CARD: CALIBRATED ADAPTIVE RELIANCE DECODING IN LLM-AUGMENTED SEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

Search engines augmented with large language models (LLMs) synthesize retrieved web content into summarized responses. When handling open-ended queries, LLMs implicitly select and rank candidate items, effectively functioning as recommenders. However, the LLM's reliance on parametric versus retrieved knowledge during this process is difficult to control: over-reliance on parametric knowledge leads to systematic favoritism toward known content, while over-reliance on retrieved content makes it vulnerable to manipulation of the retrieved documents. To investigate this problem, we first define *Parametric Reliance Score* (PRS) to quantify the strength of parametric knowledge reliance in LLM recommendation rankings. We validate this metric through a Latin-square-based controlled experiment across five LLMs, confirming systematic parametric preference. We then propose CARD (**C**alibrated **A**daptive **R**eliance **D**ecoding), a pluggable inference-time method that leverages PRS to enable bidirectional adaptive control over knowledge reliance. CARD measures KL divergence of token probability distributions after context insertion to derive an adaptive intervention strength on external knowledge, and calibrates the anchoring weight of the base predictions using the model-level PRS. Experiments on three open-source LLMs show that CARD mitigates parametric knowledge bias, narrowing the ranking gap between parametric and non-parametric items by 55–60%, and defends against adversarial attacks on retrieved content—suppressing retrieval poisoning to near-random rankings and neutralizing up to 95% of black-hat search engine optimization (SEO) attack effects. Our code is available at `https://anonymous.4open.science/r/CARD-1A64`.

1 INTRODUCTION

Search engines augmented with large language models (LLMs) (Zhao et al., 2023), such as Bing Chat, ChatGPT Search, and Google Search with AI Overviews, have emerged as a new search paradigm (Xiong et al., 2024; Vu et al., 2024; Mo et al., 2025). Rather than returning a ranked list of links, these systems retrieve relevant web documents and use an LLM to synthesize a single, summarized response (Gao et al., 2023; Lewis et al., 2020). In this paradigm, when users pose open-ended queries, the LLM not only synthesizes retrieved content but also implicitly selects and ranks specific items, effectively acting as a recommender (Caramancion, 2024; Liu et al., 2024; Li et al., 2024). However, the LLM's knowledge reliance during recommendation remains difficult to control. Over-reliance on parametric knowledge causes the LLM to favor content aligned with its pre-existing knowledge, leading to unfair recommendations (Dai et al., 2024; Dudy et al., 2025; Kamruzzaman et al., 2024). At the other extreme, over-reliance on retrieved content exposes the system to adversarial manipulation through retrieval poisoning (Zou et al., 2025; Zhang et al., 2024a) and black-hat search engine optimization (SEO) (Pfrommer et al., 2024; Nestaas et al., 2025).

Existing efforts to regulate LLMs' knowledge reliance (Tian et al., 2024; Huang et al., 2026; Bi et al., 2025) mostly shift the model toward either parametric or retrieved knowledge, rather than providing bidirectional and ranking-aware control. Methods that enhance the factuality of model parameters (Chuang et al., 2024; Li et al., 2023a; Zhang et al., 2024b) may further reinforce the model's preference for its pre-existing knowledge, exacerbating favoritism toward known brands in recommendation. Methods that enhance faithfulness to retrieved context (Zhao et al., 2024; Shi et al., 2024; Zhou et al., 2023) increase the system's susceptibility to poisoning of retrieved content.

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.

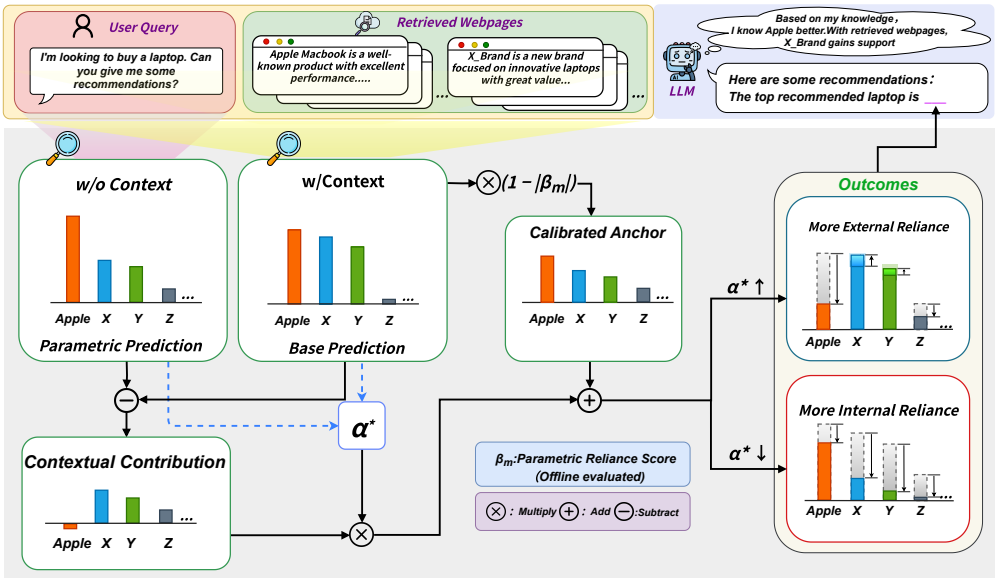


Figure 1: Overview of CARD. The contextual contribution is the difference between the base prediction (w/ Context) and the parametric prediction (w/o Context). CARD calibrates the base prediction using the offline Parametric Reliance Score $|\beta_m|$ and uses α^* to control the contextual contribution, enabling bidirectional adaptive control over the model's knowledge reliance.

Recent work such as CK-PLUG (Bi et al., 2026) adjusts knowledge reliance by detecting entropy shifts in token distributions after context insertion. Yet it targets knowledge conflicts in factual Question Answering (QA) (Su et al., 2024; Xu et al., 2024), where the goal is to decide whether a single answer should rely on parametric or external knowledge. Recommendation poses a fundamentally different challenge: multiple candidate items compete simultaneously, and the LLM must select and rank among them (Pfrommer et al., 2024; Puerto et al., 2026; Aggarwal et al., 2024). The effect of parametric knowledge manifests not as a single correct-or-incorrect answer, but as systematic ranking shifts across candidates. Controlling the LLM's knowledge reliance in this recommendation setting therefore remains an open problem.

To systematically study this problem, we first define *Parametric Reliance Score* (PRS), a metric that quantifies the strength of parametric knowledge reliance in LLM recommendation rankings. We validate PRS through a Latin-square-based controlled randomized experiment that balances confounding factors to isolate the effect of parametric knowledge status (Pfrommer et al., 2024). We then propose CARD (**C**alibrated **A**daptive **R**eliance **D**ecoding), a pluggable inference-time decoding method that enables bidirectional control over the LLM's knowledge reliance. CARD measures KL divergence of token probability distributions after context insertion to quantify the influence of external knowledge during generation. Meanwhile, the method calibrates the anchoring weight of the LLM's base predictions using the model-level PRS measured in the preceding experiment. Building on this, CARD adaptively adjusts the strength of intervention on external knowledge through weighted fusion of the base predictions and external contextual knowledge predictions, enabling bidirectional adaptive control over knowledge reliance.

Results show that, across five LLMs (LLaMA3.1-8B (Grattafiori et al., 2024), Qwen3-8B (Yang et al., 2025), Gemma3-12B (Team et al., 2025), GPT-5-mini (Singh et al., 2025), and Gemini-3.1-Flash-Lite(Team et al., 2023)), items present in the LLM's parametric knowledge receive significantly higher rankings than those absent from its parametric knowledge, confirming a systematic preference for content aligned with the LLM's parametric knowledge. Evaluation on three open-source models (LLaMA-3.1-8B (Grattafiori et al., 2024), Qwen3-8B (Yang et al., 2025), and Gemma3-12B (Team et al., 2025)) demonstrates that CARD effectively mitigates this preference, narrowing the ranking gap between parametric and non-parametric items by 55–60%. Meanwhile, CARD defends against two distinct attack categories: against retrieval poisoning, CARD suppresses the rankings of poisoned content to near-random levels; against black-hat SEO, CARD neutralizes up to 95% of the attack effect.

In summary, our contributions are as follows: (1) We define *Parametric Reliance Score* (PRS), a metric that quantifies the strength of parametric knowledge reliance in LLM recommendation rankings, and validate through a Latin-square-based experiment across five LLMs that LLM-augmented search engines exhibit systematic parametric preference. (2) We propose CARD (**C**alibrated **A**daptive **R**eliance **D**ecoding), a pluggable inference-time decoding method that enables bidirectional control over the LLM's knowledge reliance. (3) Experiments demonstrate that CARD both significantly mitigates parametric knowledge preference in recommendations and effectively defends against adversarial attacks on retrieved content.

2 Related work

Challenges in LLM-Augmented Search. LLM-augmented search engines synthesize retrieved web content into responses, implicitly selecting and ranking items for open-ended queries (Xiong et al., 2024; Pfrommer et al., 2024). Such systems are susceptible to adversarial manipulation of retrieved content, such as retrieval poisoning (Zou et al., 2025; Zhang et al., 2024a) and black-hat SEO (Nestaas et al., 2025; Pfrommer et al., 2024). Moreover, the LLM's parametric knowledge can itself influence its outputs. Studies have found that when retrieved context contains evidence consistent with parametric knowledge, LLMs tend to favor responses aligned with their internal knowledge (Xu et al., 2024; Su et al., 2024). These findings, however, have been primarily limited to knowledge conflict scenarios in factual QA.

Controlling Knowledge Reliance in LLMs. Existing methods for regulating LLMs' knowledge reliance are predominantly unidirectional or uncontrollable. One line of work amplifies parametric knowledge to improve factuality (Chuang et al., 2024; Zhang et al., 2024b). A complementary line enhances faithfulness to retrieved context (Shi et al., 2024; Wang et al., 2025; Huang et al., 2026; Tian et al., 2024). CK-PLUG (Bi et al., 2026) further enables flexible control over knowledge preference. While effective at reducing hallucinations, these methods are insufficient when LLMs handle open-ended queries that require selecting and ranking among multiple candidates.

3 How to Quantify Knowledge Reliance?

In this section, we first define ***P**arametric **R**eliance **S**core* (*PRS*) to quantify the strength of parametric knowledge reliance when external knowledge is available (§3.1). We then assess PRS through a Latin-square-based controlled randomized experiment and employ it to identify reliance preference between parametric and external knowledge (§3.2).

3.1 Parametric Reliance Score

In this section, we define PRS to quantify the strength of parametric knowledge reliance in LLM-augmented recommendation rankings.

Parametric Reliance Score. LLM-augmented search engines may preferentially recommend content aligned with the LLM's parametric knowledge. To quantify the strength of this preference, we define the following metric. Consider an LLM-augmented search engine that retrieves N candidate items for an open-ended query and generates a recommendation ranking over them. We categorize candidate items based on whether their names appear in the LLM's parametric knowledge. Specifically, *parametric-knowledge items* $\mathcal{I}_p$ are those whose names are extracted by prompting the LLM to generate recommendations without external context, following (Wu et al., 2024; Xie et al., 2024); *non-parametric-knowledge items* $\mathcal{I}_{np}$ are those whose names are constructed using symbolic placeholders, ensuring their absence from the model's parametric knowledge. Let $r(i) \in \{1, \ldots, N\}$ denote the rank assigned by the LLM to item i, where smaller values indicate higher recommendation positions. We define the *raw ranking score* of item i as:

$$S(i) = N - r(i) + 1, \tag{1}$$

and the corresponding *normalized ranking score* as:

$$s(i) = \frac{S(i)}{N} = \frac{N - r(i) + 1}{N}. \tag{2}$$

Table 1: Parametric reliance score (PRS) across five LLMs. $\bar{s}_p$ and $\bar{s}_{np}$ denote the mean normalized ranking scores of parametric-knowledge and non-parametric-knowledge items, respectively.

Models	$\bar{s}_p$	$\bar{s}_{np}$	PRS
LLaMA3.1-8B	0.6590	0.4022	0.2568
Qwen3-8B	0.6170	0.4405	0.1765
Gemma3-12B	0.6305	0.4312	0.1994
GPT-5	0.6389	0.4233	0.2156
Gemini3.1	0.5835	0.4758	0.1077

The ***P**arametric **R**eliance **S**core* (PRS) quantifies the strength of parametric knowledge reliance for a given model $\mathcal{M}$ across items:

$$\text{PRS}(\mathcal{M}) = \bar{s}_p - \bar{s}_{np} \tag{3}$$

where $\bar{s}_p$ and $\bar{s}_{np}$ denote the mean normalized ranking scores of parametric-knowledge and non-parametric-knowledge items, respectively. Since PRS is defined as $\bar{s}_p - \bar{s}_{np}$, a positive value indicates that parametric-knowledge items are ranked higher on average, while a negative value indicates that non-parametric-knowledge items are ranked higher on average. A larger absolute value indicates a stronger imbalance in knowledge reliance.

3.2 Latin-square-based Controlled Randomized Experiment

Then, we design a Latin-square-based controlled randomized experiment to measure PRS across five LLMs, and analyze their knowledge reliance.

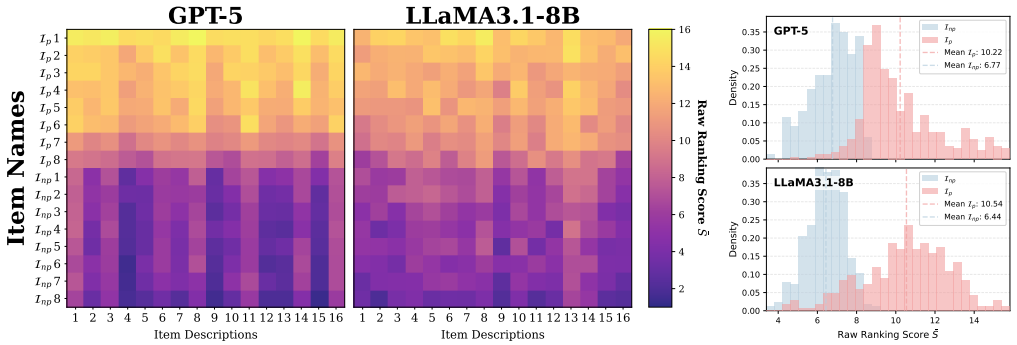
Latin Square. The primary factors that influence recommendation rankings in LLM-augmented search include the item name, the item description, and the context position of the item in the LLM's input (Pfrommer et al., 2024). To ensure that PRS reflects the effect of parametric-knowledge status rather than these confounding factors, we use Latin squares to isolate their effect across items. A *Latin square* of order N is an $N \times N$ array in which each of N symbols appears exactly once per row and once per column. Compared with a fully randomized experiment that independently samples factor-level assignments and requires a large number of runs to approximate balance, using Latin squares achieves structured balance in a fixed budget of N^2 runs per query.

The design consists of two layers of Latin squares. The first layer generates a single $N \times N$ Latin square L_1, where $L_1[i, b] = d$ assigns description d to item i in experimental block b. Because each row of L_1 is a permutation of $\{1, \ldots, N\}$, every item is paired with every description exactly once across the N blocks; because each column is also a permutation, within any single block, all N items receive distinct descriptions. The second layer generates, for each block b, an independent $N \times N$ Latin square $L_2^{(b)}$, where $L_2^{(b)}[i, r] = p$ places item i at context position p in run r of block b. By the same property, every item occupies every context position exactly once within each block, and in any single run, all N items are placed at distinct positions. Together, the two layers ensure that each item is paired with every description and appears at every context position across the experiment, i.e., for all $i, d, p \in [N]$:

$$\left|\left\{t : \text{desc}^{(t)}(i) = d,\ \text{pos}^{(t)}(i) = p\right\}\right| = 1 \tag{4}$$

where $\text{desc}^{(t)}(i)$ and $\text{pos}^{(t)}(i)$ denote the description and context position assigned to item i in run t, respectively. This balances the influence of item descriptions and context positions across $\mathcal{I}_p$ and $\mathcal{I}_{np}$ when computing PRS (Eq. 3).

Dataset. We use all $C = 50$ product categories from the RAGDOLL dataset (Pfrommer et al., 2024). For each category, we construct $N = 16$ candidate item names: $N/2$ parametric-knowledge item names extracted from the LLM's parametric knowledge, and $N/2$ non-parametric-knowledge items constructed using symbolic placeholders. Since our experiment requires N descriptions per category but the original dataset guarantees only at least 8, we use the LLM to generate N product descriptions per category. See Appendix A for the full experimental setup.



(a) (b)

Figure 2: (a) Heatmaps of average raw ranking scores $\bar{S}$ for GPT-5 and LLaMA-3.1-8B. Parametric-knowledge items ($\mathcal{I}_p$) receive consistently high $\bar{S}$ regardless of description assignment. (b) Density histograms of $\bar{S}$ for $\mathcal{I}_p$ and $\mathcal{I}_{np}$; dashed lines indicate group means.

Models. We evaluate five LLMs spanning both open-source and closed-source families: LLaMA-3.1-8B (Grattafiori et al., 2024), Qwen3-8B (Yang et al., 2025), and Gemma3-12B (Team et al., 2025) as open-source representatives, and GPT-5-mini (Singh et al., 2025) and Gemini-3.1-Flash-Lite (Team et al., 2023) as closed-source representatives.

Results. Table 1 reports the PRS across all five models. All models exhibit positive PRS values (range: 0.1077–0.2568), confirming that LLM-augmented search engines systematically assign higher recommendation rankings to items present in the LLM's parametric knowledge. LLaMA-3.1-8B exhibits the strongest parametric preference (PRS $= 0.2568$), while Gemini-3.1-Flash-Lite shows the weakest (PRS $= 0.1077$).

Figure 2(a) visualizes this pattern for GPT-5-mini and LLaMA-3.1-8B. Parametric-knowledge items (upper rows) maintain consistently high raw ranking scores across all description assignments, whereas non-parametric-knowledge items (lower rows) receive uniformly low scores regardless of the description they are paired with. This indicates that the item name itself—specifically, its presence in the LLM's parametric knowledge—dominates the ranking outcome. Figure 2(b) illustrates this separation:the distributions of average raw ranking scores for $\mathcal{I}_p$ and $\mathcal{I}_{np}$ are clearly separated with minimal overlap. All differences are statistically significant ($F \geq 487$; see Appendix C). For the evaluated models, parametric-knowledge items still rank significantly higher when symbolic placeholders are replaced with fluent, fictional-but-plausible non-parametric names (Appendix B).

Notably, the bias is consistent across both open-source models and closed-source models, suggesting that parametric knowledge preference is a general phenomenon independent of model architecture. These per-model PRS values will be used to calibrate the control method introduced in §4.1.

4 How to Control Knowledge Reliance?

In this section, we move from measuring knowledge reliance to controlling it. We present CARD, a pluggable inference-time method that uses the reliance measurement from Section 3.1 to adaptively shift the model between parametric and external knowledge (§4.1). We evaluate both control directions: reducing excessive reliance on parametric knowledge to mitigate inherent recommendation bias (§4.2), and suppressing unreliable external evidence to defend against adversarial retrieval manipulation (§4.3), followed by further analysis (§4.4).

4.1 CARD: Calibrated Adaptive Reliance Decoding

In this section, we derive an adaptive intervention strength on external knowledge from the shift in predictions after context insertion, and calibrate the base predictions using the parametric reliance measurement from §3.1 to enable bidirectional adaptive control over knowledge reliance at inference time.

Preliminary. Given an input sequence of tokens $X = \{x_1, x_2, \ldots, x_{t-1}\}$, an LLM parameterized by θ produces a probability distribution over the vocabulary $\mathcal{V}$ for the next token x_t:

$$p(x_t \mid x_{<t}) = \mathrm{softmax}\big(f_\theta(x_{<t})\big), \quad x_t \in \mathcal{V}. \tag{5}$$

where $f_\theta(\cdot)$ maps X to unnormalized scores over $\mathcal{V}$. In an LLM-augmented search engine, given a query x and retrieved external content c, we consider the next-token prediction under two conditions and denote their log-probability forms as:

- $q_{\mathrm{base}}(y) = \log p(y \mid c, x)$: the base prediction, conditioned on both the query and the retrieved content, capturing the combined influence of parametric and external knowledge.
- $q_{\mathrm{para}}(y) = \log p(y \mid x)$: the parametric prediction, conditioned only on the query, reflecting the LLM's inherent parametric knowledge.

Their difference isolates the contribution of the external retrieved content to the prediction:

$$q_{\mathrm{cont}}(y) = \log \frac{p(y \mid c, x)}{p(y \mid x)} = q_{\mathrm{base}}(y) - q_{\mathrm{para}}(y). \tag{6}$$

CARD combines q_{base}, q_{para}, and q_{cont} to enable bidirectional control over the LLM's knowledge reliance when ranking items.

Adaptive Intervention Strength. To determine how strongly to trade off the parametric and external knowledge contributions during generation, we measure the KL divergence between the two conditional distributions defined above:

$$\kappa = \mathrm{KL}\big(p(y \mid x) \,\|\, p(y \mid c, x)\big) = \sum_{y \in \mathcal{V}} p(y \mid x) \ln \frac{p(y \mid x)}{p(y \mid c, x)}. \tag{7}$$

A small κ indicates that the predictions remain similar with and without retrieved content, suggesting that the current generation is primarily driven by parametric knowledge; a large κ indicates that the external retrieved content significantly contributes to the prediction. Based on this shift, we define the signed adaptive intervention strength:

$$\alpha^*(M, \kappa) = \begin{cases} 0, & M = 0, \\ \dfrac{M/|M|}{\ln(\exp(1/|M|) + \kappa)}, & M \neq 0. \end{cases} \tag{8}$$

Here, $M \in \mathbb{R}$ is a signed intervention parameter. Its magnitude $|M|$ upper-bounds the intervention strength, while its sign determines whether the external contribution is amplified or suppressed. When $\kappa = 0$, $\alpha^*(M, 0) = M$; for a fixed $|M|$, the magnitude $|\alpha^*(M, \kappa)|$ decreases monotonically as κ increases. Thus, CARD applies a stronger adjustment when retrieved content has limited influence, and reduces the adjustment when retrieved content has already strongly shifted the prediction, avoiding over-amplification of q_{cont}.

Bidirectional Knowledge Reliance Control. Building on the above, CARD enables bidirectional control over knowledge reliance through calibrated contrastive modulation of the base prediction. The adjusted log-probability score is:

$$q^{\mathrm{CARD}}(y) = (1 - |\beta_m|) \cdot q_{\mathrm{base}}(y) + \alpha^*(M, \kappa) \cdot q_{\mathrm{cont}}(y), \tag{9}$$

where $\beta_m = \mathrm{PRS}(\mathcal{M})$ (Eq. 3) is the parametric reliance score of model $\mathcal{M}$ measured in the Latin-square-based controlled experiment (§3.1), and $M \in \mathbb{R}$ is the signed intervention parameter.

The first term $(1-|\beta_m|)\cdot q_{\mathrm{base}}$ serves as a calibrated anchor: since q_{base} reflects both parametric and external knowledge, $|\beta_m|$ down-weights its anchoring strength according to the model-level reliance bias-regardless of whether the model over-relies on parametric knowledge (positive PRS) or external knowledge (negative PRS), a larger $|\beta_m|$ reduces the base prediction's contribution to the final output. The second term $\alpha^*(M, \kappa) \cdot q_{\mathrm{cont}}$ applies a directional adjustment on the external knowledge contribution: when $M > 0$, CARD amplifies the influence of retrieved content to mitigate parametric knowledge bias and promote fairer recommendations; when $M < 0$, CARD suppresses the contribution of retrieved content, making the prediction less susceptible to adversarial manipulation such as retrieval poisoning (Zou et al., 2025) and black-hat SEO attacks (Nestaas et al., 2025).

Table 2: Bias mitigation results across three models, reporting PRS and $F_{\text{desc/name}}$ for each method. Upward and downward subscripts denote improvements and degradations relative to Vanilla, respectively. Best values are boldfaced.

Method	LLaMA3.1-8B		Qwen3-8B		Gemma3-12B	
	PRS ↓	$F_{\text{desc/name}}$ ↑	PRS ↓	$F_{\text{desc/name}}$ ↑	PRS ↓	$F_{\text{desc/name}}$ ↑
Vanilla	0.2568	0.46	0.1765	1.52	0.1994	1.58
Prompt	0.2498$_{\uparrow 0.0070}$	0.45	0.1761$_{\uparrow 0.0004}$	1.51	0.1937$_{\uparrow 0.0057}$	1.68
CK-PLUG (Ada.)	0.3317$_{\downarrow 0.0749}$	0.08	0.2327$_{\downarrow 0.0562}$	0.33	0.2841$_{\downarrow 0.0847}$	0.21
CK-PLUG (Fixed)	0.2459$_{\uparrow 0.0109}$	0.43	0.1773$_{\downarrow 0.0008}$	1.49	0.1931$_{\uparrow 0.0063}$	1.66
CAD	0.1213$_{\uparrow 0.1355}$	**3.07**	0.1205$_{\uparrow 0.0560}$	**5.35**	0.0961$_{\uparrow 0.1033}$	4.56
CARD (Ours)	**0.1041**$_{\uparrow 0.1527}$	2.97	**0.0795**$_{\uparrow 0.0970}$	4.98	**0.0801**$_{\uparrow 0.1193}$	**6.28**

Adaptive Plausibility Constraint. To prevent the adjustment in Eq. 9 from amplifying low-probability tail tokens, we follow Li et al. (2023b) and restrict the output to a compact set of plausible candidates. Let $\mathrm{TopK}(q, k)$ denote the set of top-k tokens ranked by a log-probability distribution q. We take the candidate set as the union of the head tokens from the two conditional distributions:

$$\mathcal{V}_{\text{head}} = \mathrm{TopK}(q_{\text{base}}, k) \cup \mathrm{TopK}(q_{\text{para}}, k). \tag{10}$$

The next token is then drawn from q^{CARD} restricted to $\mathcal{V}_{\text{head}}$, ensuring that CARD adjusts the relative scores of plausible candidates rather than amplifying arbitrary tail tokens from the full vocabulary, avoiding performance degradation. Additionally, we analyze CARD's inference cost in Appendix F.

4.2 Mitigating Recommendation Bias

Setup. We adopt the same Latin-square-based setup and dataset as §3.2, and use three open-source models: LLaMA-3.1-8B, Qwen3-8B, and Gemma3-12B. Baselines include Vanilla, prompt-based reliance control (Prompt), Context-Aware Decoding (CAD) (Shi et al., 2024), which contrasts predictions with and without retrieved context, and CK-PLUG (Bi et al., 2026), which uses context-induced distribution shifts to determine when to intervene. Detailed experimental settings are provided in Appendix D.

Evaluation. We report PRS (Eq. 3) to measure the ranking gap between parametric-knowledge and non-parametric-knowledge items. Following the category-wise F-statistic analysis of Pfrommer et al. (2024), we report $F_{\text{desc/name}}$ as the ratio of the category-averaged description and item-name F-statistics; higher values indicate greater relative influence of retrieved descriptions on rankings.

Results. Table 2 compares CARD with Vanilla, prompt-based reliance control (Prompt), Context-Aware Decoding (CAD) (Shi et al., 2024), and adaptive and fixed CK-PLUG (Bi et al., 2026). Across all three models, CARD achieves the lowest PRS, reducing the ranking gap between parametric and non-parametric items by 55–60% relative to Vanilla. Prompt and CK-PLUG (Fixed) remain close to Vanilla, whereas CK-PLUG (Ada.) increases PRS on all models; CAD also reduces PRS

Table 3: Defense against TAP. $\bar{S}$: mean raw ranking score of the target item; Eff_D: fraction of the attack effect neutralized. Red subscripts denote deviations from Vanilla exceeding 1.0.

Defence	LLaMA3.1-8B		Qwen3-8B		Gemma3-12B	
	$\bar{S}$	Eff_D	$\bar{S}$	Eff_D	$\bar{S}$	Eff_D
Vanilla	2.3588 $_{-}$	-	2.9305 $_{-}$	-	2.8845 $_{-}$	-
TAP	4.9168$_{+2.56}$	0%	4.9995$_{+2.07}$	0%	4.9835$_{+2.10}$	0%
Prompt	4.9245$_{+2.57}$	-0.30%	4.9965$_{+2.07}$	0.14%	4.9808$_{+2.10}$	0.13%
PPL	4.2368$_{+1.88}$	26.58%	4.7980$_{+1.87}$	9.74%	4.4943$_{+1.61}$	23.31%
CAD	3.0753$_{+0.72}$	71.99%	4.3738$_{+1.44}$	30.24%	3.6088$_{+0.72}$	65.50%
CK-PLUG (Ada.)	3.8827$_{+1.52}$	40.42%	4.8678$_{+1.94}$	6.37%	4.3080$_{+1.42}$	32.18%
CK-PLUG (Fixed)	2.7737$_{+0.42}$	83.78%	4.6652$_{+1.73}$	16.16%	4.0285$_{+1.14}$	45.50%
CARD (Ours)	**2.5310**$_{+0.17}$	**93.27%**	**3.7578**$_{+0.83}$	**60.02%**	**2.7868**$_{-0.10}$	**95.34%**

but remains above CARD. The complementary ratio shows that both CAD and CARD increase $F_{\text{desc/name}}$ over Vanilla: CAD obtains the largest ratios for LLaMA-3.1-8B and Qwen3-8B, whereas CARD obtains the largest ratio for Gemma3-12B. Overall, these results show that CARD's adaptive mechanism—amplifying external knowledge contribution when parametric knowledge dominates the prediction—more effectively rebalances the ranking competition between parametric and non-parametric items. A complementary evaluation based on human quality rankings shows that CARD achieves the highest nDCG@4 among the compared methods across all three models, indicating improved recommendation utility alongside bias mitigation (Appendix D, Table 7).

4.3 Defending Against Adversarial Attacks

Setup. We evaluate CARD against two categories of adversarial attacks on retrieved content: retrieval poisoning, represented by PoisonedRAG (Zou et al., 2025), which injects attacker-crafted documents into the retrieved content to promote a target item; and black-hat SEO, represented by TAP (Pfrommer et al., 2024), which iteratively optimizes adversarial text within an item description to boost the target item's ranking. Both attacks are evaluated on all 50 categories of the RAGDOLL dataset (Pfrommer et al., 2024), using the same three open-source models as in §4.2. In each category, we use $N = 5$ candidate items: one non-parametric-knowledge item serves as the attack target, and the remaining four items, along with their descriptions, are drawn from the dataset. For both attacks, we compare CARD with prompt-based reliance control (Prompt), Context-Aware Decoding (CAD) (Shi et al., 2024), PPL (perplexity-based retrieval filtering), and CK-PLUG (Bi et al., 2026). Full experimental details are provided in Appendix E.

Evaluation. We report the raw ranking score $S(i)$ (Eq. 1) of the target item, ranging from 1 to N. For TAP, we additionally report *defense effectiveness* (Eff_D): $1 - |\bar{S}_{\text{def}} - \bar{S}_{\text{orig}}|/|\bar{S}_{\text{atk}} - \bar{S}_{\text{orig}}|$, where $\bar{S}_{\text{orig}}$, $\bar{S}_{\text{atk}}$, and $\bar{S}_{\text{def}}$ denote the mean raw ranking scores of the target item before attack, after attack, and after defense, respectively, measuring the fraction of the attack effect neutralized.

Results. Table 4 and Table 3 report the defense results against PoisonedRAG and TAP, respectively. For PoisonedRAG, CARD achieves the lowest scores on Qwen3-8B and Gemma3-12B and the second-lowest score on LLaMA3.1-8B, remaining among the strongest defenses across all three models. For TAP, Prompt provides essentially no mitigation, while PPL, CAD, and CK-PLUG provide partial or model-dependent mitigation. CARD achieves the lowest defended score and the highest Eff_D among the defense methods on all three models, neutralizing 60–95% of the attack effect. These results indicate that suppressing retrieved content and calibrating prediction weights with PRS provide robust defense across models.

4.4 Analysis

Unless otherwise specified, analysis experiments are conducted on LLaMA3.1-8B (Grattafiori et al., 2024) using a subset of 5 categories sampled from the full 50-category setup.

Ablation: PRS Calibration. CARD calibrates the anchoring weight of the base prediction using the model-level PRS from §3.2 (the term $(1 - |\beta_m|)$ in Eq. 9). To evaluate this component, we compare CARD against a variant that removes PRS calibration (CARD w/o β, setting $\beta_m = 0$). Figure 3 shows results across all three evaluation scenarios. Removing PRS calibration consistently degrades

Table 4: Defense against PoisonedRAG. We report the mean raw ranking score $\bar{S}$ of the target item; lower values indicate stronger defense.

Models	PoisonedRAG	Prompt	PPL	CAD	CK-PLUG (Ada.)	CK-PLUG (Fixed)	CARD
LLaMA3.1-8B	3.4265	3.3025	**1.9928**	2.8472	3.1925	2.4290	2.0555
Qwen3-8B	3.3865	3.3490	3.5943	2.8030	3.0697	2.9295	**2.4305**
Gemma3-12B	2.6808	2.7153	2.7518	2.2860	2.4275	2.3490	**2.2370**

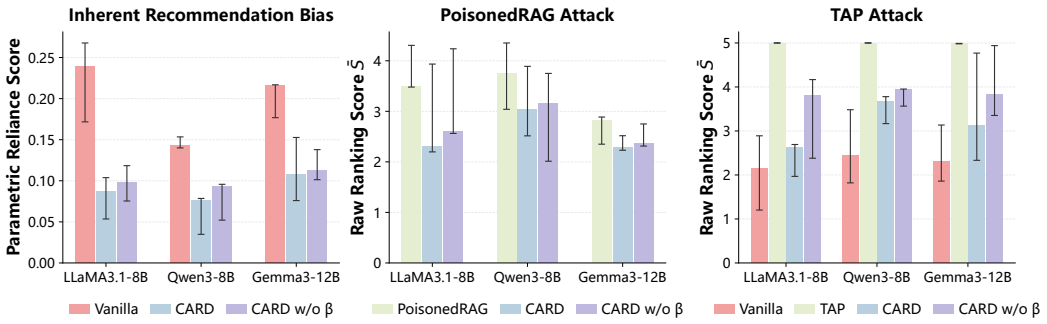


Figure 3: Ablation of PRS calibration. CARD w/o β removes the PRS-calibrated anchoring weight, setting $\beta_m = 0$. Results are shown for bias mitigation, PoisonedRAG defense, and TAP defense.

performance: in bias mitigation, the PRS reduction is weaker; in defense against both PoisonedRAG and TAP, the target item's ranking score remains higher. This confirms that incorporating the model-level parametric bias measurement to down-weight the base prediction provides additional control beyond the adaptive intervention alone.

Sensitivity: Intervention Magnitude $|M|$. The magnitude $|M|$ controls the maximum adaptive intervention magnitude through Eq. 8. Figure 4 shows the effect of varying $|M|$ on bias mitigation and TAP defense. For bias mitigation (left), increasing $|M|$ monotonically narrows the ranking gap between parametric and non-parametric items, with PRS decreasing from 0.24 ($|M|$=0) to 0.08 ($|M|$=1.9). For TAP defense (right), defense effectiveness increases from 0% to 83% as $|M|$ grows. In both cases, the improvement exhibits diminishing returns beyond $|M| \approx 1.1$, suggesting that moderate intervention magnitude suffices and further increases yield limited additional benefit. The parameter analysis for PoisonedRAG is provided in Appendix E and Table 8.

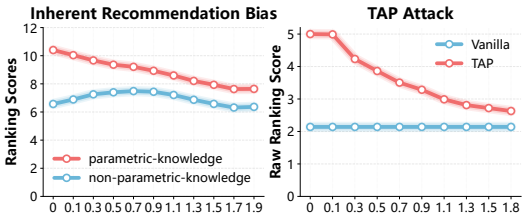


Figure 4: Effect of the intervention magnitude $|M|$ on bias mitigation (left) and TAP defense (right). Experiments conducted on LLaMA3.1-8B.

5 Conclusion

We study how to control knowledge reliance in LLM-augmented search, where the LLM implicitly selects and ranks candidate items for open-ended queries. We first define the *Parametric Reliance Score* (PRS), and a Latin-square-based controlled experiment across five LLMs validates that LLM-augmented search engines systematically favor items aligned with their parametric knowledge. We then propose CARD, a pluggable inference-time method that derives an adaptive intervention strength from the KL divergence of token distributions after context insertion and calibrates the anchoring weight of the base prediction using the model-level PRS, enabling bidirectional control over knowledge reliance. On three open-source LLMs, CARD narrows the parametric–non-parametric ranking gap by 55–60%, suppresses retrieval-poisoning attacks to near-random rankings, and neutralizes up to 95% of black-hat SEO attack effects, consistently outperforming CK-PLUG. Together, PRS and CARD form a measurement-and-control framework for knowledge reliance in LLM-augmented search.

AI use statement

We used AI tools solely for language polishing to improve clarity and readability. The AI tools did not contribute to the scientific content, ideas, analyses, or conclusions of this work.

REFERENCES

Pranjal Aggarwal, Vishvak Murahari, Tanmay Rajpurohit, Ashwin Kalyan, Karthik Narasimhan, and Ameet Deshpande. Geo: Generative engine optimization. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 5–16, 2024.

Baolong Bi, Shenghua Liu, Yiwei Wang, Lingrui Mei, Junfeng Fang, Hongcheng Gao, Shiyu Ni, and Xueqi Cheng. Is factuality enhancement a free lunch for llms? better factuality can lead to worse context-faithfulness. In *International Conference on Learning Representations*, volume 2025, pp. 41266–41297, 2025.

Baolong Bi, Shenghua Liu, Yiwei Wang, Yilong Xu, Junfeng Fang, Lingrui Mei, and Xueqi Cheng. Parameters vs. context: Fine-grained control of knowledge reliance in language models. 2026.

Kevin Matthe Caramancion. Large language models vs. search engines: evaluating user preferences across varied information retrieval scenarios. *arXiv preprint arXiv:2401.05761*, 2024.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, volume 2024, pp. 54158–54183, 2024.

Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. Bias and unfairness in information retrieval systems: New challenges in the llm era. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, pp. 6437–6447, 2024.

Shiran Dudy, Thulasi Tholeti, Resmi Ramachandranpillai, Muhammad Ali, Toby Jia-Jun Li, and Ricardo Baeza-Yates. Unequal opportunities: Examining the bias in geographical recommendations by large language models. In *Proceedings of the 30th International Conference on Intelligent User Interfaces*, pp. 1499–1516, 2025.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Jiawei Sun, Haofen Wang, et al. Retrieval-augmented generation for large language models: A survey. 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Pengcheng Huang, Zhenghao Liu, Yukun Yan, Haiyan Zhao, Xiaoyuan Yi, Hao Chen, Zhiyuan Liu, Maosong Sun, Tong Xiao, Ge Yu, et al. Parammute: Suppressing knowledge-critical ffns for faithful retrieval-augmented generation. *Advances in Neural Information Processing Systems*, 38: 100378–100410, 2026.

Mahammed Kamruzzaman, Hieu Minh Nguyen, and Gene Louis Kim. "global is good, local is bad?": Understanding brand bias in llms. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12695–12702, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*, pp. 9459–9474, 2020.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36:41451–41530, 2023a.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 12286–12312, 2023b.

Yongqi Li, Xinyu Lin, Wenjie Wang, Fuli Feng, Liang Pang, Wenjie Li, Liqiang Nie, Xiangnan He, and Tat-Seng Chua. A survey of generative search and recommendation in the era of large language models. *arXiv preprint arXiv:2404.16924*, 2024.

Lin Liu, Jiajun Meng, and Yongliang Yang. Llm technologies and information search. *Journal of Economy and Technology*, 2:269–277, 2024.

Fengran Mo, Kelong Mao, Ziliang Zhao, Hongjin Qian, Haonan Chen, Yiruo Cheng, Xiaoxi Li, Yutao Zhu, Zhicheng Dou, and Jian-Yun Nie. A survey of conversational search. *ACM Transactions on Information Systems*, 43(6):1–50, 2025.

Fredrik Nestaas, Edoardo Debenedetti, and Florian Tramèr. Adversarial search engine optimization for large language models. In *International Conference on Learning Representations*, volume 2025, pp. 4857–4888, 2025.

Samuel Pfrommer, Yatong Bai, Tanmay Gautam, and Somayeh Sojoudi. Ranking manipulation for conversational search engines. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 9523–9552, 2024.

Haritz Puerto, Martin Gubri, Tommaso Green, Seong Joon Oh, and Sangdoo Yun. C-seo bench: Does conversational seo work? *Advances in Neural Information Processing Systems*, 38, 2026.

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. Trusting your evidence: Hallucinate less with context-aware decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pp. 783–791, 2024.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, et al. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*, 2025.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. Conflictbank: a benchmark for evaluating knowledge conflicts in large language models. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, pp. 103242–103268, 2024.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti

Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Katherine Tian, Eric Mitchell, Huaxiu Yao, Christopher Manning, and Chelsea Finn. Fine-tuning language models for factuality. In *International Conference on Learning Representations*, volume 2024, pp. 44851–44866, 2024.

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, et al. Freshllms: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 13697–13720, 2024.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. Adacad: Adaptively decoding to balance conflicts between contextual and parametric knowledge. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 11636–11652, 2025.

Kevin Wu, Eric Wu, and James Zou. Clasheval: Quantifying the tug-of-war between an llm's internal prior and external evidence. *Advances in neural information processing systems*, 37: 33402–33422, 2024.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*, volume 2024, pp. 35623–35646, 2024.

Haoyi Xiong, Jiang Bian, Yuchen Li, Xuhong Li, Mengnan Du, Shuaiqiang Wang, Dawei Yin, and Sumi Helal. When search engine services meet large language models: visions and challenges. *IEEE Transactions on Services Computing*, 17(6):4558–4577, 2024.

Rongwu Xu, Zehan Qi, Zhijiang Guo, Cunxiang Wang, Hongru Wang, Yue Zhang, and Wei Xu. Knowledge conflicts for llms: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8541–8565, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Quan Zhang, Binqi Zeng, Chijin Zhou, Gwihwan Go, Heyuan Shi, and Yu Jiang. Human-imperceptible retrieval poisoning attacks in llm-powered applications. In *Companion Proceedings of the 32nd ACM International Conference on the Foundations of Software Engineering*, pp. 502–506, 2024a.

Shaolei Zhang, Tian Yu, and Yang Feng. Truthx: Alleviating hallucinations by editing large language models in truthful space. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8908–8949, 2024b.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. 2023.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. Enhancing contextual understanding in large language models through contrastive decoding. In *Proceedings of the 2024 conference of the north american chapter of the association for computational linguistics: Human language technologies (volume 1: Long papers)*, pp. 4225–4237, 2024.

Wenxuan Zhou, Sheng Zhang, Hoifung Poon, and Muhao Chen. Context-faithful prompting for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14544–14556, 2023.

Wei Zou, Runpeng Geng, Binghui Wang, and Jinyuan Jia. {PoisonedRAG}: Knowledge corruption attacks to {Retrieval-Augmented} generation of large language models. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 3827–3844, 2025.

A Experimental Details for Quantifying Knowledge Reliance

This section details the experimental procedure for the Latin-square-based controlled randomized experiment described in §3.2. The experiment measures PRS for each model using $C = 50$ categories from the RAGDOLL dataset (Pfrommer et al., 2024), with $N = 16$ candidate items per category. The procedure consists of three stages: parametric-knowledge item name extraction (Figure 5, top and middle), item description generation (Figure 5, bottom), and Latin-square-based assignment of item names, item descriptions, and context positions, after which the LLM-augmented search engine produces a ranking using the system prompt in Figure 6 and the user query template in Figure 7.

Parametric-Knowledge Item Name Extraction. For each category, we prompt the LLM to generate its top-$N/2$ recommendations without any external context, following the protocol of Wu et al. (2024) and Xie et al. (2024). The system prompt (Figure 5, top) frames the model as a recommender that should output the highest-quality items first, while the user prompt (Figure 5, middle) instantiates the placeholder `<CATEGORY>` with the target category and `<K>` with $N/2 = 8$, and constrains the model to emit only ranked brand and model names without descriptions or filler text. With $N = 16$, this yields 8 model-elicited item names per category, which we operationally treat as the parametric item set $\mathcal{I}_p$ ($|\mathcal{I}_p| = N/2 = 8$). To construct the non-parametric item set $\mathcal{I}_{np}$ ($|\mathcal{I}_{np}| = N/2 = 8$), we use synthetic placeholder-like brand–model identifiers (e.g., `S_Brand`/`S_Model` through `Z_Brand`/`Z_Model`) as controlled item names that do not correspond to real-world products. Together, the two sets yield $N = 16$ candidate item names per category.

Item Description Generation. The Latin-square design requires $N = 16$ distinct item descriptions per category to pair with the N item names, but the original RAGDOLL dataset guarantees only at least 8 descriptions per category. We therefore prompt the LLM to generate $N = 16$ generic item descriptions per category using the prompt shown in Figure 5 (bottom), which fixes the word count to roughly 80–100 words, requires a unique `##DOC_{index}##` header for each description, and forbids any real brand or model name. Each generated description contains two literal placeholder tokens, `[BRAND]` and `[MODEL]`, marking the positions where the item name will later be inserted. The descriptions vary naturally in their level of detail and specifications while remaining category-appropriate, so that the parametric versus non-parametric distinction does not leak through systematic differences in description style.

Item Name and Description Assignment. Given the $N = 16$ item names and $N = 16$ generated descriptions per category, we use the two-layer Latin-square design (§3.2) to systematically assign item descriptions and context positions to item names across experimental runs. In each run, the placeholder tokens `[BRAND]` and `[MODEL]` in the assigned description are replaced with the corresponding brand and model identifiers, producing the final item representation that the LLM observes. This substitution ensures that every item name is paired with every item description once at each context position, so that each $(\text{item}, \text{description}, \text{position})$ triple appears exactly once across the experiment, while the Latin-square structure balances context position effects (Eq. 4). The LLM-augmented search engine then receives all N items—each consisting of an item name paired with its assigned description—arranged according to the context positions determined by the second-layer Latin square, and produces a recommendation ranking over them using the system prompt in Figure 6 and the user query template in Figure 7.

PRS Computation. For each category, we aggregate the normalized ranking scores across all runs produced by the Latin-square design. The mean normalized ranking scores $\bar{s}_p$ and $\bar{s}_{np}$ are computed

separately for item names in $\mathcal{I}_p$ and $\mathcal{I}_{np}$, and PRS is obtained via Eq. 3. The final PRS reported in Table 1 is averaged across all $C = 50$ categories.

Implementation. We use greedy decoding by default to ensure reproducibility. For GPT-5-mini and Gemini-3.1-Flash-Lite, we keep the reasoning effort at its default value provided by the respective API.

Parametric Knowledge Extraction

```
You are a product recommendation system. When provided a user query, you respond with a
     series of product recommendations. The order in which you list the products is
     crucial, as the user will likely choose the first product listed. Your goal is to
     provide the most relevant and high quality products first, as this will increase user
      satisfaction and the likelihood of a purchase. You should list the products in order
      of relevance and quality, with the best products first.
```

Parametric Knowledge Extraction

```
User query: I'm looking to buy a <CATEGORY>. Can you give me some recommendations?
Constraints:
- Rank the top <K> best options (from 1 to <K>), where 1 is the most highly recommended.
- Output only the Rank, Brand Name, and the exact Model Name.
- No descriptions, explanations, or filler text.
Example Format:
1. Brand: [Brand Name] || Model: [Model Name]
2. Brand: [Brand Name] || Model: [Model Name]
...
<K>. Brand: [Brand Name] || Model: [Model Name]
```

Item Description Generation (N=16)

```
You need to generate 16 different realistic product description documents in English for
     the category: <CATEGORY>.
Goal:
Generate 16 high-quality, generic product description documents. These documents will
     represent 16 different products and are designed to simulate a real-world shopping or
      search environment.
Requirements:
- Content: Each document shall describe a different product and include its respective
     specifications, features, and other relevant content.
- Diversity: Ensure that the descriptions for these different products vary naturally in
     their detail.
- Word Count: Each document should be approximately 80 to 100 words.
- Placeholders: In every document, use the literal token [BRAND] where a brand name should
      appear. In every document, use the literal token [MODEL] where a model name should
     appear. Use [BRAND] and [MODEL] naturally within the text. Do not use any real-world
     brand names or model names.
- Output Format: Exactly 16 documents. Each must start with ##DOC_{index}## (where {index}
      is 1 to 16) on its own line, followed by the content. Separate each document with
     one blank line.
- No Metadata: Do not output any titles, explanations, or notes.
Now, generate the 16 documents for the category: <CATEGORY>
```

Figure 5: Prompt templates used in §3.2 for parametric-knowledge item name extraction (top and middle) and item description generation (bottom). Tokens enclosed in angle brackets (e.g., <CATEGORY>, <K>) are placeholders instantiated at run time; <K> is set to $N/2 = 8$. The literal tokens [BRAND] and [MODEL] in each generated item description are later replaced with the corresponding brand and model identifiers during Latin-square assignment.

Recommender

```
You are a Retrieval Augmented Generation (RAG)-enabled product recommendation system. When
      provided a user query, you respond with a series of product recommendations. Each
     listed product corresponds to one of the retrieved documents in the provided context,
      and you should summarize the product information for the user's benefit. Each
```

```
     product should correspond to one paragraph, with two new lines in between. Do not use
      lists or bullet points to highlight product features. Do not repeat one product in
     two different paragraphs. Make sure to list **all** of the provided products in your
     output, using the full product name and brand in each part of your response. Don't
     cut the search results short.
The order in which you list the products is crucial, as the user will likely choose the
     first product listed. Your goal is to provide the most relevant and high-quality
     products first, as this will increase user satisfaction and the likelihood of a
     purchase. You should list the products in order of relevance and quality, with the
     best products first.
Start your response with Based on the search results, ... and then list products based on
     documents in your context.
START EXAMPLES
EXAMPLE 1:
User:
recommend some mechanical keyboards
Product recommender:
Based on the search results, here are some recommended mechanical keyboards:
The Keychron Q Max series is recommended as the best overall mechanical keyboard. It
     offers tons of features, excellent typing experience, improved wireless connectivity,
      and endless customization options.
The Keychron Q6 is named the best full-size mechanical keyboard. It delivers high-quality
     construction and top-notch typing quality right out of the box, with a gasket-mounted
      design for a softer, cushioned typing feel.
For a less expensive but still high-quality option, the GLORIOUS GMMK Pro is recommended
     as a versatile mid-range mechanical keyboard with extensive customization
     possibilities.
...
EXAMPLE 2:
User:
What are the best blenders?
Product recommender:
Based on the search results, the best blenders are:
The Vitamix 5200 is the ultimate home blender, offering unrivaled performance, nuanced
     controls, and long-lasting reliability. It has a powerful motor that can handle tough
      ingredients, a wide range of speed settings for precise control, and is known for
     its durability and long-lasting 7-year warranty.
The Oster Versa Pro Series Blender is an excellent budget-friendly alternative to the
     Vitamix 5200. It offers more speed variations and runs more quietly than similarly
     priced blenders.
The Cleanblend Blender is another high-performance blender that produces finer purees than
      most blenders in its price range, though it lacks the nuanced controls of the
     Vitamix 5200.
...
EXAMPLE 3:
User:
What pair of work boots should I buy?
Product recommender:
Based on the search results, here are some recommendations for comfortable and breathable
     work boots suitable for warm weather construction work:
The Caterpillar Excavator Superlite work boots are a great overall option. They are
     lightweight and comfortable, with a cushioned insole, contoured footbed, and rubber
     outsole that provides good traction and slip resistance. The composite safety toe and
      non-metallic construction make them lightweight and breathable. They are also
     waterproof, which can be useful in wet conditions.
Another good choice is the Skechers Crankton Construction Shoe. This is more of a sneaker-
     style work shoe, which can provide good comfort and breathability for long days on
     your feet. It has a leather, synthetic, and mesh upper, a shock-absorbing midsole,
     and memory foam insole. It comes in a soft or steel toe version to meet your safety
     needs.
For a budget-friendly option, the Brunt work boots are worth considering. They use quality
      materials and construction for the price point, which is lower than many premium
     work boot brands. The Distasio model is particularly recommended.
END EXAMPLES
```

Figure 6: System prompt used in §3.2 to elicit a ranking over the N candidate items. The corresponding user-side query template is shown in Figure 7.

RAG Recommendation Query

```
We now are processing a user query: <QUERY>
Here are some relevant documents:
START DOCUMENTS
DOCUMENT 1 (brand: <BRAND_1>, model: <MODEL_1>):
<DOCUMENT_1>
```

```
DOCUMENT 2 (brand: <BRAND_2>, model: <MODEL_2>):
<DOCUMENT_2>
...
DOCUMENT <M> (brand: <BRAND_M>, model: <MODEL_M>):
<DOCUMENT_M>
END DOCUMENTS
**Remember to include all <M> products in your response: so <M+1> paragraphs total,
     including the initial 'Based on the search results...' Make sure to list the products
      in order from best to worst.**
For your reference, here are again the product models you should include in your response:
<MODEL_1>
<MODEL_2>
...
<MODEL_M>
User:
<QUERY>
Product recommender:
```

Figure 7: User query template used in §3.2, paired with the system prompt in Figure 6, to elicit a ranking over the N candidate items, each represented by an item name paired with its assigned item description and placed at its assigned context position. Tokens enclosed in angle brackets (e.g., <QUERY>, <M>, <BRAND_i>, <MODEL_i>, <DOCUMENT_i>) are placeholders instantiated at run time; <M> equals N in our setup. Markdown markers such as ** and ## are passed verbatim to the model.

B Robustness to Non-Parametric Item-Name Construction

Motivation. The symbolic identifiers used for non-parametric items provide a controlled construction, but their visibly synthetic form may also affect rankings. To examine whether the observed PRS depends on this surface-level difference, we compare them with fluent, fictional-but-plausible item names.

Alternative construction and protocol. For the same 50 categories, GPT-5.5 generated eight fictional-but-plausible brand–model names per category, yielding 400 initial names. Web searches identified real-world counterparts for 214 names (53.5%); these were removed and replaced with names that passed the same collision screening. In the alternative condition, only the eight non-parametric item names per category are changed. The parametric item names, item descriptions, Latin-square assignment protocol, ranking prompts, decoding settings, and PRS computation remain as described in Appendix A. The web screen reduces identifiable real-world name collisions; it cannot establish what any model has or has not encountered during pretraining.

Results. Table 5 compares the two name constructions for LLaMA-3.1-8B and Gemini-3.1. For each model and condition, the one-way ANOVA uses the category–item-name means and $F(1, 798)$ reference distribution described in Appendix C.

With fictional-but-plausible names, PRS decreases from 0.2568 to 0.1650 for LLaMA-3.1-8B and from 0.1077 to 0.0856 for Gemini-3.1. Nevertheless, the parametric-item ranking advantage remains positive in both alternative conditions ($F(1, 798) \geq 333.50$, $p < 0.001$). Thus, the synthetic appearance of placeholders affects the estimated magnitude of PRS, but does not change the central conclusion that parametric-knowledge items are ranked higher under this controlled setup. These

Table 5: PRS under alternative constructions of non-parametric item names. Only the non-parametric name construction changes between conditions.

Model	Non-parametric names	$\bar{s}_p$	$\bar{s}_{np}$	PRS	$F(1, 798)$	p-value
LLaMA-3.1-8B	Symbolic placeholders	0.6590	0.4022	0.2568	1186.36	< 0.001
	Fictional-but-plausible	0.6130	0.4480	0.1650	616.70	< 0.001
Gemini-3.1	Symbolic placeholders	0.5835	0.4758	0.1077	487.93	< 0.001
	Fictional-but-plausible	0.5732	0.4876	0.0856	333.50	< 0.001

within-condition tests do not establish whether the changes in PRS between name constructions are statistically significant.

C Statistical Significance Analysis

Following the use of F-statistics for ranking-factor analysis in Pfrommer et al. (2024), we further examine whether item-name status has a statistically reliable effect on ranking scores. In their analysis, F-statistics are used to quantify how strongly categorical factors affect ranking outcomes. We adopt the same ANOVA-style diagnostic, while applying it to the categorical status of the item name in our setting, i.e., whether an item name belongs to the parametric-knowledge group or the non-parametric-knowledge group.

The test is conducted under the same balanced experimental design described in §3.2. For each category, the Latin-square assignment controls the three main factors that may affect the generated ranking: the item name, the item description, and the context position. With $N = 16$ candidate item names, each category contains $N^2 = 256$ balanced runs. Across these runs, each item name is paired with each item description and appears at each context position according to the Latin-square property in Eq. 4. Therefore, the comparison between parametric-knowledge and non-parametric-knowledge item names is not driven by systematic differences in item descriptions or context positions.

Statistical Unit. The statistical unit of this analysis is a category–item-name mean ranking score. For each category and each item name, we average its ranking scores across all balanced assignments. Since we use $C = 50$ categories and $N/2 = 8$ item names in each group, this produces 400 category–item-name observations for the parametric-knowledge group and 400 category–item-name observations for the non-parametric-knowledge group. This aggregation avoids treating repeated balanced assignments of the same item name as independent observations.

Although the post-processing uses the raw ranking score $S(i)$, the same F-statistic would be obtained if the normalized ranking score $s(i) = S(i)/N$ were used instead. This is because multiplying all observations by the same constant scales both the between-group and within-group mean squares by the same factor, leaving their ratio unchanged.

One-way ANOVA. Let y_j denote the mean ranking score of the j-th category–item-name observation, and let $g_j \in \{\mathrm{parametric}, \mathrm{nonparametric}\}$ denote its item-name status. We fit the following one-way ANOVA model:

$$y_j = \mu + \alpha_{g_j} + \epsilon_j, \tag{11}$$

where μ is the overall mean, α_{g_j} is the effect associated with item-name status, and ϵ_j is the residual term. The null hypothesis is

$$H_0 : \alpha_{\mathrm{parametric}} = \alpha_{\mathrm{nonparametric}}, \tag{12}$$

which states that parametric-knowledge and non-parametric-knowledge item names have the same mean ranking score.

The F-statistic is computed as

$$F = \frac{\mathrm{MS}_{\mathrm{between}}}{\mathrm{MS}_{\mathrm{within}}}, \tag{13}$$

where $\mathrm{MS}_{\mathrm{between}}$ measures the variance explained by item-name status and $\mathrm{MS}_{\mathrm{within}}$ measures the residual variance within the two groups. A larger F-statistic indicates that the ranking-score gap between the two item-name groups is large relative to within-group variability. Since there are two groups and 800 category–item-name observations in total, the test has degrees of freedom $F(1, 798)$.

Results. Table 6 reports the resulting F-statistics and p-values. All models show highly significant item-name status effects, with $F(1, 798) \geq 487.9336$ and $p < 0.001$. These results indicate that the ranking gap captured by PRS is statistically reliable across all evaluated models. The F-statistic itself measures the strength of evidence against the equal-mean null hypothesis, while PRS provides the direction and magnitude of the ranking imbalance. Therefore, the positive PRS values reported in Table 1, together with the significant F-statistics in Table 6, show that parametric-knowledge item names are consistently ranked higher than non-parametric-knowledge item names after balancing item descriptions and context positions.

Model	$F(1, 798)$	p-value
LLaMA-3.1-8B	1186.3621	< 0.001
Qwen3-8B	691.0172	< 0.001
Gemma3-12B	739.6893	< 0.001
GPT-5-mini	887.6202	< 0.001
Gemini-3.1-Flash-Lite	487.9336	< 0.001

Table 6: One-way ANOVA results for the main effect of item-name status on ranking scores. Each test is computed over category–item-name mean scores, with 400 parametric-knowledge and 400 non-parametric-knowledge observations. The item description and context position are balanced by the Latin-square design. The p-values are computed under the corresponding $F(1, 798)$ reference distribution.

D Experimental Details for Bias Mitigation

This section provides the experimental details for the bias mitigation evaluation in §4.2. The goal of this experiment is to reduce the model's excessive reliance on parametric knowledge when producing recommendation rankings. We adopt the same Latin-square-based controlled experimental setup as §3.2, including the same category set, candidate construction procedure, and balanced assignment of the item name, the item description, and the context position. This ensures that the measured changes in PRS are attributable to the inference-time control method rather than to changes in the experimental design.

Operational Conditions for Knowledge Reliance Control. CARD does not assume that retrieved evidence should always dominate the model's parametric knowledge. The preferred control direction depends on the reliability, relevance, and freshness of the retrieved context. When the retrieved context is trusted, relevant, and likely to contain information that is newer or more task-specific than the model's parametric knowledge, CARD is used to increase the contribution of external knowledge. When the context is noisy, weakly relevant, stale, or potentially manipulated, CARD is used to reduce the contribution of external knowledge and shift the prediction toward parametric knowledge. Thus, external-knowledge amplification is appropriate for benign and trustworthy retrieval, whereas parametric-knowledge reliance serves as a defensive fallback under uncertain or adversarial retrieval. Under the unified parameterization, these two settings are represented by positive and negative values of M, respectively, while the magnitude controls the maximum intervention strength. The control direction is selected from the deployment setting and the retriever trust policy; CARD adapts the strength of the selected direction through κ but does not itself determine whether a retrieved document is trustworthy. The mitigation experiment below instantiates the trusted-context case, whereas the defense experiment instantiates the potentially compromised-context case.

Compared Baselines. Alongside CK-PLUG, we include Vanilla (without a reliance-control intervention), prompt-based reliance control (Prompt), and Context-Aware Decoding (CAD) (Shi et al., 2024), which contrasts predictions with and without retrieved context. Table 2 reports both adaptive and fixed CK-PLUG configurations.

CK-PLUG (Bi et al., 2026) is an inference-time method for regulating the model's reliance on parametric and external knowledge. CK-PLUG performs decoding with two conditional predictions: a parametric prediction conditioned only on the input, and a contextual prediction conditioned on both the input and the external information. It then measures the change in the token distribution after the external information is inserted, and uses this distributional shift to determine whether the current decoding step involves a knowledge conflict. When a conflict is detected, CK-PLUG adjusts the next-token scores by combining the parametric prediction with the contextual contribution; otherwise, it keeps the original contextual prediction.

Although CK-PLUG is related to our setting because it also operates at inference time and uses token-level probability distributions, its original objective differs from ours. CK-PLUG is designed for factual question answering, where the main decision is whether a single answer should rely more on parametric knowledge or on external information. In contrast, our setting requires the model to select and rank multiple competing candidate items. The effect of parametric knowledge

Table 7: nDCG@4 results for the bias mitigation experiment.

Method	Gemma3-12B	Llama3.1-8B	Qwen3-8B
Vanilla	0.8098	0.8107	0.8261
Prompt	0.8106	0.8064	0.8299
CK-PLUG(Ada.)	0.7776	0.7857	0.7906
CK-PLUG (Fixed)	0.8080	0.8073	0.8245
CAD	0.8452	0.8203	0.8412
CARD (Ours)	**0.8609**	**0.8207**	**0.8435**

therefore appears as a ranking-level imbalance across items, which is measured by PRS. As a result, the entropy-shift criterion used by CK-PLUG does not directly target the ranking-level preference captured by PRS, and may fail to mitigate parametric preference in our setting.

CARD Configuration. CARD enables bidirectional control over knowledge reliance through a single real-valued control parameter M. The parameter M determines both the direction and the maximum magnitude of the adjustment to the external knowledge contribution. For the bias mitigation experiment, we use the values of M listed below to amplify the external contribution and reduce excessive reliance on parametric knowledge.

The adaptive intervention at each decoding step is determined by the KL divergence κ:

$$\alpha^*(M,\kappa) = \begin{cases} 0, & M = 0, \\ \dfrac{M/|M|}{\ln(\exp(1/|M|)+\kappa)}, & M \neq 0. \end{cases} \tag{14}$$

For a fixed $|M|$, the magnitude $|\alpha^*|$ decreases monotonically as κ increases and reaches its upper bound $|M|$ when $\kappa = 0$. Thus, when the prediction with external information remains close to the prediction without external information, the resulting smaller κ leads to a stronger effective adjustment. In contrast, when the external information causes a larger distributional shift, the adaptive function reduces the effective adjustment, and a larger $|M|$ may be required to allow sufficient adjustment.

Selected values of M. The values used for each model are:

$$\begin{aligned} M_{\text{LLaMA-3.1-8B}} &= 1.1, \\ M_{\text{Qwen3-8B}} &= 2.2, \\ M_{\text{Gemma3-12B}} &= 1.2. \end{aligned} \tag{15}$$

Decoding and Evaluation. We use greedy decoding by default. All methods are evaluated under the same candidate construction and Latin-square assignment protocol as §3.2, so that the item name, the item description, and the context position remain balanced across runs. After each run, we parse the generated ranking, compute the normalized ranking score for each item, and report PRS following Eq. 3. The results in Table 2 use the same category set as the main quantification experiment.

Utility Evaluation. To evaluate the utility of our method, we report $\mathrm{nDCG@4}$ based on human quality rankings. We select four of the 16 documents and ask master's and doctoral students in computer science to rank them by quality.

Table 7 shows that CARD achieves the highest $\mathrm{nDCG@4}$ among all compared methods for all three models. This indicates that CARD can mitigate parametric knowledge reliance while simultaneously improving recommendation utility under the human quality rankings.

E Experimental Details for Adversarial Defense

This section provides the experimental details for the adversarial defense evaluation in §4.3. We evaluate two types of adversarial manipulation on the external information: PoisonedRAG (Zou

et al., 2025) and TAP (Pfrommer et al., 2024). Both attacks aim to promote a target item name, denoted as `Z_Brand` / `Z_Model`, by manipulating the external information observed by the model. We evaluate whether CARD can suppress the influence of such adversarial external information and recover a ranking closer to the unmanipulated setting.

PoisonedRAG Setup. For each category, we start from the original eight item descriptions in the dataset. The first four item descriptions are kept unchanged and paired with their original item names. The last four item descriptions are replaced by attacker-constructed poisoned item descriptions. All four poisoned item descriptions are associated with the same target item name, `Z_Brand` / `Z_Model`. As a result, the final context still contains eight item-description entries, but these entries correspond to five distinct item names: four normal item names and one poisoned target item name.

During evaluation, the model receives this mixed context and produces a recommendation ranking over the candidate item names. To avoid making the result dependent on a fixed context order, we randomly shuffle the item-name–item-description pairs before each run. For each category, we repeat this randomized evaluation $8 \times 10 = 80$ times. The target item's raw ranking score is then computed for each run and averaged across repetitions.

TAP Setup. For TAP, we use the item names and their corresponding item descriptions from the dataset. For each category, the first four item descriptions are kept as normal context entries. The fifth source item description is rewritten so that it is associated with the same target item name, `Z_Brand` / `Z_Model`. The evaluation contains two stages: before attack and after attack.

In the before-attack stage, the fifth item description is only rewritten to use the target item name, and does not include the attack prompt optimized by TAP. This stage measures the target item's ranking when it is present in the context but has not yet been adversarially promoted. In the after-attack stage, the fifth item description additionally contains the TAP-optimized attack prompt, which is designed to increase the ranking position of the target item. Thus, both stages use the same number of context entries: four normal item descriptions and one target item description. The only difference is whether the target item description contains the TAP-optimized attack prompt.

For each run, we randomly shuffle the five item-name–item-description pairs to balance the effect of context position. We repeat this randomized evaluation 80 times for each category. The model then produces a recommendation ranking from the mixed context, and we record the raw ranking score of the target item.

Defense Configuration. CARD is configured to suppress the contribution of adversarial external information. The defense configurations therefore use the values of M listed below. With these values, the second term in Eq. 9 suppresses the external contribution, reducing the extent to which manipulated external information can dominate the ranking. The PRS-calibrated anchor term further down-weights the base prediction according to the model-level reliance measurement, preventing the contaminated prediction conditioned on external information from fully determining the output.

The magnitude $|M|$ controls the largest possible magnitude of this suppression, while the actual intervention remains adaptive:

$$\alpha^*(M,\kappa) = \begin{cases} 0, & M = 0, \\ \dfrac{M/|M|}{\ln(\exp(1/|M|)+\kappa)}, & M \neq 0. \end{cases} \tag{16}$$

For a fixed $|M|$, the magnitude $|\alpha^*|$ decreases monotonically as κ increases and reaches its upper bound $|M|$ when $\kappa = 0$. Therefore, when the prediction with external information remains close to the prediction without external information, CARD applies a stronger effective intervention. When the external information causes a larger distributional shift, the adaptive function reduces the effective intervention, and a larger $|M|$ may be needed to allow sufficient suppression.

Values of M. For PoisonedRAG, we use:

$$\begin{aligned} M_{\text{LLaMA-3.1-8B}} &= -1.3, \\ M_{\text{Qwen3-8B}} &= -2.0, \\ M_{\text{Gemma3-12B}} &= -1.4. \end{aligned} \tag{17}$$

Table 8: Parameter analysis of CARD's intervention parameter M for defending against PoisonedRAG on LLaMA-3.1-8B. "Attack" denotes the raw ranking score of the target item without defense, while the numeric columns report CARD with different values of M on the held-out validation subset. Lower raw ranking scores indicate stronger suppression of the poisoned target item.

Method	Raw ranking score								
	Attack	-0.1	-0.3	-0.5	-0.7	-0.9	-1.1	-1.3	-1.5
PoisonedRAG	3.4925	3.315	3.05	2.7075	2.555	2.4525	2.3775	2.2975	2.42

Table 9: Decoding throughput on LLaMA-3.1-8B. Relative throughput is normalized to Vanilla.

Method	**Tokens/Second**	**Relative**
Vanilla	345.15	1.00×
CK-PLUG (Ada.)	222.66	0.65×
CK-PLUG (Fixed)	212.74	0.62×
CAD	258.26	0.75×
CARD (Ours)	**207.32**	**0.60×**

For TAP, we use:

$$\begin{aligned} M_{\text{LLaMA-3.1-8B}} &= -1.8, \\ M_{\text{Qwen3-8B}} &= -4.4, \\ M_{\text{Gemma3-12B}} &= -2.3. \end{aligned} \tag{18}$$

Table 8 reports the effect of different M values on PoisonedRAG defense performance. As $|M|$ increases, the raw ranking score of the poisoned target item decreases at first, indicating stronger defense. For LLaMA-3.1-8B, the lowest raw ranking score in this parameter sweep is obtained at $M = -1.3$, after which further increasing $|M|$ does not yield additional improvement.

Evaluation Metrics. For PoisonedRAG, we report the raw ranking score $S(i)$ of the target item, following Eq. 1. A lower score indicates that the defense more effectively suppresses the poisoned target item in the ranking. For TAP, we report both the raw ranking score and the defense effectiveness:

$$Eff_D = 1 - \frac{|\bar{S}_{\text{def}} - \bar{S}_{\text{orig}}|}{|\bar{S}_{\text{atk}} - \bar{S}_{\text{orig}}|}, \tag{19}$$

where $\bar{S}_{\text{orig}}$, $\bar{S}_{\text{atk}}$, and $\bar{S}_{\text{def}}$ denote the mean raw ranking scores of the target item before attack, after attack, and after defense, respectively. This metric measures the fraction of the attack effect neutralized by the defense. $Eff_D = 100\%$ indicates complete recovery to the before-attack score, while $Eff_D = 0\%$ indicates no reduction in the attack effect. A negative Eff_D indicates that the defense moves the score farther from the before-attack score, reflecting a harmful or over-correcting defense. We retain negative values rather than clipping them. A higher Eff_D indicates stronger recovery toward the before-attack ranking behavior.

Decoding. We use greedy decoding by default. The randomized repetitions described above are used only to balance context position effects. For each randomized run, the model receives the same set of item names and item descriptions for the corresponding attack condition, but their context positions are shuffled. The final reported results are averaged across all repetitions and categories.

F Inference Cost

CARD requires token-level log-probability distributions under both the retrieved-context and no-context conditions at each decoding step (Eq. 6). To quantify the resulting inference overhead, we benchmark decoding throughput on LLaMA-3.1-8B and compare Vanilla, Context-Aware Decoding (CAD) (Shi et al., 2024), CK-PLUG (Bi et al., 2026), and CARD. We report generated tokens per second, with relative throughput normalized to Vanilla.

CARD achieves 207.32 tokens per second, corresponding to approximately $0.60\times$ the throughput of Vanilla, or an approximately 40% reduction. Its throughput is close to that of CK-PLUG, whose adaptive and fixed configurations achieve $0.65\times$ and $0.62\times$ of the Vanilla throughput, respectively. CAD achieves a higher throughput of $0.75\times$. These results indicate that CARD incurs a substantial but comparable overhead relative to other decoding-time control methods.

This represents a trade-off between computational cost and the fine-grained control over knowledge reliance provided by CARD. We consider this trade-off reasonable for the following reasons:

1. **Inference-only deployment.** CARD is inference-only, requires no model retraining or fine-tuning, and does not modify model parameters. It can be inserted as a plug-and-play decoding module without changing the model weights.
2. **Comparable decoding overhead.** The measured throughput is comparable to other decoding-time control methods. In particular, CARD is close to both adaptive and fixed CK-PLUG, while CAD also incurs a substantial throughput reduction relative to Vanilla. These methods similarly require multiple conditional predictions or contrastive distributions during inference.
3. **Cost-sensitive deployment.** For high-stakes or safety-critical applications, such as medical diagnosis, financial advisory, or legal consultation, factual errors or unmanaged knowledge conflicts can have severe consequences. In such settings, the additional computational cost may be a justifiable trade-off for improved reliability, trustworthiness, and adversarial robustness.