💳 CARD: Calibrated Adaptive Reliance Decoding in LLM-Augmented Search

Anonymous ACL submission

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 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. 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

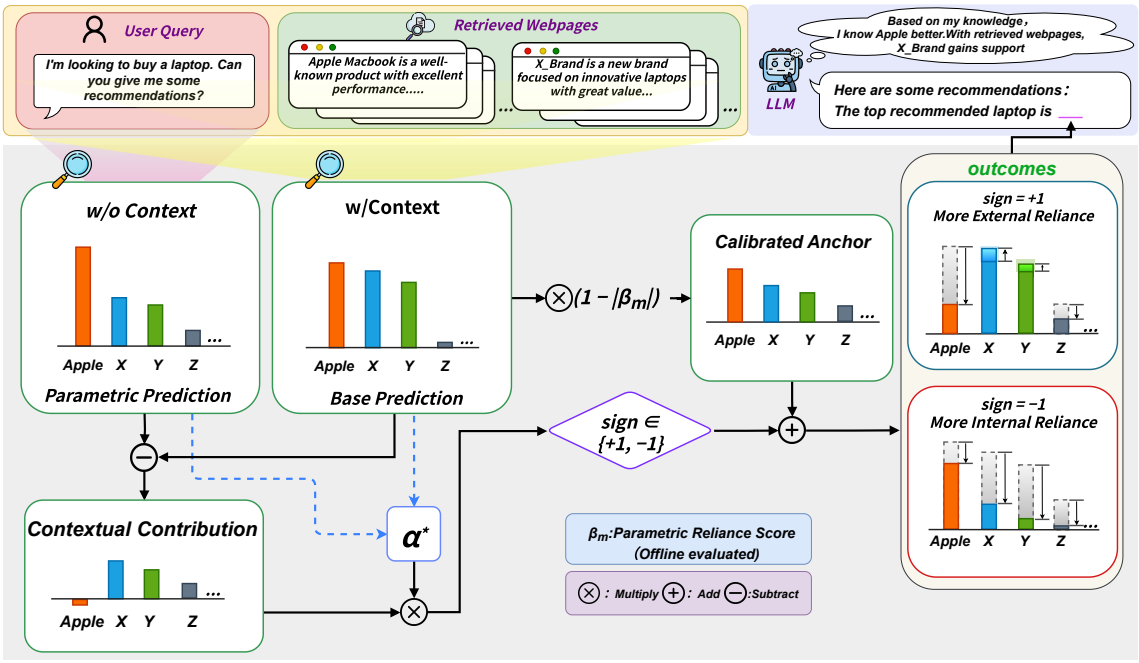


Figure 1: Overview of CARD. The contextual contribution is the difference between the base prediction (w/ Context) and the parametric prediction (w/o Context). CARD weights it by an adaptive strength α^* from the KL shift between the two, anchors the base prediction by the offline Parametric Reliance Score $|\beta_m|$, and uses sign to amplify $(+1)$ or suppress (-1) external knowledge.

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; Li et al., 2023a; 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:

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

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

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

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

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

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	Vanilla	CK-PLUG	CARD (Ours)
LLaMA3.1-8B	0.2568	$0.3317_{\downarrow 0.07}$	$0.1041_{\uparrow 0.15}$
Qwen3-8B	0.1765	$0.2327_{\downarrow 0.06}$	$0.0795_{\uparrow 0.10}$
Gemma3-12B	0.1994	$0.2841_{\downarrow 0.08}$	$0.0801_{\uparrow 0.12}$

Table 2: PRS after applying bias mitigation methods. $\uparrow$ and $\downarrow$ denote improvement and degradation relative to Vanilla, respectively.

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 $i, d, p \in [N]$:

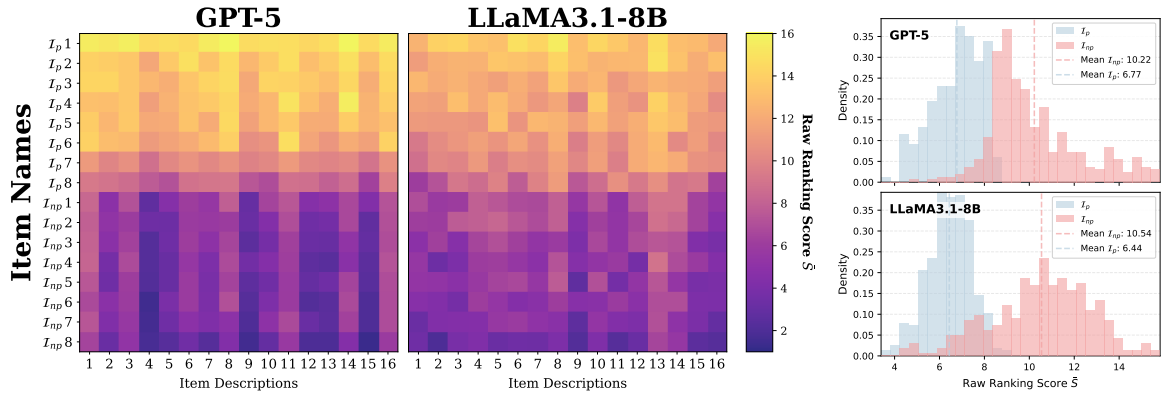
$$\left|\left\{t : \mathrm{desc}^{(t)}(i) = d,\ \mathrm{pos}^{(t)}(i) = p\right\}\right| = 1 \qquad (4)$$

where $\mathrm{desc}^{(t)}(i)$ and $\mathrm{pos}^{(t)}(i)$ 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.

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 prefer-



(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.

ence (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) further 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 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:

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

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

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:

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

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 adaptive intervention strength:

$$\alpha^* = \frac{1}{\ln\big(\exp(1/M) + \kappa\big)}, \tag{8}$$

where $M > 0$ upper-bounds the intervention strength: when $\kappa = 0$, $\alpha^* = M$, and as κ increases, α^* decreases monotonically. 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:

$$\begin{aligned}q^{\text{CARD}}(y) = (1 - |\beta_m|) \cdot q_{\text{base}}(y) \\ + \text{sign} \cdot \alpha^* \cdot q_{\text{cont}}(y),\end{aligned} \tag{9}$$

where $\beta_m = \text{PRS}(\mathcal{M})$ (Eq. 3) is the parametric reliance score of model $\mathcal{M}$ measured in the Latin-square-based controlled experiment (§3.1), and $\text{sign} \in \{+1, -1\}$ is a direction signal.

The first term $(1 - |\beta_m|) \cdot q_{\text{base}}$ 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 $\text{sign} \cdot \alpha^* \cdot q_{\text{cont}}$ applies a directional adjustment on the external knowledge contribution: when $\text{sign} = +1$, CARD amplifies the influence of retrieved content to mitigate parametric knowledge bias and promote fairer recommendations; when $\text{sign} = -1$, 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).

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 $\text{TopK}(q, k)$ 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:

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

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.

4.2 Mitigating Recommendation Bias

Setup. We adopt the same Latin-square-based experimental setup and dataset as §3.2, and use three open-source models: LLaMA-3.1-8B, Qwen3-8B, and Gemma3-12B. We apply CARD with $\text{sign} = +1$ to amplify the influence of retrieved content. Detailed experimental settings are provided in Appendix C.

Results. Table 2 compares CARD against CK-PLUG (Bi et al., 2026), which detects entropy shifts in token distributions after context insertion to adjust knowledge reliance at inference time. CARD substantially reduces PRS across all three models, narrowing the ranking gap between parametric and non-parametric items by 55–60%. In contrast, CK-PLUG increases PRS on all models (by 0.06–0.08), exacerbating parametric knowledge preference. This is because CK-PLUG targets binary knowledge conflicts in factual QA, where the goal is to decide whether a single answer should rely on parametric or external knowledge. In the recommendation setting, where multiple candidates compete simultaneously, its entropy-

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%
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%

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.

based detection misaligns with the ranking objective and inadvertently amplifies parametric preference. These results demonstrate that CARD's adaptive mechanism—amplifying external knowledge contribution when parametric knowledge dominates the prediction—effectively rebalances the ranking competition between parametric and non-parametric items.

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. Full experimental details are provided in Appendix D.

Evaluation. CARD is configured to suppress the influence of retrieved content and is compared against CK-PLUG (Bi et al., 2026). 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.

Defence	LLaMA3.1-8B	Qwen3-8B	Gemma3-12B
PoisonedRAG	3.4265	3.3865	2.6808
CK-PLUG (Ada.)	3.1925	3.0697	2.4275
CK-PLUG (Fixed)	2.4290	2.9295	2.3490
CARD (Ours)	2.0555	2.4305	2.2370

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

Results. Table 4 and Table 3 report the defense results against PoisonedRAG and TAP, respectively.

Against PoisonedRAG, CARD consistently achieves the lowest target ranking scores across all three models, suppressing the poisoned item to near-random levels. CK-PLUG provides partial mitigation but remains substantially less effective, particularly in its adaptive variant.

Against TAP, the attack nearly ranks the target item first on all models without defense. CARD neutralizes 60–95% of the attack effect, restoring the target item's ranking to near its pre-attack level. CK-PLUG exhibits inconsistent behavior: the fixed variant achieves 84% effectiveness on LLaMA3.1-8B but only 16% on Qwen3-8B. In contrast, CARD maintains effective defense across all three models. These results demonstrate that by suppressing the external knowledge contribution and down-weighting the contaminated base prediction through PRS calibration, CARD effectively isolates the model's ranking from adversarial manipulation in retrieved content.

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

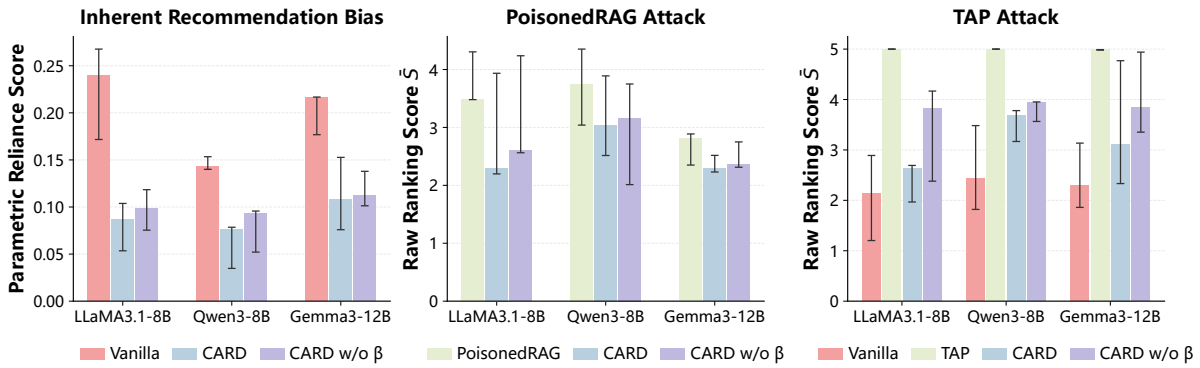


Figure 3: Ablation of PRS calibration. CuRe w/o β removes the PRS-calibrated anchoring weight (setting $\beta_m = 0$). Results are shown for bias mitigation (left), PoisonedRAG defense (middle), and TAP defense (right).

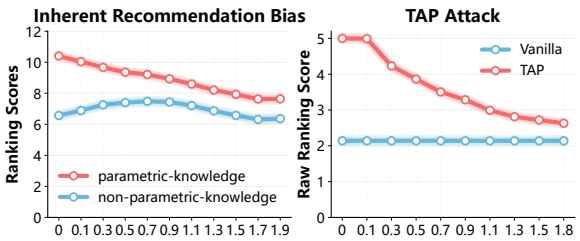


Figure 4: Effect of maximum intervention strength M on bias mitigation (left) and TAP defense (right). Experiments conducted on LLaMA3.1-8B.

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 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: Maximum Intervention Strength M. The hyperparameter M upper-bounds the adaptive intervention strength α^* (Eq. 8): larger M permits stronger adjustment of the external knowledge contribution. Figure 4 shows the effect of varying M in $[0.1, 1.9]$ 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 strength suffices and further increases yield limited additional benefit. The parameter analysis for PoisonedRAG is provided in Appendix D and Table 6.

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.

Limitations

Our study has several limitations. First, our non-parametric-knowledge items are constructed with symbolic placeholders. This construction is deliberate: it guarantees that the items are absent from the LLM's parametric knowledge. As a side

effect, the resulting names differ distributionally from real items that simply happen to be unknown to the model. Second, due to model capability and resource constraints, each category in our experiments contains only $N = 16$ candidate items (8 parametric-knowledge items and 8 non-parametric-knowledge items); we observed that further enlarging the candidate set causes the LLMs to produce malformed or incomplete rankings, which prevents stable measurement of PRS. Third, PRS captures parametric preference at the ranking level, while LLM-augmented search ultimately produces natural-language summaries; potential biases in surface realization (e.g., more elaborate or more positive descriptions for parametric-knowledge items) are not measured by our metric and warrant separate investigation.

References

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

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

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

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

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

Sunhao Dai, Chen Xu, Shicheng Xu, Liang Pang, Zhenhua Dong, and Jun Xu. 2024. 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*, pages 6437–6447.

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

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Jiawei Sun, Haofen Wang, and 1 others. 2023. Retrieval-augmented generation for large language models: A survey.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Pengcheng Huang, Zhenghao Liu, Yukun Yan, Haiyan Zhao, Xiaoyuan Yi, Hao Chen, Zhiyuan Liu, Maosong Sun, Tong Xiao, Ge Yu, and 1 others. 2026. Parammute: Suppressing knowledge-critical ffns for faithful retrieval-augmented generation. *Advances in Neural Information Processing Systems*, 38:100378–100410.

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

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *NeurIPS*, pages 9459–9474.

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

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori B Hashimoto, Luke Zettlemoyer, and Mike Lewis. 2023b. 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)*, pages 12286–12312.

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

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

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

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

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

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

Weijia Shi, Xiaochuang Han, Mike Lewis, Yulia Tsvetkov, Luke Zettlemoyer, and Wen-tau Yih. 2024. 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)*, pages 783–791.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

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

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

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

Tu Vu, Mohit Iyyer, Xuezhi Wang, Noah Constant, Jerry Wei, Jason Wei, Chris Tar, Yun-Hsuan Sung, Denny Zhou, Quoc Le, and 1 others. 2024. Freshllms: Refreshing large language models with search engine augmentation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13697–13720.

Han Wang, Archiki Prasad, Elias Stengel-Eskin, and Mohit Bansal. 2025. 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)*, pages 11636–11652.

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

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

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

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Shaolei Zhang, Tian Yu, and Yang Feng. 2024b. 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)*, pages 8908–8949.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, and 1 others. 2023. A survey of large language models.

Zheng Zhao, Emilio Monti, Jens Lehmann, and Haytham Assem. 2024. 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)*, pages 4225–4237.

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

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

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.

B 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 $g_j \in \{\text{parametric}, \text{nonparametric}\}$ 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

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

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

The F-statistic is computed as

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

where $\text{MS}_{\text{between}}$ measures the variance explained by item-name status and $\text{MS}_{\text{within}}$ 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 5 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 5, show that parametric-knowledge item names are consistently ranked higher than non-parametric-knowledge item names after balancing item descriptions and context positions.

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.

C 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.

Compared Method: CK-PLUG. We compare CARD with CK-PLUG (Bi et al., 2026), 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

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 5: 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.

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 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 the direction signal sign and the maximum intervention strength M. The direction signal determines whether CARD increases or decreases the contribution of external knowledge during decoding. When $\mathrm{sign} = +1$, CARD amplifies the external contribution, thereby reducing excessive reliance on parametric knowledge. When $\mathrm{sign} = -1$, CARD suppresses the external contribution, shifting the generation toward the model's parametric prediction. For the bias mitigation experiment, we set $\mathrm{sign} = +1$ for all models, because the objective is to reduce the ranking advantage of parametric-knowledge items.

The hyperparameter M controls the maximum intervention strength of CARD, while the actual intervention at each decoding step is adaptively determined by the KL divergence κ:

$$\alpha^* = \frac{1}{\ln\big(\exp(1/M) + \kappa\big)}. \tag{14}$$

For a fixed M, α^* 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 already leads to a larger effective intervention strength. In contrast, when the external information causes a larger distributional shift, the adaptive function reduces α^*, and a larger M may be required to allow sufficient adjustment.

Selection of M. We select M on a held-out validation subset consisting of one tenth of the full evaluation data. For each model, we choose the value that most effectively reduces PRS while preserving stable ranking generation. The selected values are:

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

These values are consistent with the role of M as an upper bound. LLaMA-3.1-8B has a larger PRS than Qwen3-8B, indicating stronger model-level reliance on parametric knowledge in the ranking results. Such reliance tends to yield smaller KL divergence after external information is inserted; under the same M, this would already produce a larger effective α^*. Therefore, LLaMA-3.1-8B can use a smaller maximum intervention strength, whereas Qwen3-8B requires a larger M to obtain sufficient effective adjustment.

Decoding and Evaluation. For both CARD and CK-PLUG, we use greedy decoding to ensure deterministic evaluation. 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 final results in Table 2 are averaged over the same category set used in the main quantification experiment.

D 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. Therefore, we set $\mathrm{sign} = -1$ for all adversarial defense experiments. With this direction, CARD subtracts the external contribution term in Eq. 9, 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 maximum intervention strength M controls the largest possible magnitude of this suppression, while the actual intervention strength remains adaptive:

$$\alpha^* = \frac{1}{\ln\big(\exp(1/M) + \kappa\big)}. \tag{16}$$

For a fixed M, α^* 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.

Selection of M. We select M on a held-out validation subset consisting of one tenth of the full evaluation data. For each model and each attack type, we choose the value that most effectively reduces the target item's ranking score while preserving stable ranking generation. Because PoisonedRAG and TAP manipulate the external information in different ways, we select M separately for the two attacks.

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 6: Parameter analysis of CARD's maximum intervention strength 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.

For PoisonedRAG, the selected values are:

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

For TAP, the selected values are:

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

These values are consistent with the role of M as an upper bound rather than a fixed intervention strength. The effective intervention is determined jointly by M and the model-specific KL divergence κ at each decoding step. A smaller κ already yields a larger effective α^* under the same M, so a model whose predictions remain closer before and after the insertion of external information may require a smaller maximum intervention strength. Conversely, when the manipulated external information causes a larger distributional shift, the adaptive function suppresses α^* more strongly, and a larger M may be required to provide enough room for defense. This also explains why the selected M values for TAP are generally larger than those for PoisonedRAG: TAP directly optimizes the target item description to promote the target item, which can create stronger distributional shifts and therefore requires a larger upper bound for effective suppression.

Table 6 provides the parameter analysis for PoisonedRAG on the validation subset. As M increases, the raw ranking score of the poisoned target item decreases at first, indicating stronger defense. The best validation result is obtained at $M = 1.3$, after which further increasing M does not yield additional improvement. This supports the selected value for LLaMA-3.1-8B in the PoisonedRAG defense experiment.

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. A higher Eff_D indicates stronger recovery toward the before-attack ranking behavior.

Decoding. All adversarial defense experiments use greedy decoding to ensure deterministic generation under a fixed context order. 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.

AI Writing Assistance Disclosure

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.

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.