Qwen models: P rises from near zero up to about 0.15 for



(a) P(Yes) score

(b) Hard-error rate

Figure 6: **Fact Corruption In Explicit Disqualifier Resolution.** Variation of the `Yes` score P for the disqualfier only as n varies (*Left*), and the accompanying hard-error rates (*Right*) for the four models presented in §3.2. Observe that for Qwen models, there is progressive corruption of the disqualification fact in presence of increasing true context, while the control curves again exhibit decay. Nevertheless, the rate of this corruption is significantly smaller than the rate of overall distraction (c.f. Fig. 2). Prompt counts are presented in §E.

these as n reaches 7, and the hard error plots likewise grow to 10-15% at $n = 7$ (again exhibiting the concentration we saw in Fig. 2). These observations suggest that fact corruption indeed occurs in these models, and its extent depends on the number of true facts in the context.

Fact corruption rate is smaller than distraction rate. Although fact corruption does appear to exist, its extent is much smaller than the distraction rates in §3.2. Indeed, comparing Fig. 6 to Fig. 2, we see that over half of the overall answers are wrong, but only about 15–18% of the individual disqualifiers are incorrectly resolved with the list. Thus, it is clear that, at least to the extent this experiment detects corruption, it cannot identify it as the sole source of these errors.

Confounder: asking ALL may change individual evaluation. A significant confounder in this experiment is that the prompt only demands the resolution of the disqualifier fact, whereas in §3 the prompt demanded the resolution of the conjunction. Since evaluating the conjunction requires one to resolve all category memberships, this may introduce a distinct mode of computation wherein the process of resolving the truth value of one category interferes with the process of resolving this for another. This distinction can, in principle, explain the entirety of the difference we see between the corruption rates in this section, and the high overall distraction rates in §3.2.

5.2 Full Inline Resolution

To control for differences between demanding resolution of one category versus all together, as well as to explicitly allow us to investigate logic errors, we employ a prompt format that demands that the model explicitly resolves each membership before predicting their joint AND in an *inline* format.

Setup. This is done through the following prompt.

```
Consider "<candidate>".
For each group below, say whether "<candidate>" belongs to it.
Then say whether "<candidate>" belongs to ALL of them at once.
1. <category 0>
– ...
<n+1>. <category n>
1. <ans> 2. <ans> ... <n+1>. <ans>  All: <ans>
```

Two worked examples precede the prompt to fix the format in context. The answers are computed thus: in order, we issue the prompt up to the first unfilled `<ans>` slot, compute the LLM score P for this, and append the most likely answer. Each verdict is thus conditioned on the ones the model has already written, and the conjunction is judged in the presence of explicitly resolved predicates. To prevent confounding of the disqualifier predicate resolution from the distractor-answers, **we place the disqualifier first**. §F reports further experiments without this disqualifier control.

Results. Fig. 7 presents the main results. Observations are divided along 3 bits: the final 'All' answer, the disqualfier resolution, and whether any distractor is resolved as No. We show fraction of prompts with these 8 cases as bar charts.

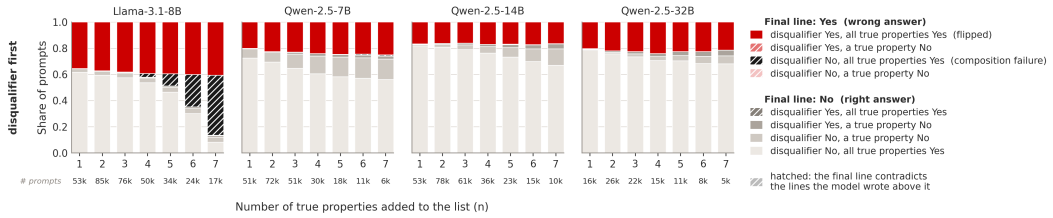


Figure 7: **Fact Corruption and Logic Errors in Full Inline Resolution.** Fraction of prompts with varying 'All', disqualifier, and distractor resolutions are shown. Logic errors ('All' disagrees with the individual predicates) are marked in hatched lines. Disqualifiers are always placed first in these experiments. Distraction (All = `Yes`) is dominated by fact corruption, except for Llama-8B at large n, where logic errors become prevalent. Also note the corruption of true facts growing with n.

Logic Errors are Rare, and Corruption Dominates. A key observation is that for all models, for $n \leq 4$, the rate of logic errors is practically 0, and instead all distractions are accompanied by a fact corruption of the disqualifier. This observation continues to hold for the Qwen models at large n, but Llama-8B shows increasing logic errors for $n \geq 5$, with these errors dominating distractions when $n = 7$. Thus, at least in this inline mode, distractibility of LLMs stems from fact corruption for Qwen models, and mostly so for Llama.

It is also interesting to note that distractors also suffer from corruption: for large n (and especially for Qwen7B), it is common for both a distractor and a disqualifier to be corrupted together. In the absence of logic errors, this yields a correct All=`Yes` resolution, but for the wrong reasons. This observation not only underscores that fact corruption is more prevalent than exhibited only in the distraction cases above, and also that evaluation of all categories can influence one another.

Corruption with the knowledge of AND evaluation. We again note since the disqualifier was placed first, the fact corruption of disqualifiers cannot be occurring due to the model relying on explicit resolution of distractor memberships. Therefore, these corruptions are essentially arising from the presence of the list of groups, along with the information that the prompt demands resolving all category memberships. While the prompt structures are of course distinct, it is interesting to compare this with Fig. 6. In particular, observe that, especially for small n, the rate of fact corruption is noticeably higher (from 10% for Qwen models to 20%).

Overall. These observations paint a fairly distinct picture: at least when n is small, LLMs are reliably able to compute the AND of n Yes/No predicate explicitly presented in their context, and thus the bulk of distraction is induced by fact corruption. Further, modulo prompt differences, the high-level goal indicated to the model also appears to affect fact corruption rates.

Confounders. We note that this experiment is also confounded when comparison with results in previous sections is concerned. First, the change in the prompt makes direct comparison with prior experiments unclear. Second, since the 'All' Yes/No resolution can depend on explicit category-level conclusions, the lack of logic errors in this experiment does not mean that logic errors do not occur in earlier distraction experiments. This potentially explains the difference in base rate of distractions.

6 Discussion

We establish that in the presence of true properties in the context, an LLM can fail at a simple compositional task even when it includes a disqualifier that the LLM could atomically detect. Moreover, this phenomenon entails some degree of fact corruption, whereby the model comes to endorse a false fact that it could atomically reject in the presence of other true distractor properties in the context. Distracibility manifests both in verification and generation versions of the catfishing-style puzzles we study, and is modulated in part by order of presentation of categories, and by their informativeness. This distractibility presents a concrete mode which is a plausible way for wider hallucination may occur, and warrants further study to judge whether this indeed occurs in other tasks.

Limitations. There are two main limitations to this research. First, we are limited by the breadth of the catfishing data, and the extent to which we could explore it (the main barrier being the computational costs of evaluating prompts). Second, our study is behavioural: we identify the phenomenon of distractibility, but we do not locate internal mechanisms behind it. While beyond the scope of the present paper, this is a fascinating direction for future work.

AI USE STATEMENT

We used generative AI tools to help write code for the experiments and figures, and to edit the writing. We did not use them for any other research subtask. The authors reviewed all AI-assisted code and text and take full responsibility for the content of this paper.

ETHICS STATEMENT

The catfishing dataset involves human-generated data, both in terms of curation of puzzles, and wrong answers. The latter are presented to us in an entirely anonymised way: we simply receive the list and frequency of wrong answers, and no information about the identity of people who wrote them. The puzzles were scraped with explicit permission from their curator.

REPRODUCIBILITY STATEMENT

The principal barrier to reproducibility is that we cannot share the dataset scraped from `catfishing.net`, since we did not receive permission to do so from the owner of this website. Nevertheless, we note that the entire archive is publicly accessible, and permission to collect the data may be requested from the owner.

REFERENCES

Catfishing. `https://catfishing.net`, 2026.

Neeladri Bhuiya, Viktor Schlegel, and Stefan Winkler. Seemingly plausible distractors in multi-hop reasoning: Are large language models attentive readers?, 2024. URL `https://arxiv.org/abs/2409.05197`.

Eden Biran, Daniela Gottesman, Sohee Yang, Mor Geva, and Amir Globerson. Hopping too late: Exploring the limitations of large language models on multi-hop queries, 2024. URL `https://arxiv.org/abs/2406.12775`.

Eunseong Choi, June Park, Hyeri Lee, and Jongwuk Lee. Conflict-aware soft prompting for retrieval-augmented generation, 2025. URL `https://arxiv.org/abs/2508.15253`.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93, January 2024. ISSN 1946-5319. doi: 10.1093/jla/laae003. URL `http://dx.doi.org/10.1093/jla/laae003`.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems*, 43(2):1–55, January 2025. ISSN 1558-2868. doi: 10.1145/3703155. URL `http://dx.doi.org/10.1145/3703155`.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38, March 2023. ISSN 1557-7341. doi: 10.1145/3571730. URL `http://dx.doi.org/10.1145/3571730`.

Mandar Joshi, Eunsol Choi, Daniel S. Weld, and Luke Zettlemoyer. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension, 2017. URL `https://arxiv.org/abs/1705.03551`.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom

Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know, 2022. URL https://arxiv.org/abs/2207.05221.

Adam Tauman Kalai and Santosh S. Vempala. Calibrated language models must hallucinate, 2024. URL https://arxiv.org/abs/2311.14648.

Adam Tauman Kalai, Ofir Nachum, Santosh S. Vempala, and Edwin Zhang. Why language models hallucinate, 2025. URL https://arxiv.org/abs/2509.04664.

Yubin Kim, Hyewon Jeong, Shan Chen, Shuyue Stella Li, Chanwoo Park, Mingyu Lu, Kumail Alhamoud, Jimin Mun, Cristina Grau, Minseok Jung, Rodrigo Gameiro, Lizhou Fan, Eugene Park, Tristan Lin, Joonsik Yoon, Wonjin Yoon, Maarten Sap, Yulia Tsvetkov, Paul Liang, Xuhai Xu, Xin Liu, Chunjong Park, Hyeonhoon Lee, Hae Won Park, Daniel McDuff, Samir Tulebaev, and Cynthia Breazeal. Medical hallucinations in foundation models and their impact on healthcare, 2025. URL https://arxiv.org/abs/2503.05777.

Yunseo Lee, John Youngeun Song, Dongsun Kim, Jindae Kim, Mijung Kim, and Jaechang Nam. Hallucination by code generation llms: Taxonomy, benchmarks, mitigation, and challenges, 2025. URL https://arxiv.org/abs/2504.20799.

Moxin Li, Yong Zhao, Wenxuan Zhang, Shuaiyi Li, Wenya Xie, See-Kiong Ng, Tat-Seng Chua, and Yang Deng. Knowledge boundary of large language models: A survey, 2025. URL https://arxiv.org/abs/2412.12472.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023. URL https://arxiv.org/abs/2307.03172.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories, 2023. URL https://arxiv.org/abs/2212.10511.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. Llms know more than they show: On the intrinsic representation of llm hallucinations, 2025. URL https://arxiv.org/abs/2410.02707.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H. Miller, and Sebastian Riedel. Language models as knowledge bases?, 2019. URL https://arxiv.org/abs/1909.01066.

Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model?, 2020. URL https://arxiv.org/abs/2002.08910.

Yufei Tao, Adam Hiatt, Erik Haake, Antonie J. Jetter, and Ameeta Agrawal. When context leads but parametric memory follows in large language models, 2024. URL https://arxiv.org/abs/2409.08435.

Siye Wu, Jian Xie, Jiangjie Chen, Tinghui Zhu, Kai Zhang, and Yanghua Xiao. How easily do irrelevant inputs skew the responses of large language models?, 2024. URL https://arxiv.org/abs/2404.03302.

Minglai Yang, Ethan Huang, Liang Zhang, Mihai Surdeanu, William Wang, and Liangming Pan. How is llm reasoning distracted by irrelevant context? an analysis using a controlled benchmark, 2025. URL https://arxiv.org/abs/2505.18761.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018. URL https://arxiv.org/abs/1809.09600.

Mert Yuksekgonul, Varun Chandrasekaran, Erik Jones, Suriya Gunasekar, Ranjita Naik, Hamid Palangi, Ece Kamar, and Besmira Nushi. Attention satisfies: A constraint-satisfaction lens on factual errors of language models, 2024. URL https://arxiv.org/abs/2309.15098.

A Dataset Details, and More on Catfishing

The catfishing puzzle was first proposed in 2006 by S. Harihareshwara, and K. Devan produced an early implementation of the game in the same year (see the FAQ page on `catfishing.net`). The game we draw our data from has been active since 2024. Questions are curated by the owner of this website, known mononymously as Matthew, and the process is exhaustively detailed at `https://catfishing.net/help`. All rights to the data we employ rest with the owner, and our use is with explicit permission, obtained by contacting them directly (details are available at `https://catfishing.net/contact`.

The main features of the data pertinent to our exploration were listed in §2. Below, we describe how the (target, wrong answer) pairs are built, and how often models reject wrong answers in isolation.

Building (target, wrong answer) pairs. We match each wrong answer to a Wikipedia page by its title, after light normalisation (lower case, no leading article, no trailing parenthetical, no punctuation), and drop titles that match more than one page.Among the wrong answers, a further 17,207 titles resolved to a Wikipedia page with known categories, beyond those already among the puzzle targets. We keep a wrong answer only if it was given by at least 5 players, differs from the target, and has a page with known categories. For each puzzle we take its 5 most frequent such wrong answers, and keep a pair only if the wrong answer shares at least 3 categories with the target and misses at least one ($|\mathcal{C}_{\text{sh}}(W,T)| \geq 3$, $|\mathcal{C}_{\text{mis}}(W;T)| \geq 1$). A wrong answer that fails this is not replaced by a less frequent one. This gives 7,811 candidate pairs. Each experiment then verifies the categories of every pair with the model (§3.1), and removes duplicate pairs, i.e., the same target and wrong answer appearing under different puzzles.

B Details for the Main Distractor Experiment

We present details omitted from §3.2 and §3.3.

Prompt. Each prompt lists the $n+1$ categories and asks for a single verdict about the candidate, as described in §3.1:

```
Groups:
- <category 0>
- ...
- <category n>

Does "<candidate>" belong to all of these groups? Answer Yes or No.
```

The prompt is preceded by the system prompt

```
You judge whether a given thing belongs to every one of the listed
groups. Answer with a single word: Yes or No.
```

and wrapped in each model's chat template. No in-context examples are used. The atomic check of the disqualifier ($n = 0$) uses the same system prompt, with the question `Does "<candidate>" belong to the group "<disqualifier>"? Answer Yes or No.` Responses are scored with $P(\texttt{Yes})$ as in equation 1.

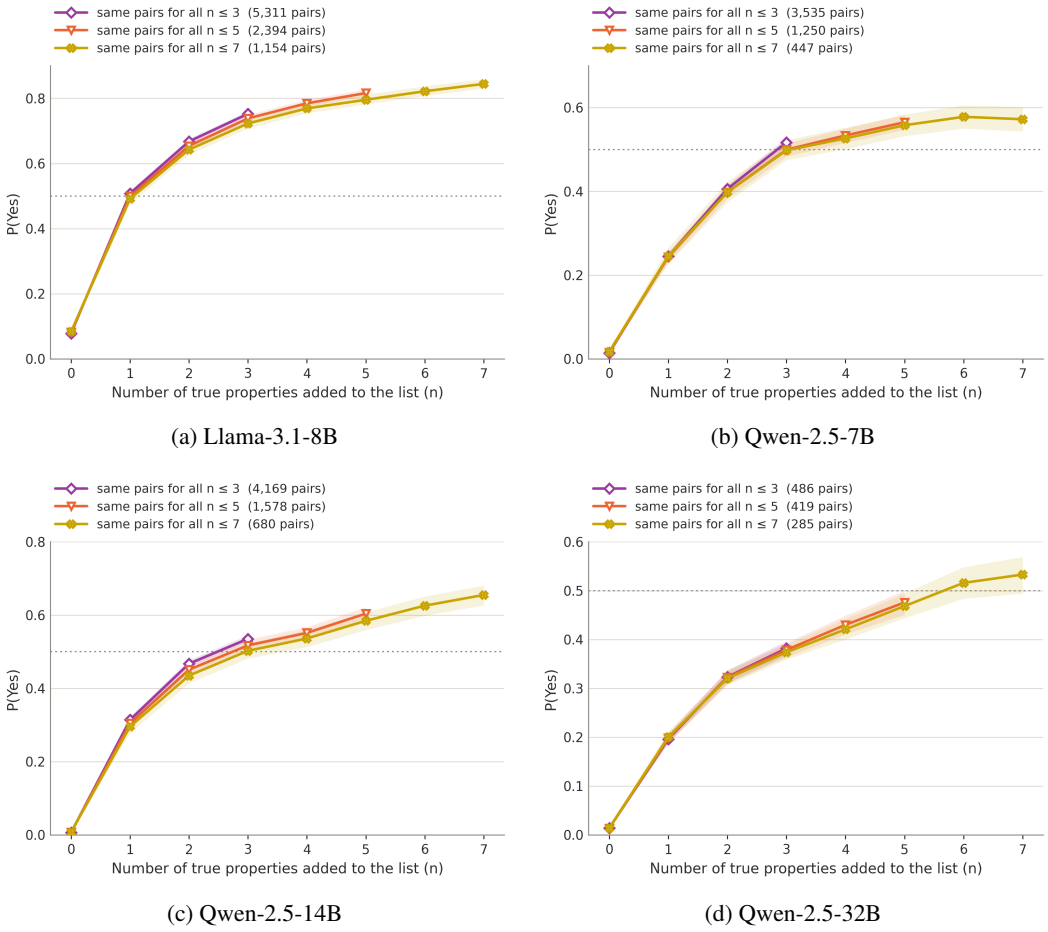
Number of prompts used in Fig. 2. While most plots display the number of prompts used to compute each point for each model and n, we omitted this from Fig. 2 for the sake of clarity. This is detailed in Table 1. The control arm is run only at $n \in \{0, 1, 3, 5, 7\}$. Qwen-2.5-32B was evaluated on a subset of about a quarter of the puzzles.

Distribution of $P(\texttt{Yes})$. As noted in §3.2, the hard-error rate and the soft $P(\texttt{Yes})$ nearly match, which indicates that $P(\texttt{Yes})$ is concentrated near 0 and 1. Fig. 10 shows this directly.

Same pairs at every n. Larger n requires more shared categories, so fewer (T, W) pairs are available as n grows. To rule out that the rise in P comes from this change in the set of pairs, Fig. 8 restricts to pairs available at every $n \leq k$, for $k \in \{3, 5, 7\}$: the rise is the same on these fixed sets.

Model	Arm	$n = 0$	1	2	3	4	5	6	7
Llama-3.1-8B	true	17,545	105,222	255,354	227,220	149,535	101,991	71,544	49,608
	false	17,372	104,164	–	239,817	–	210,591	–	181,332
Qwen-2.5-7B	true	17,163	102,762	216,705	153,693	90,276	53,967	32,883	19,431
	false	17,091	102,514	–	246,471	–	225,990	–	203,103
Qwen-2.5-14B	true	17,643	105,594	235,239	181,641	108,210	68,775	45,099	28,638
	false	17,576	105,416	–	253,452	–	233,010	–	209,583
Qwen-2.5-32B	true	5,426	32,586	77,793	66,492	45,630	33,273	23,220	16,335
	false	5,399	32,408	–	76,593	–	69,390	–	61,254

Table 1: Number of prompts behind each point of Fig. 2, per model, arm (true or false padding) and n. $n = 0$ is the disqualifier asked alone.



(a) Llama-3.1-8B

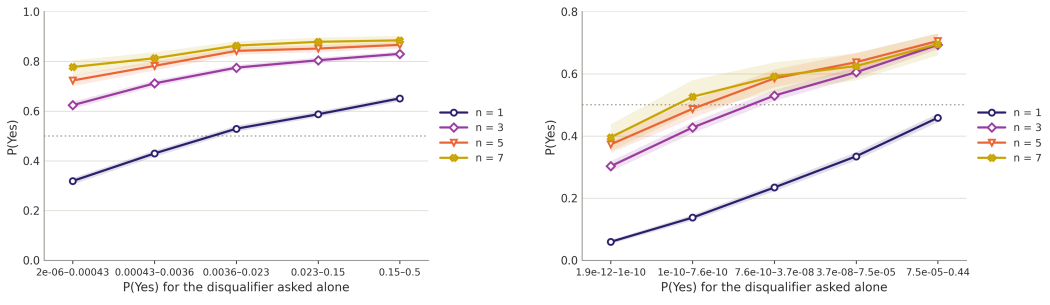
(b) Qwen-2.5-7B

(c) Qwen-2.5-14B

(d) Qwen-2.5-32B

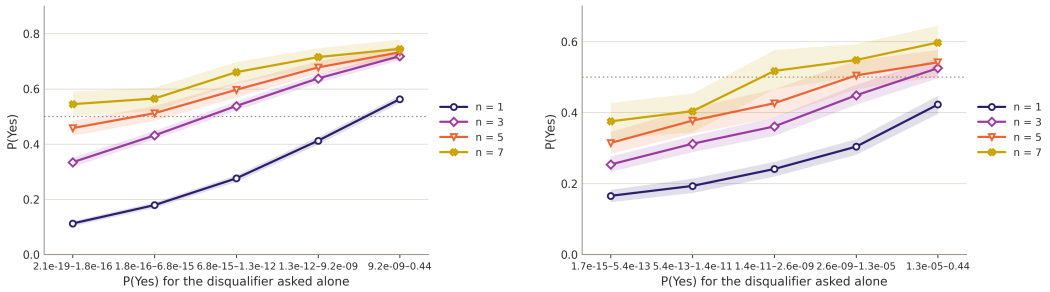
Figure 8: **Fixed sets of pairs, per model.** $P(\texttt{Yes})$ against n, computed only on pairs available at every $n \leq k$. Shaded bands are 95% CIs.

How firmly the disqualifier is rejected alone. All disqualifiers pass the atomic check ($P(W; d) < 0.5$), but some are rejected more firmly than others. Fig. 9 splits prompts by the disqualifier's atomic score into equal-count bins and plots P in each bin for $n = 1, 3, 5, 7$: Weakly rejected disqualifiers are more easily overridden: in every model, P increases from the most firmly to the least firmly rejected bin. However, distraction is not limited to borderline facts. Even in the most firmly rejected bin, where the atomic score is as low as 10^{-16} for the Qwen models, P rises with n and reaches 0.37–0.55 for the Qwen models and 0.78 for Llama-3.1-8B at $n = 7$. Thus the effect of §3.2 is not explained by facts the model only weakly knows.



(a) Llama-3.1-8B

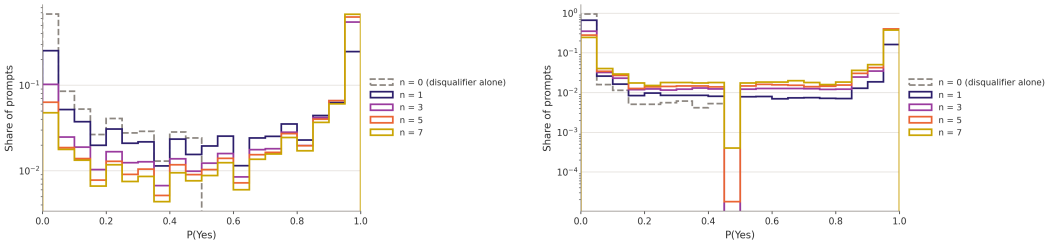
(b) Qwen-2.5-7B



(c) Qwen-2.5-14B

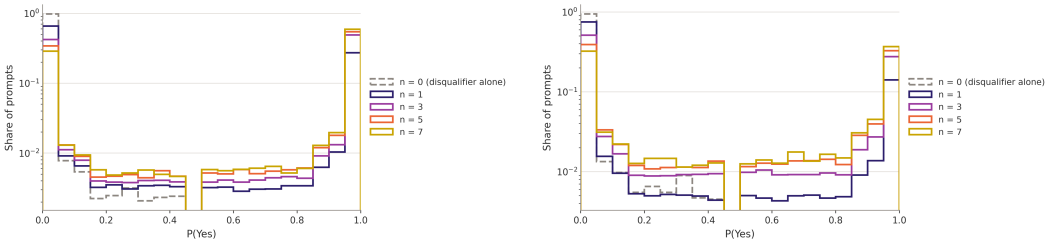
(d) Qwen-2.5-32B

Figure 9: **Distraction against the atomic score of the disqualifier, per model.** $P(\texttt{Yes})$ against $P(W; d)$, the score of the disqualifier asked alone (equal-count bins), for $n = 1, 3, 5, 7$. Shaded bands are 95% CIs.



(a) Llama-3.1-8B

(b) Qwen-2.5-7B



(c) Qwen-2.5-14B

(d) Qwen-2.5-32B

Figure 10: **Distribution of $P(Yes)$, per model.** Share of prompts in each bin of width 0.05, for $n = 0, 1, 3, 5, 7$ (log scale); each pair counts once at each n. The mass sits near 0 and 1, so the hard-error rate closely tracks the mean of $P(\texttt{Yes})$.

C Details on the Effects of Qualitative Aspects of Categories and Prompts

The main text presented plots capturing the effect of category informativeness and position on distraction only for Qwen-2.5-14B, primarily for the sake of space. Below, in §C.2, we provide the

same plots for the remaining three models we study. The overall conclusions remain the same for all models.

Before that, however, we describe three other qualitative properties of puzzles and categories that we studied, which did not have a significant effect in our experiments.

C.1 Qualitative Properties of Puzzles Without a Significant Effect

We consider three properties:

- **Popularity of Wrong Answers**: *Are wrong answers given more frequently by humans more likely to suffer distraction as n grows?*
- **Size of $|\mathcal{C}_{\text{sh}}|$**: *Are wrong answers with many common categories more likely to suffer distraction as more shared categories are added to the prompts?*
- **Size of $|\mathcal{C}_{\text{mis}}|$**: *Are wrong answers that differ from a target in many categories more likely to suffer distraction as true shared categories are added to the prompts?*

Fig. 11 shows all three for Qwen-2.5-7B. For each n, the curves are nearly flat: none of the three properties changes P much, while P still grows with n itself. The other models show no or only a weak effect of these properties. Since pairs with few verified shared categories cannot be used at large n, some bins have no point at large n.

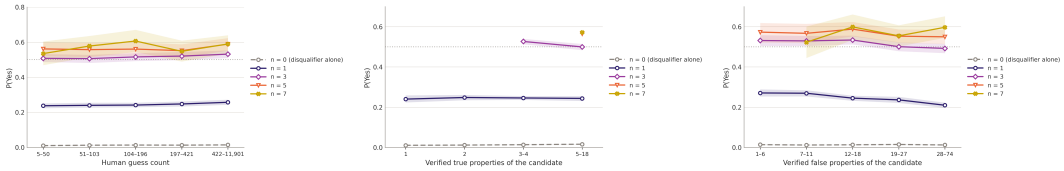
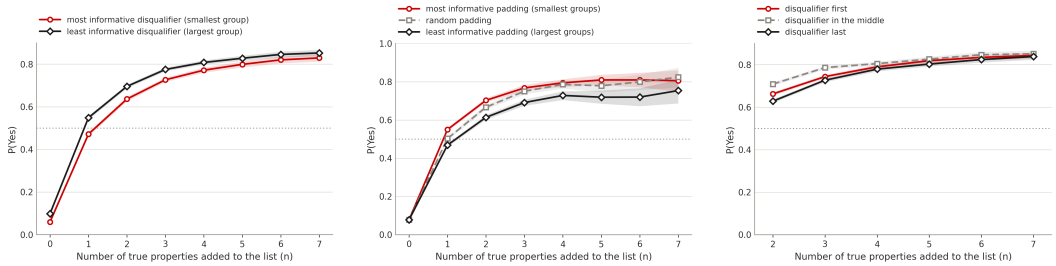


Figure 11: **Null effects.** Popularity of the wrong answer (*Left*), and the sizes of $\mathcal{C}_{\text{sh}}$ (*Middle*) and $\mathcal{C}_{\text{mis}}$ (*Right*) do not lead to variation in the distraction rate, as seen by the flat lines for each n for Qwen-2.5-7B. The distraction rate still varies with n itself, as seen by the vertical separation of the lines.

C.2 Further Plots of the Effect of Category Informativeness and Placement

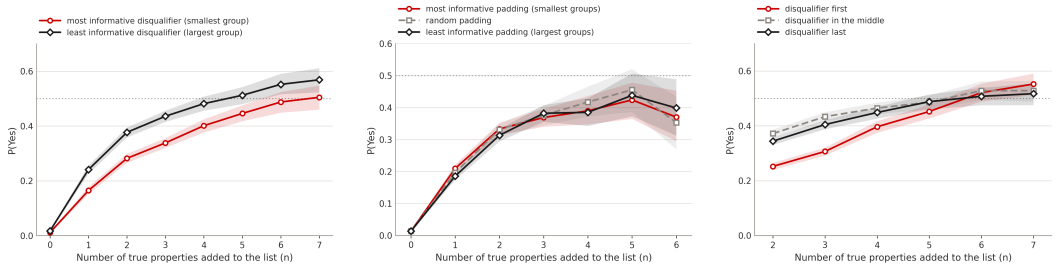
Fig. 12 repeats the plots of §3.3 for the remaining three models: disqualifier informativeness (*Left*), distractor informativeness (*Middle*), and disqualifier position (*Right*). The conclusions of §3.3 carry over. In every model, the least informative disqualifier leads to more false acceptance than the most informative one at every n with a relative gap of around 15%. Informative distractors raise P over uninformative ones by about 0.05–0.08 at larger n for Llama-3.1-8B and Qwen-2.5-7B, while for Qwen-2.5-32B the three curves overlap. For the disqualifier position, placing the disqualifier first gives the lowest P at small n in the Qwen models, while the highest curve is the disqualifier last for Qwen-2.5-7B and in the middle for Llama-3.1-8B and Qwen-2.5-32B; in all models, the curves come together by $n \approx 4$–6.



(a) Llama-3.1-8B



(b) Qwen-2.5-7B



(c) Qwen-2.5-32B

Figure 12: **Category informativeness and disqualifier position, remaining models.** Left: most against least informative disqualifier. Middle: most informative, random, and least informative distractors. Right: disqualifier first, in the middle, or last. $P(\texttt{Yes})$ against n; shaded bands are 95% CIs; the correct answer is NO at every n.

D Details for Distraction in Freeform Generation

Prompt. As stated in the main text, the triples (T, W, c_0^n) are selected in exactly the same way as in §3.1, and in particular these have one disqualifier and n distractors. These are fed to the LLM through a prompt of the following form:

```
Groups:
- <category 0>
- ...
- <category n>
Which Wikipedia article belongs to all of these groups?
Answer: <ans>
```

Here, the `<ans>` slot corresponds to the state where LLM generation begins. We use two in-context examples before this prompt to clarify the formatting of the puzzles to the models.

Scoring. The two considered answers of T and W are scored thus: let $\pi(W)$ denote the LLM's probability (given the above prompt) to generate W in the first $|W|$ iterated outputs, where $|W|$ denotes the token length of the string comprising W, and iterated output refers to autoregressive generation of $|W|$ tokens. Similarly, let $\pi(T)$ denote the same for the target T. We measure $P_{\text{ff}}(W, T; c_0^n) = \pi(W)/(\pi(T) + \pi(W))$.

D.1 Ablations

Answer length. The token lengths of T and W differ, which shifts

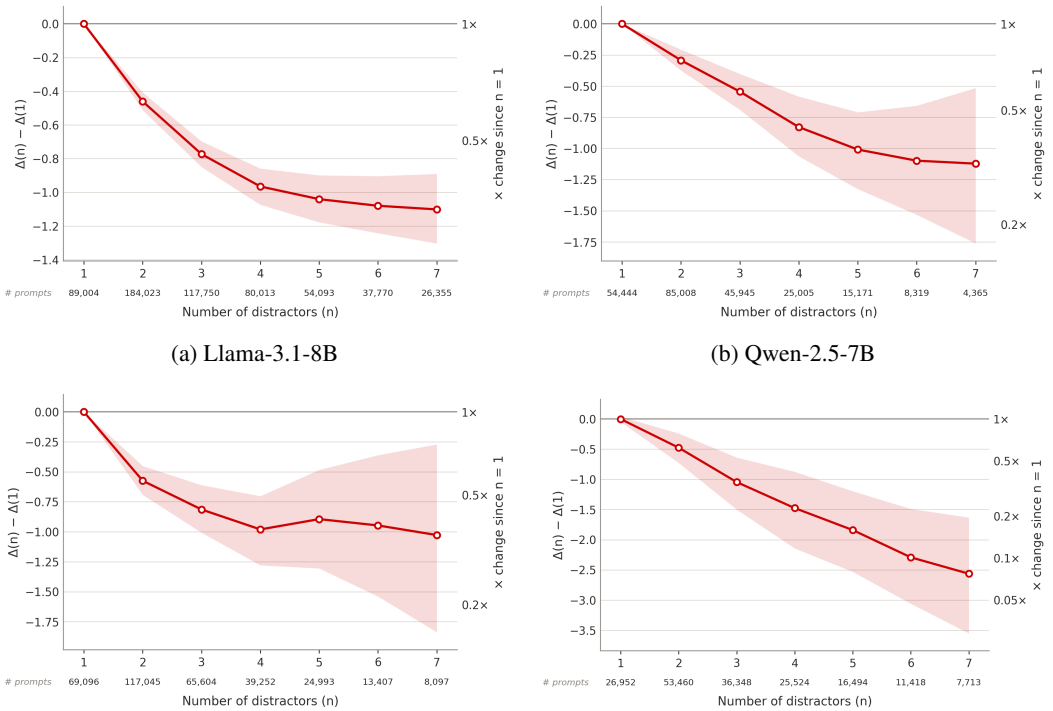
$$\Delta = \log(P_{\text{ff}}(T)/P_{\text{ff}}(W))$$

by an amount that does not depend on n. Fig. 13 shows $\Delta(n) - \Delta(1)$ per model, which cancels any such fixed offset: the decrease with n remains in every model.

Same pairs at every n. Larger n requires more verified distractors, so fewer pairs are available as n grows. To rule out that the trend comes from this change in the set of pairs, Fig. 14 restricts to pairs that are available at every $n \leq k$, for $k \in \{3, 5, 7\}$: the trend is the same on these fixed sets.

Disqualifier informativeness. Mirroring §3.3, Fig. 15 compares the most informative disqualifier (smallest category), a middle-sized one, and the least informative one (largest category), on the same distractors: In every model the three curves keep the same order: the most informative disqualifier keeps the preference for T highest, the least informative keeps it lowest, and the middle-sized one lies between them. This matches the verification setting (§3.3), where informative disqualifiers also attenuate distraction..

Disqualifier position. Fig. 16 places the disqualifier first, in the middle, or last in the list, for $n \geq 2$: Position has little effect: in all four models the three curves overlap within their confidence bands and decline with n at the same rate.



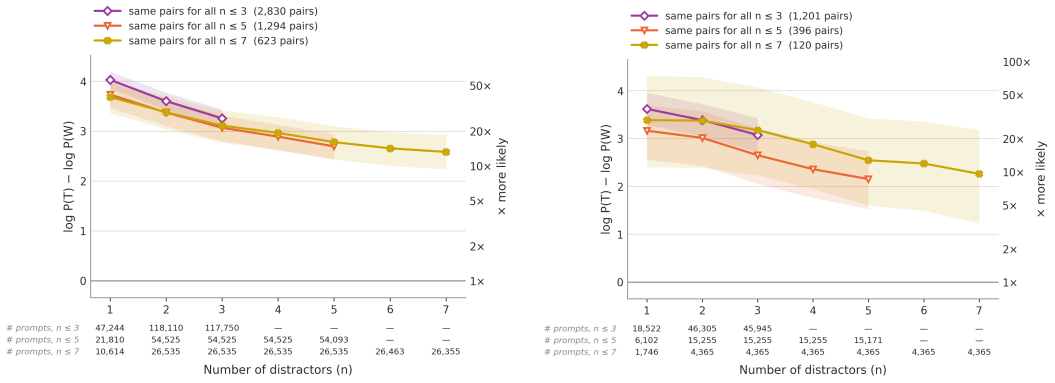
(a) Llama-3.1-8B

(b) Qwen-2.5-7B

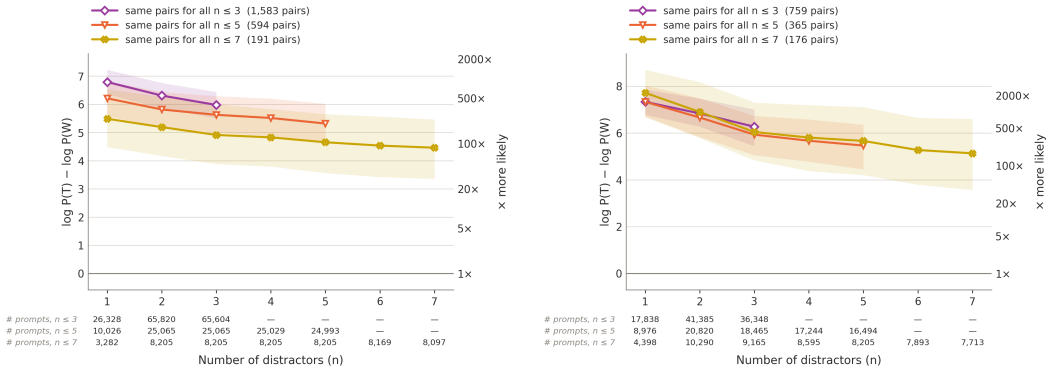
(c) Qwen-2.5-14B

(d) Qwen-2.5-32B

Figure 13: **How much the preference for the target shrinks, per model.** $\Delta(n) - \Delta(1)$: the change in the model's preference for the target T over the wrong answer W, relative to a list with one shared property. Below zero means the list brings the two answers closer. Shaded bands are 95% CIs; numbers below the axis are prompts per point.

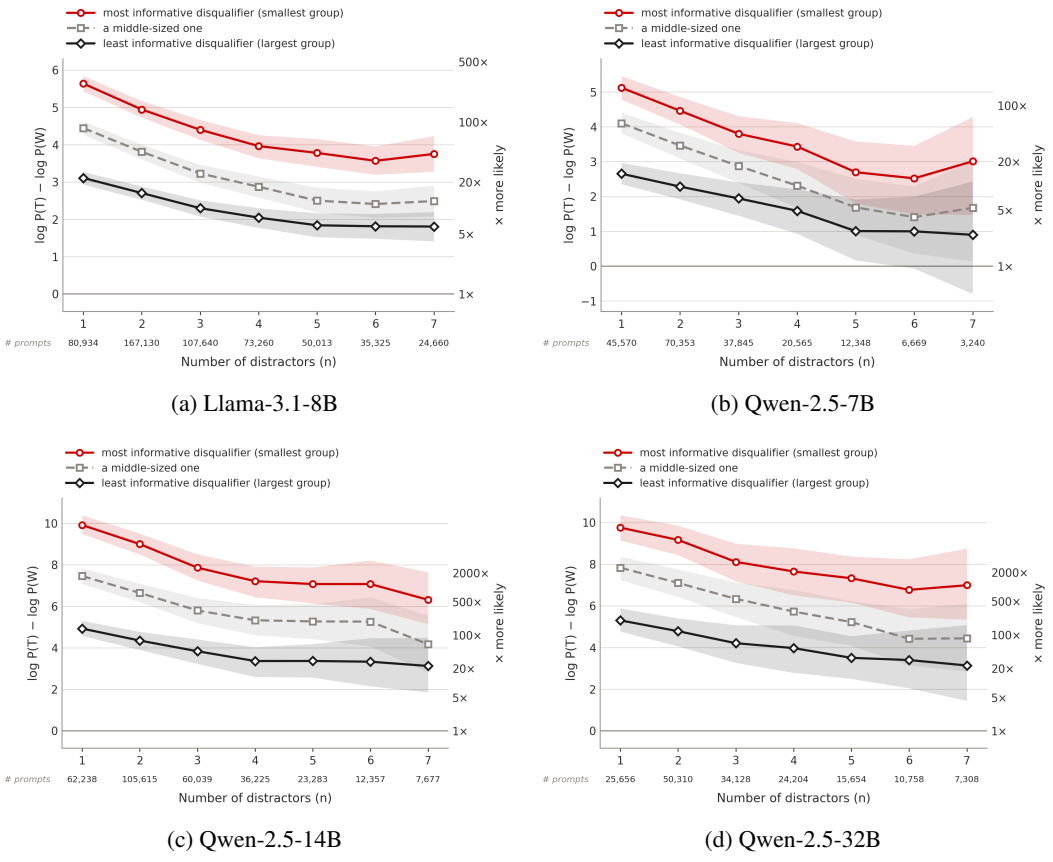


(a) Llama-3.1-8B (b) Qwen-2.5-7B



(c) Qwen-2.5-14B (d) Qwen-2.5-32B

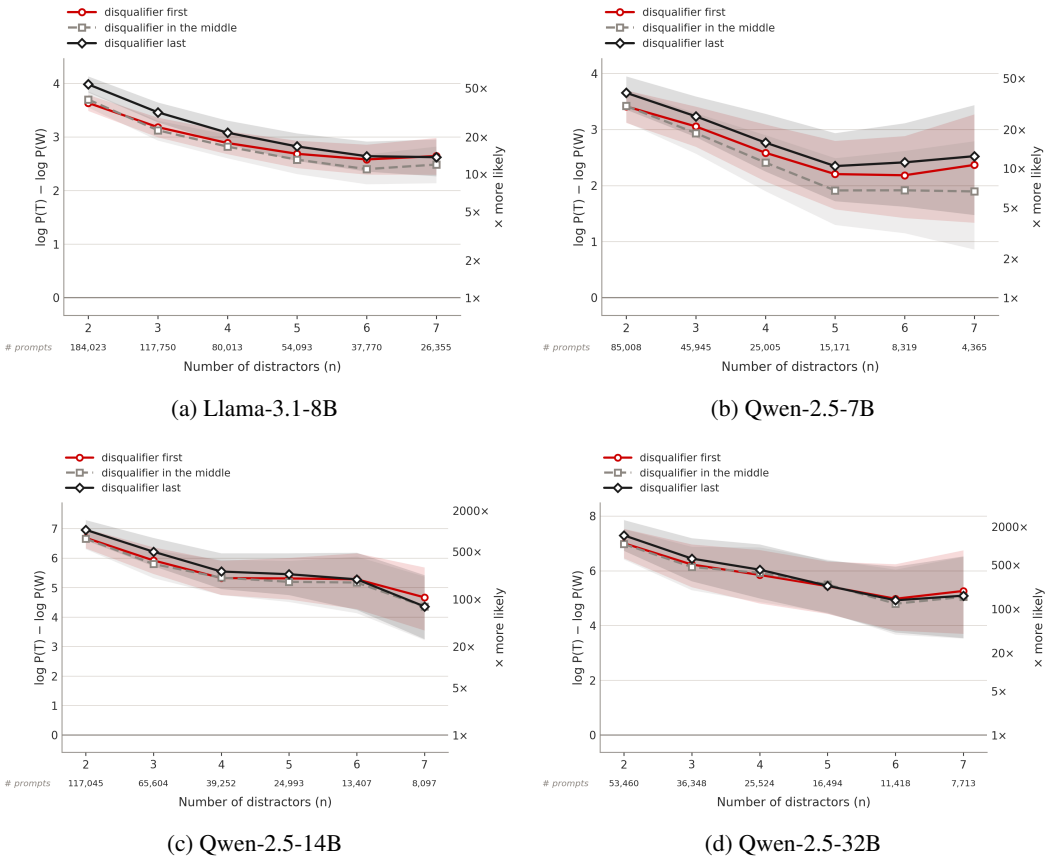
Figure 14: **Fixed sets of pairs, per model.** Δ against n, computed only on pairs available at every $n \leq k$. Shaded bands are 95% CIs.



(a) Llama-3.1-8B (b) Qwen-2.5-7B

(c) Qwen-2.5-14B (d) Qwen-2.5-32B

Figure 15: **Disqualifier informativeness, per model.** Δ against n for the most informative, a middle-sized, and the least informative disqualifier. Shaded bands are 95% CIs.



(a) Llama-3.1-8B

(b) Qwen-2.5-7B

(c) Qwen-2.5-14B

(d) Qwen-2.5-32B

Figure 16: **Disqualifier position, per model.** Δ against n with the disqualifier first, in the middle, or last. Shaded bands are 95% CIs.

E Details for Explicit Disqualifier Resolution

We describe the full prompt, the scoring, and the filtering used in §5.1.

Prompt. Every prompt of §5.1 reuses the same category list as the main experiment (§3.1), in the same order and with the disqualifier in the same slot. Only the final question changes, and it asks about the disqualifier alone:

```
Groups:
- <category 0>
- ...
- <category n>

Does "<candidate>" belong to the group "<disqualifier>"? Answer Yes or No.
```

The question line is word for word the atomic question used to verify the disqualifier. Unlike §3, which uses the system prompt given in App. B, the prompts here are preceded by

```
You answer questions about Wikipedia categories.
Answer with a single word: Yes or No.
```

and wrapped in each model's chat template. No in-context examples are used. Within this section, the $n = 0$ point is the atomic question asked under this same system prompt, so the only difference between $n = 0$ and $n > 0$ is the list shown above the question.

Scoring. Responses are scored with $P(\texttt{Yes})$ as in equation 1, read from the next-token distribution at the first answer position.

Re-verification of the disqualifier. The main experiment (§3.1) verifies disqualifiers under a different system prompt. To make sure that the $n = 0$ point of §5.1 is measured under the same conditions as the rest of the curve, we re-ask the atomic question $P(W; d)$ under the system prompt above, and keep only items for which the disqualifier is rejected ($P < 0.5$) under *both* system prompts. This keeps 76.8% of prompts for Llama-3.1-8B and 96.0–97.9% for the Qwen models. As a result, the hard-error rate is exactly 0 at $n = 0$ for every model, and any rise with n is caused by the list.

Control. The control arm is the same as in §3.1: the n added categories are further categories that W does not belong to, each verified in isolation, instead of shared ones.

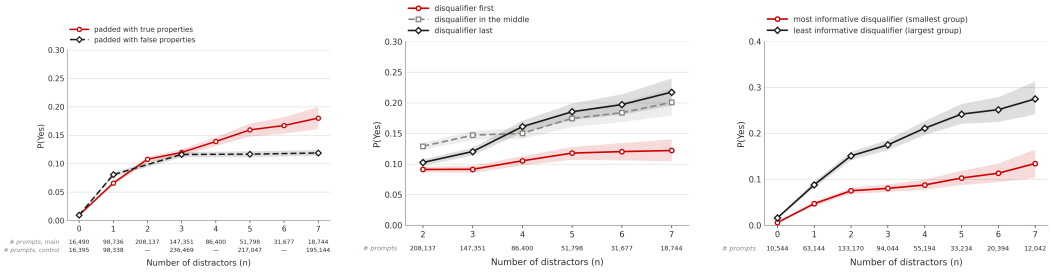
Number of prompts. Table 2 gives the number of prompts behind each point of Fig. 6, after re-verification.

Model	Arm	$n = 0$	1	2	3	4	5	6	7
Llama-3.1-8B	true	13,534	81,160	196,728	174,612	113,400	77,028	53,406	36,903
	false	13,444	80,616	–	185,790	–	163,146	–	140,628
Qwen-2.5-7B	true	16,490	98,736	208,137	147,351	86,400	51,798	31,677	18,744
	false	16,395	98,338	–	236,469	–	217,047	–	195,144
Qwen-2.5-14B	true	17,290	103,484	230,586	177,876	105,660	67,146	43,992	27,942
	false	17,223	103,298	–	248,316	–	228,198	–	205,197
Qwen-2.5-32B	true	5,259	31,572	75,303	64,257	43,917	31,902	22,317	15,684
	false	5,232	31,394	–	74,193	–	67,245	–	59,313

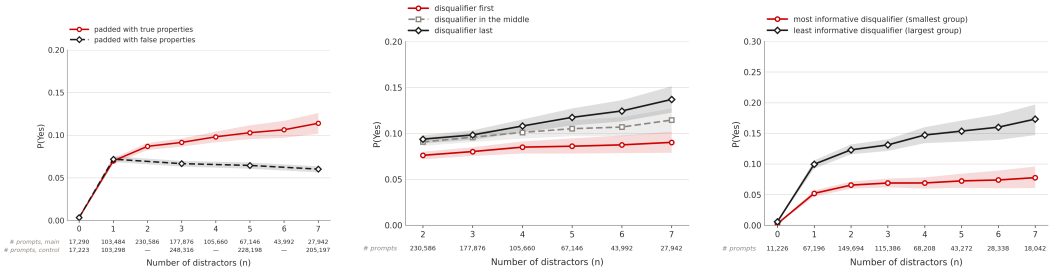
Table 2: Number of prompts behind each point of Fig. 6, per model, arm (true or false padding) and n, after re-verification of the disqualifier. $n = 0$ is the disqualifier asked alone.

E.1 Effects of Disqualifier Position and Informativeness

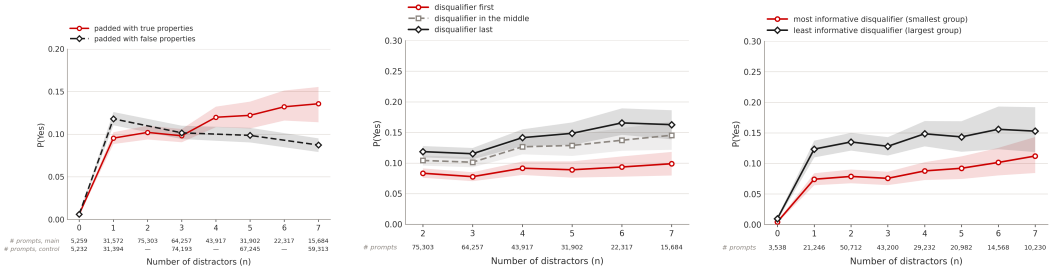
Fig. 17 repeats the ablations of §3.3 for the disqualifier-only question: true against false padding, the position of the disqualifier, and its informativeness. In the three Qwen models, true and false padding



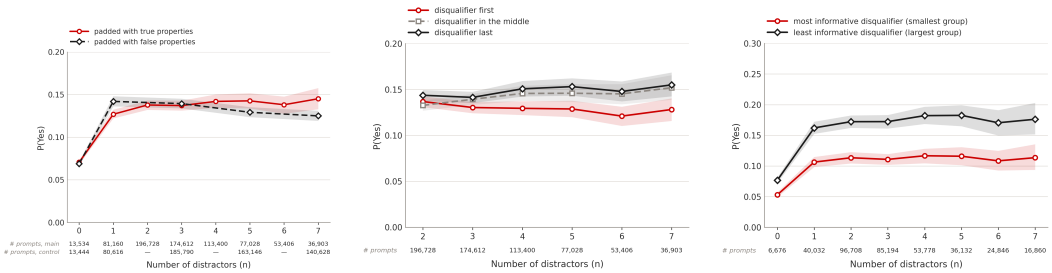
(a) Qwen-2.5-7B



(b) Qwen-2.5-14B

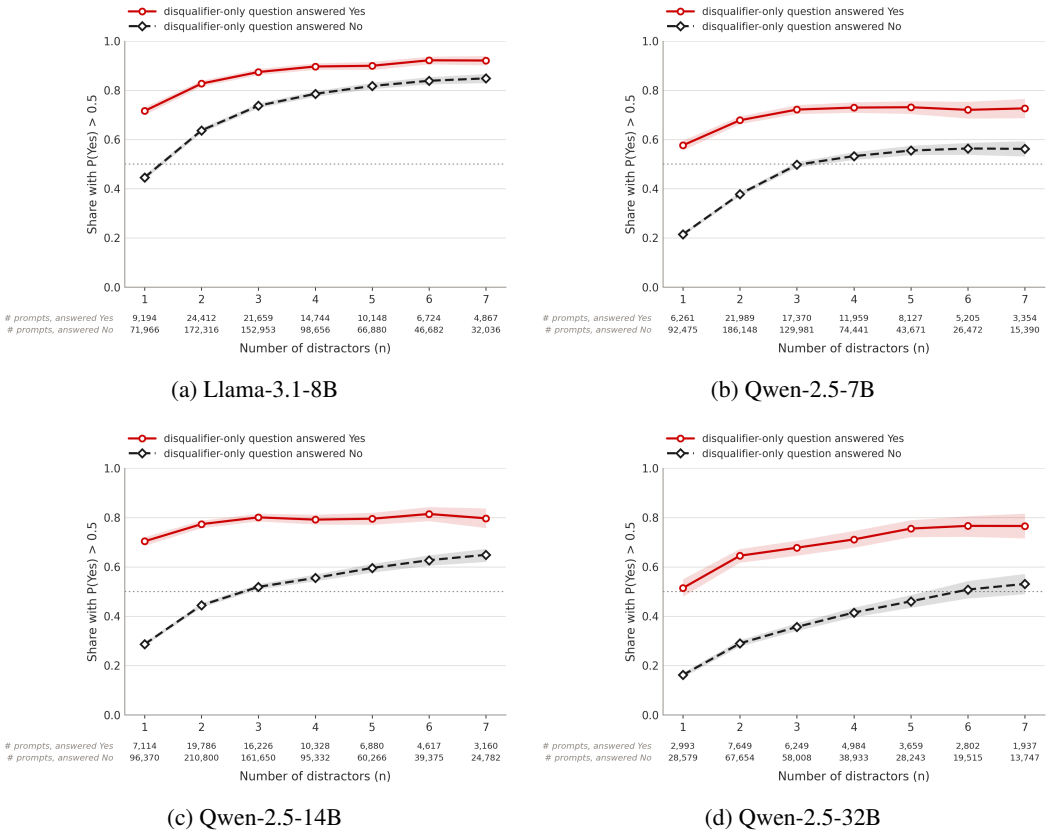


(c) Qwen-2.5-32B



(d) Llama-3.1-8B

Figure 17: **Disqualifier asked alone, per model.** $P(\texttt{Yes})$ when the model is asked only about the disqualifier, with the list shown above it. Columns: padding with true (solid) or false (dashed) categories; disqualifier first, in the middle or last; most and least informative disqualifier. The correct answer is NO at every n.



(a) Llama-3.1-8B

(b) Qwen-2.5-7B

(c) Qwen-2.5-14B

(d) Qwen-2.5-32B

Figure 18: **Corrupting the disqualifier makes the full verdict wrong more often.** For each prompt we ask both the disqualifier-only question and the full question on the same list, and split prompts by the answer to the first. Lines show the share whose full question is answered YES ($P(\texttt{Yes}) > 0.5$).

raise P by a similar amount up to $n \approx 3$, so list length alone accounts for the early rise; beyond that, true padding keeps raising P while false padding leaves it flat or lowers it. The disqualifier is accepted more often when it comes later in the list (with the disqualifier last, P reaches 0.22 for Qwen-2.5-7B at $n = 7$, against 0.12 with it first), and broad disqualifiers are accepted more often than narrow ones, with a gap that grows with n. In Llama-3.1-8B, P rises at $n = 1$ and then stays nearly flat for both padding types and all positions; only the informativeness of the disqualifier makes a clear difference.

E.2 JOINT COMPARISON OF DISQUALIFIER CORRUPTION AND ALL-DISTRACTION

Finally, we study whether puzzles wherein facts were corrupted in the explicit disqualifier resolution experiments have a systematically different distraction rate in the setting of our main results §3.2. Concretely, we ask: *if a puzzle displayed corruption in the explicit disqualifier resolution experiment, is it more likely to suffer distraction when a conjunction is queried?* Figure 18 shows that this is indeed the case. This further suggests that the fact corruption we observed indeed has a direct role to play in the distraction phenomenon.

F Further Details for Full Inline Resolution

We discuss the depression of the overall distraction rate in the inline experiments, give further detailed plots that illustrate the prevalence of logic errors in these experiments. The latter plots also show the effect of placing the disqualifier last in these inline plots.

F.1 Lowered Rate of Overall Distraction.

As we mentioned in the main text, the overall rate of distraction in these inline experiments is significantly lower than that observed in Fig. 2. We present Fig. 19, which reports the distraction rates for both experiments together to allow for easier comparison.

Note that this suppression of the distraction rate could be arising because the inline generation significantly depresses logic errors (as discussed in §5.2, or because of systematic differences introduced due to the difference in the prompt structure.

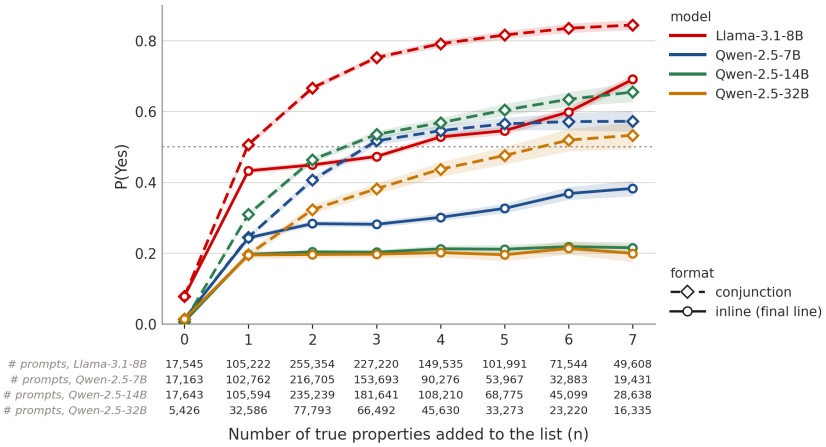
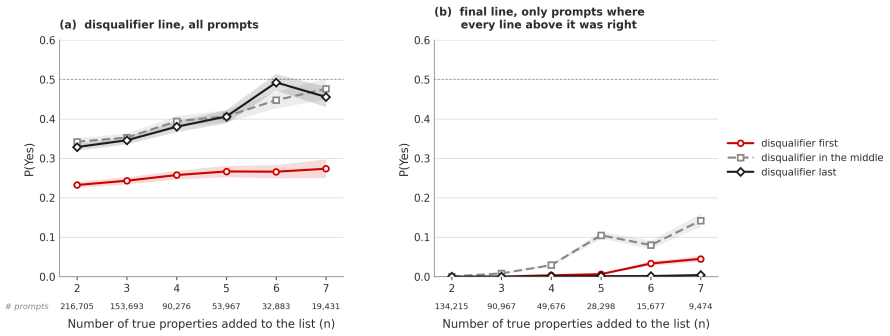


Figure 19: **Answering line by line helps, but does not fix it.** $P(\text{YES})$ on the final verdict, single-verdict format (dashed) against the final line of the inline format (solid), on the same prompts. Colours mark models; $n = 0$ is the disqualifier asked alone. The inline format lowers false acceptance in every model, most in Qwen2.5-14B and 32B, which stay near 0.2 at every n. The correct answer is NO throughout. Numbers below the axis are prompts per point.

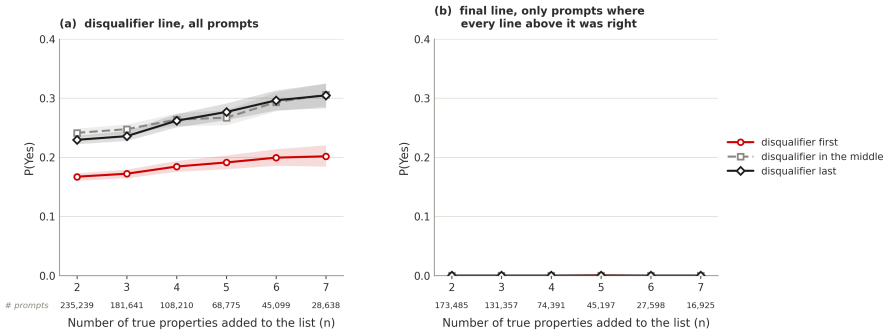
F.2 Further Presentations of Absence of Logic Errors, and the Effect of Disqualifier Placement.

Fig. 20 shows $P(\texttt{Yes})$ on the disqualifier line (left), and logic errors, measured as $P(\texttt{Yes})$ on the final line for prompts where every line above it was answered correctly (right). While both effects are evident in Fig. 7 in the main text, these plots enable clearer visualisation of these rates.

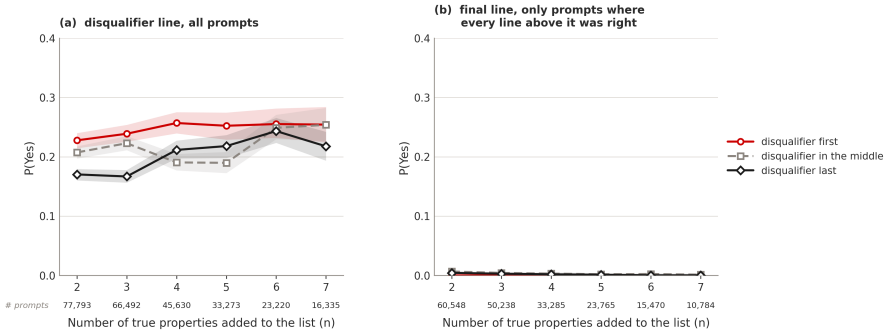
In addition, we include corresopnding lines for the same experiment, but wherein the disqualifier is placed *last*. Naturally, one expects that a disqualifier placed in this position is more 'brittle' in that its resolution can now be biased by the resolution of distractors earlier in the list. Interestingly, while this is true for Qwen2.5-7B,14B, and Llama3.1-8B, this expected ordering is switched for Qwen2.5-32B. Nevertheless, we note that logic errors remain similarly controlled even when disqualifier position is varied. In fact, placing the disqualifier last eliminates the logic errors that we saw in Llama3.1-8B when the disqualifier was placed first. This may be due to increased fact corruption (i.e., puzzles where we were earlier seeing logic errors now may simply have the disqualifier corrupted already).



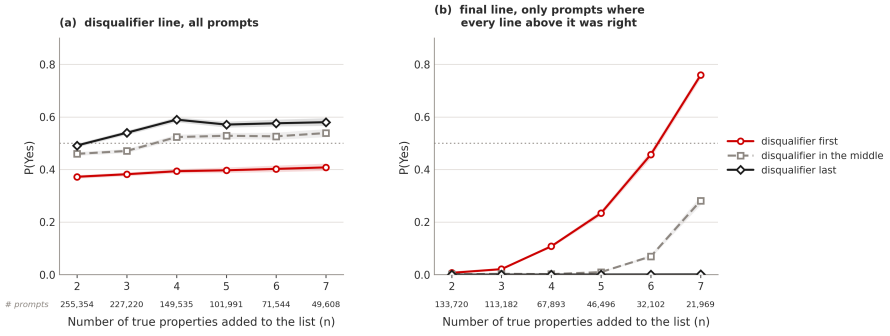
(a) Qwen2.5-7B



(b) Qwen2.5-14B



(c) Qwen2.5-32B



(d) Llama-3.1-8B

Figure 20: **Full inline resolution: disqualifier corruption, logic errors, and the effect of disqualifier placement.** Left: $P(\texttt{Yes})$ on the disqualifier line, all prompts. Right: $P(\texttt{Yes})$ on the final line, only for prompts where every line above it was right, so any `Yes` here is a logic error. For Qwen2.5-14B and 32B, logic errors are absent, and for Qwen2.5-7B they are rare even at high n. Llama-3.1-8B makes few logic errors at small n, but its final-line $P(\texttt{Yes})$ reaches about 0.75 at $n = 7$. Placing the disqualifier later usually raises disqualifier corruption (except for Qwen2.5-32B), but does not increase logic errors. For Llama, placing the disqualifier last *eliminates logic errors*.