

Why Thinking Hurts: Diagnosing and Rectifying Linguistic Inertia in Large Language Models for Recommendation

Luankang Zhang¹ Yonghao Huang¹ Hang Lv¹ Xuyang Zhi¹
Mingjia Yin¹ Yuyang Ye¹ Wei Guo² Hao Wang^{1*} Enhong Chen¹

¹University of Science and Technology of China

²Huawei Technologies

{zhanglk5,yhhuang213,lvhang1001,zxy_zds,mingjia-yin}@mail.ustc.edu.cn

yeyuyang@mail.ustc.edu.cn, guowei67@huawei.com,

{wanghao3,cheneh}@ustc.edu.cn

Abstract

Chain-of-Thought (CoT) reasoning is widely used to improve LLM performance, and recent foundation recommender models adopt it by generating textual reasoning before predicting target items represented by Semantic IDs (SIDs). However, we observe that enabling thinking mode in models such as OpenOneRec can degrade recommendation quality by up to 25%. We investigate this failure and identify *Linguistic Inertia*: when a textual CoT segment is inserted before SID generation, the model relies more on natural-language context and less on historical SID evidence. Further analyses show that this effect is amplified by reduced access to historical information and longer CoT lengths. To mitigate it, we propose *Linguistic-Inertia-Calibrated Decoding* (LICD), a training-free framework that combines *Reasoning-Chain Compression* and *Bias-Subtracted Contrastive Inference*. Experiments on three large-scale benchmarks show that LICD consistently outperforms both no-thinking and original-thinking baselines. Our code is available at <https://github.com/USTC-StarTeam/LICD>.

1 Introduction

Large language models (LLMs) have demonstrated strong reasoning abilities through Chain-of-Thought (CoT) prompting, where step-by-step reasoning improves their performance on a wide range of tasks (Wei et al., 2022; Guo et al., 2025). A natural expectation is that such reasoning should also benefit recommender systems. By explaining why a user may prefer certain items, the model may form a more coherent understanding of user

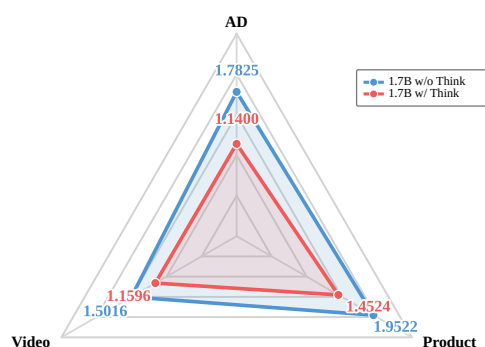


Figure 1: Impact of Thinking Mode on OpenOneRec.

intent before making a prediction. Recent foundation recommender models (Zhou et al., 2025a; Liu et al., 2025) have adopted this paradigm: a unified LLM first generates a textual reasoning chain and then predicts the target item represented by hierarchical Semantic IDs (SIDs).

However, we observe a counter-intuitive phenomenon: enabling thinking mode can sometimes reduce recommendation quality. As shown in Fig. 1, on three large-scale benchmarks, activating CoT reasoning in OpenOneRec leads to a performance drop of up to 25% compared with direct prediction without reasoning. This observation raises a question for both NLP and recommender systems: *Why can reasoning be harmful, and can we make it beneficial for SID prediction?*

We investigate this question through an empirical analysis and identify a phenomenon we call **Linguistic Inertia**. *It refers to the tendency of an LLM to rely more on natural-language context and less on historical SID evidence when a textual CoT segment is inserted before SID generation.* In foundation recommender models, this means that the model may underuse the rich information in

*Corresponding author.

historical interactions and instead depend more on the reasoning chain. Since the CoT is not always correct or sufficiently informative, amplified Linguistic Inertia can negatively affect the final recommendation. Our analysis shows that two factors strengthen this effect: reduced historical information and increased CoT length.

Motivated by these findings, we propose *Linguistic-Inertia-Calibrated Decoding* (LICD), a training-free inference framework that controls the two factors above to prevent textual information from overwhelming the SID history. LICD consists of two components. First, *Reasoning-Chain Compression* distills the verbose CoT into a single structured preference statement, reducing textual length while retaining intent-related signals. The compressed reasoning chain is then used to construct the *Expert* context. Second, *Bias-Subtracted Contrastive Inference* weakens historical information to construct an *Amateur* context, which exposes the text-dominated behavior caused by Linguistic Inertia. LICD then calibrates the final ranking by subtracting the normalized Amateur score from the Expert score. Experiments on three public datasets show that LICD consistently outperforms both the no-thinking and original-thinking baselines. Our main contributions are as follows:

- We identify *Linguistic Inertia*, a systematic phenomenon in thinking-enabled foundation recommender models, where textual reasoning makes SID prediction rely more on natural-language context and less on historical SID evidence.
- We identify two factors that amplify Linguistic Inertia: reduced historical information and increased CoT length.
- We propose LICD, an analysis-driven training-free framework that calibrates this reasoning shift through *Reasoning-Chain Compression* and *Bias-Subtracted Contrastive Inference*.
- Extensive experiments show that LICD consistently improves over both thinking and no-thinking baselines, mitigating the negative effect of Linguistic Inertia.

2 Empirical Analysis

In this section, we investigate how the thinking mode changes Semantic-ID prediction in foundation recommender models. Our analysis reveals a

systematic phenomenon we term *Linguistic Inertia*: when a textual CoT segment is inserted before SID generation, the model tends to rely more on the textual reasoning region and less on the historical SID sequence when making the final prediction. We further analyze the factors that amplify this effect and find that Linguistic Inertia becomes significantly stronger when the available historical information is reduced or when the textual CoT segment becomes longer.

2.1 Identifying Linguistic Inertia

We begin by examining how the textual CoT segment alters the model’s decision-making at the critical prediction position—the token where the model outputs the target Semantic ID.

Setup. Let x denote the input sequence containing both Semantic ID (SID) tokens from the user’s interaction history and textual instruction tokens. In thinking mode, the model generates a textual CoT c before predicting the target item y . We extract the attention distribution at the prediction position from the last Transformer layer, averaged over all heads, and compare Think=Off, where y is predicted from x alone, with Think=On, where y is predicted from the concatenated context (x, c) .

Observation 1: Attention shifts from SID history to textual context. We define the **SID Attention Ratio** as the fraction of total attention mass allocated to SID tokens at the prediction position. Formally, let $\mathbf{a} \in \mathbb{R}^L$ be the attention vector and $\mathcal{I}$ the set of SID token positions:

$$\text{SID Attention Ratio (SAR)} = \sum_{i \in \mathcal{I}} a_i / \sum_{j=1}^L a_j. \quad (1)$$

As shown in Fig. 2(a), enabling thinking mode reduces the SID Attention Ratio on both datasets (Product: -9.6% ; AD: -15.5%). This suggests that inserting a textual CoT segment reallocates attention away from historical SID tokens, leaving relatively more attention mass on non-SID textual context at the prediction position.

Observation 2: History+CoT logits become closer to text-only logits. The attention analysis shows reduced reliance on SID history; we next examine whether this change is reflected in the SID logit space. Let $\delta = \ell(y \mid x, c) - \ell(y \mid x)$ denote the logit change caused by inserting the textual CoT into the history context, and let $\mathbf{b} = \ell(y \mid$

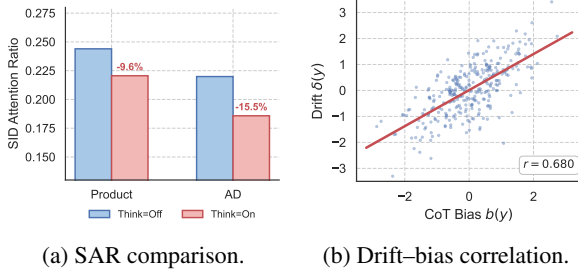


Figure 2: Evidence of Linguistic Inertia. (a) The SID Attention Ratio drops when thinking mode is enabled, indicating that attention shifts away from historical SID tokens. (b) The logit change caused by adding CoT is highly correlated with the logit pattern from the CoT-only context ($r = 0.680$), suggesting that the final decision is more influenced by textual information.

$c) - \ell(y | x)$ denote the contrast between a CoT-only textual context and the history-only context. If δ and $\mathbf{b}$ are highly correlated, then the model’s decision under history+CoT is closer to a text-only decision than to a primarily history-based one.

We define the drift-bias correlation as:

$$r = \text{PearsonCorr}(\delta, \mathbf{b}). \quad (2)$$

A larger r means that the effect of adding CoT is more consistent with the CoT-only contrast. As shown in Fig. 2(b), Product obtains a strong positive correlation ($r = 0.680$), indicating that the SID logits under history+CoT are substantially dominated by the textual CoT.

These two observations jointly reveal:

Linguistic Inertia

Once a textual CoT is fed into the LLM before SID generation, the LLM tends to rely more on information expressed in natural language and less on information encoded by historical SID tokens.

Although the textual CoT does not always contain wrong information, it may include hallucinated details, over-generalized descriptions, or coarse preference summaries. The risk is that, when producing the target Semantic ID, the LLM may over-rely on the textual CoT and underuse the SID sequence, allowing text-derived cues to dominate the recommendation even when the history contains more precise item-level evidence.

2.2 Probing Linguistic Inertia via Information Masking

We next examine which factors amplify Linguistic Inertia. At the prediction position, the LLM condi-

tions on both the historical SID sequence x and the textual CoT segment c . We therefore selectively mask one source at a time: masking history tests whether reduced SID information makes the LLM rely more on the textual CoT, while masking CoT tests whether this effect depends on the readable content of the CoT itself.

Approach. For each sample, we generate the textual CoT from the full history and then mask either the historical SID tokens or the CoT tokens with $\langle \text{endof text} \rangle$ at ratios from 0% to 90%. **Mask History** weakens the SID-history signal while keeping CoT unchanged, whereas **Mask CoT** weakens the textual CoT while keeping history unchanged. We keep the sequence length fixed and report SID Attention Ratio and drift-bias correlation r .

Results. Fig. 3 presents three findings.

Finding 1: Weakening history amplifies textual dominance. When history tokens are progressively masked (Fig. 3(a,d)), the SID Attention Ratio drops and r rises sharply (Product: $0.68 \rightarrow 0.96$). This indicates that as the historical SID sequence becomes less informative, the LLM relies less on history and makes decisions increasingly consistent with the CoT-only context.

Finding 2: Masking CoT content does not proportionally restore history reliance. When CoT tokens are masked (Fig. 3(b,e)), the SID Attention Ratio increases only marginally (Product: $+1.5\%$ at 90% mask), while r increases monotonically (Product: $0.68 \rightarrow 0.87$). Since larger r means stronger consistency with the CoT-only logits, this suggests that the LLM remains strongly influenced by the textual CoT region even when much of its readable content is masked. Thus, Linguistic Inertia is not solely determined by the specific semantic content of CoT tokens.

Finding 3: CoT length is a major contributor to attention dilution. We truncate the CoT to different lengths while keeping the history intact. As shown in Fig. 3(c), the SID Attention Ratio decreases as CoT length increases, and at 32 tokens it nearly recovers to the Think=Off level on Product (0.243 vs. 0.243). Since truncation changes both length and content, this does not rule out semantic effects; however, it shows that the amount of textual CoT is an important factor behind attention dilution. Fig. 3(f) suggests that shortening CoT restores SID attention more effectively than it removes logit-level text consistency.

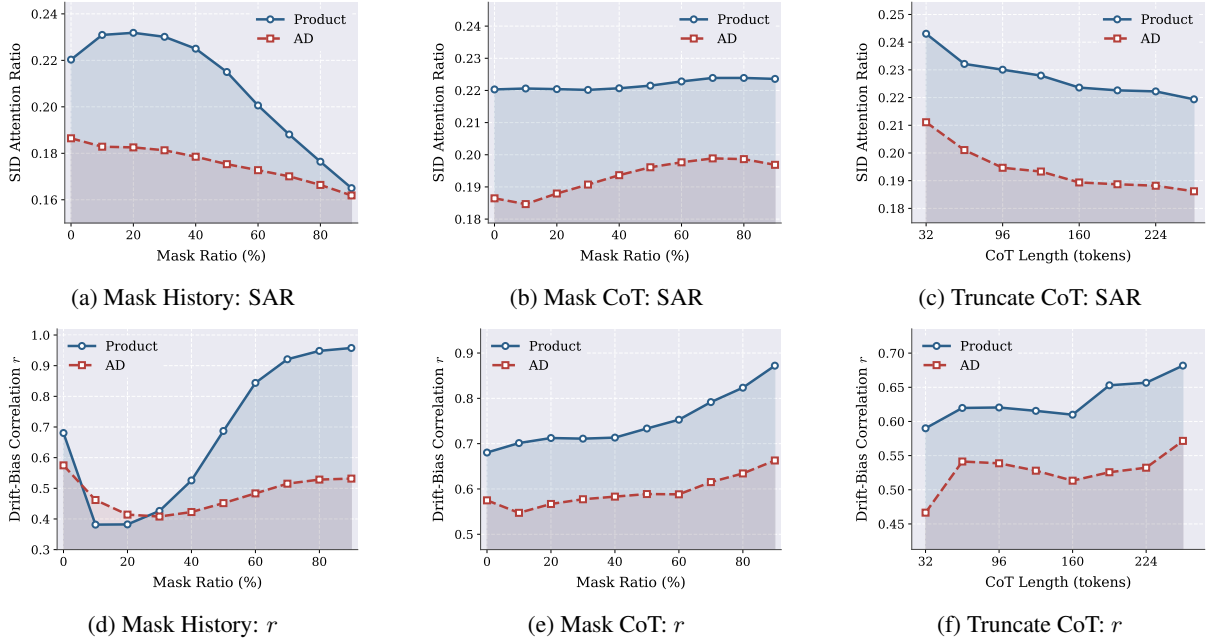


Figure 3: Probing Linguistic Inertia via information masking and length control. **(a,d)** Masking history SID tokens reduces SID attention and increases r , indicating that weakened history makes the decision more text-dominated. **(b,e)** Masking CoT tokens only mildly restores SID attention while r continues to increase, suggesting that Linguistic Inertia is not solely determined by the semantic content of CoT tokens. **(c,f)** Truncating CoT shows that textual length is a major contributor to attention dilution. Full details are provided in Appendix B.

Implications for method design. The analysis identifies two factors that amplify Linguistic Inertia: reduced history information and increased CoT length. Finding 3 suggests that the textual reasoning channel should be length-controlled before SID prediction, since long free-form CoT reduces attention to SID tokens. Finding 1 shows that when the historical SID sequence is weakened, the LLM relies more on the textual CoT, revealing how it behaves under text-dominant conditions.

Therefore, our method controls these two factors separately. We use a short, structured Expert context to limit the textual CoT length, and a weakened-history Amateur context to expose the text-dominant behavior for contrastive calibration. The goal is not to discard CoT, but to prevent textual information from overwhelming the SID history during final ranking.

3 Method: Linguistic-Inertia-Calibrated Decoding

Motivated by the empirical findings above, we propose *Linguistic-Inertia-Calibrated Decoding* (LICD), a training-free inference framework for SID prediction. The key idea is to preserve the useful preference information introduced by thinking while preventing the textual reasoning seg-

ment from overwhelming the historical SID evidence. To this end, LICD calibrates the final ranking through two complementary contexts: an *Expert* context that uses the full history with a compact preference control, and an *Amateur* context that weakens the history to expose the text-dominated behavior caused by Linguistic Inertia.

LICD consists of two components. First, *Reasoning-Chain Compression with Entropy Gating* converts the free-form CoT into a short, structured preference statement, and rejects uncertain compressed controls. This reduces attention dilution while retaining preference-relevant cues. Second, *Bias-Subtracted Contrastive Inference* constructs a weakened-history Amateur context to expose Linguistic Inertia, and subtracts its normalized score from the Expert score during inference.

3.1 Reasoning-Chain Compression

Section 2.2 shows that longer textual CoT reduces the SID Attention Ratio even when the historical SID sequence is intact. This indicates that verbose reasoning can dilute SID-grounded evidence during prediction. However, the CoT may still contain useful high-level preference cues. We therefore compress the free-form reasoning chain into a short and structured preference control before fi-

nal SID prediction.

Compression operator. Given the interaction history x , the model first generates a free-form reasoning chain c under thinking mode. We then apply a compression operator:

$$\hat{c} = \mathcal{T}(c), \quad \hat{c} \in \mathcal{C}_{\text{pref}}, \quad (3)$$

where $\mathcal{C}_{\text{pref}}$ is a restricted space of concise preference statements. The compressed control $\hat{c}$ is constrained by a fixed template and a length budget:

$$|\hat{c}| \leq L_c. \quad (4)$$

Instead of preserving the reasoning trajectory, $\hat{c}$ keeps only the preference-relevant information needed for SID prediction, reducing the verbose linguistic artifacts that amplify Linguistic Inertia.

Instantiation. We instantiate $\mathcal{T}$ with an off-the-shelf instruction-following model. The compressor removes intermediate reasoning and keeps preference-relevant information from the CoT. The full prompt is shown in Fig. 4. It outputs one sentence in the format: *The current user’s preference is [summary content]*. The sentence is used as the compact control $\hat{c}$ in the Expert context.

3.2 Bias-Subtracted Contrastive Inference

Reasoning-chain compression reduces the length of the textual context, but it does not explicitly remove the text-dominated bias identified in Section 2.2. The masking analysis shows that weakening historical SID tokens makes the prediction increasingly aligned with the CoT-only decision pattern. We therefore construct an *Amateur* context with degraded history to expose Linguistic Inertia, and subtract its signal from an *Expert* context that uses the full history and compressed reasoning.

Expert context. The Expert context conditions on the full interaction history x and the compressed preference control $\hat{c}$:

$$z_E(y) = \log P_\theta(y | x, \hat{c}), \quad y \in \mathcal{Y}. \quad (5)$$

This context preserves SID-grounded evidence while retaining the compact preference cues extracted from the reasoning chain.

Amateur context. To capture the text-dominated bias, we weaken the history by masking a fraction ρ of SID tokens:

$$x_\rho = \text{Mask}_\rho(x). \quad (6)$$

System Message

You are a user profiling expert. Please read the analysis process below and directly summarize the user’s core preferences.

Requirements:

1. Ignore filler words such as “I need to analyze”, “First”, etc.
2. Provide only one conclusion sentence.
3. Format: The current user’s preference is [summary content].

Input Example: ... Output Example: ...

User Message

{Original_CoT}

Figure 4: Prompt structure for reasoning-chain compression. The free-form CoT is compressed into a single structured preference statement.

We then regenerate a free-form reasoning chain from the degraded history,

$$c_A \sim P_\theta(\cdot | x_\rho), \quad (7)$$

and compute the Amateur score:

$$z_A(y) = \log P_\theta(y | x_\rho, c_A). \quad (8)$$

Since the SID evidence in x_ρ is weakened, high-probability candidates under this context are more likely to reflect Linguistic Inertia rather than precise history-grounded preference.

Score normalization. Because the two contexts may have different score scales, we normalize scores within the candidate set $\mathcal{Y}$:

$$\tilde{z}_k(y) = \frac{z_k(y) - \mu_k}{\sigma_k + \epsilon}, \quad k \in \{E, A\}, \quad (9)$$

where μ_k and σ_k are the mean and standard deviation of $\{z_k(y)\}_{y \in \mathcal{Y}}$, and ϵ is a small constant.

Bias-subtracted scoring. The ranking score is:

$$S(y) = \tilde{z}_E(y) - \alpha \cdot \tilde{z}_A(y), \quad (10)$$

where $\alpha \geq 0$ controls the strength of bias subtraction. The final prediction is:

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} S(y). \quad (11)$$

This contrastive score rewards candidates supported by the full-history Expert context and penalizes candidates favored by the weakened-history Amateur context. In this way, the model can use reasoning-derived preference cues while reducing the Linguistic Inertia on the final SID ranking.

4 Experiments

4.1 Experimental Setup

Datasets and Metrics. We evaluate on three large-scale industrial recommendation datasets from RecIF-Bench (Zhou et al., 2025a)¹: **AD**, **Product**, and **Video**, covering advertising click prediction, e-commerce purchase prediction, and short-video engagement prediction, respectively. Items are represented as hierarchical Semantic ID triplets $\langle s_a, s_b, s_c \rangle$, where each level has an 8,192-token vocabulary. User histories longer than 100 interactions are truncated to the most recent 100 interactions. We report Recall@ K and NDCG@ K for $K \in \{5, 10\}$ on the official test splits. Dataset statistics are provided in Appendix A.

Baselines. We compare with three categories of recommendation models. (i) *Discriminative recommenders*: **SASRec** (Kang and McAuley, 2018), **HSTU** (Zhai et al., 2024), and **ReaRec** (Tang et al., 2025). (ii) *Generative recommenders*: **TIGER** (Rajput et al., 2023) and **LC-Rec** (Zheng et al., 2024). (iii) *Foundation recommender models*: **OpenOneRec** (Zhou et al., 2025a) at both 1.7B and 8B scales. For OpenOneRec, we evaluate both the non-thinking mode (*ThinkOff*) and the raw reasoning mode (*ThinkOn*). All non-OpenOneRec baselines are trained or evaluated following their domain-specific benchmark protocols, while OpenOneRec is used as released without additional domain-specific fine-tuning.

Implementation. Our method is training-free and is instantiated on OpenOneRec-1.7B and OpenOneRec-8B. The reasoning-chain compressor is Qwen2.5-7B-Instruct used off-the-shelf, without recommendation-specific training. All hyperparameters are selected on the validation split and then fixed for test evaluation. Detailed implementation settings and search ranges are reported in Appendix A.

¹<https://huggingface.co/datasets/OpenOneRec/OpenOneRec-RecIF>

4.2 Main Results

Table 1 shows that our method consistently achieves the best overall performance across all datasets and model scales. First, all discriminative and generative recommenders are substantially outperformed by LLM-as-Rec models such as OpenOneRec, even including LLM-based baselines such as LC-Rec. This highlights the strong advantage of foundation-scale pre-training for recommendation.

Second, the comparison between OpenOneRec *ThinkOn* and *ThinkOff* shows that directly enabling reasoning often degrades recommendation performance rather than improving it. This supports our finding in Section 2: raw CoT can induce linguistic inertia, shifting SID prediction away from interaction-driven evidence toward textual reasoning signals. In contrast, our method recovers from this degradation and further surpasses the *ThinkOff* baseline, indicating that calibrated reasoning can preserve useful preference cues while suppressing text-dominant bias. The relatively smaller improvement on Video is likely because the base model often fails to activate thinking mode in this domain, leaving limited reasoning signal and thus less linguistic inertia to correct.

Third, the gains are more pronounced on the 1.7B model than on the 8B model. This suggests that smaller models are more vulnerable to reasoning-induced attention distraction and therefore benefit more from our correction mechanism.

Finally, since LICD is designed as a training-free inference-time calibration method, its practicality depends on whether the additional compression and contrastive scoring introduce acceptable overhead. We therefore report a latency analysis in Appendix 4.6, showing that the added cost is moderate compared with the *ThinkOn* pipeline.

4.3 Ablation Study

To evaluate the contribution of each component, we conduct an ablation study on OpenOneRec-1.7B over the AD and Product datasets. We consider two CoT-processing variants and two decoding variants. For CoT processing, (1) *w/o Compression* replaces the compressed CoT $\hat{c}$ with raw reasoning chain c in the Expert view, while (2) *Short Thinking* prompts the model to generate a shorter reasoning chain without post-hoc compression. For decoding, (a) *w/o Contrastive* removes the Amateur penalty by setting $\alpha = 0$, while (b)

	Method	AD				Product				Video			
		R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
Disc.	SASRec	0.3609	0.6689	0.3403	0.4570	0.1701	0.3038	0.2365	0.2451	0.1881	0.3649	0.2932	0.3472
	HSTU	0.4593	0.8548	0.4337	0.5848	0.1862	0.3491	0.2352	0.2756	0.2029	0.3886	0.3224	0.3552
	ReaRec	0.4121	0.7207	0.3959	0.5121	0.2211	0.3954	0.2814	0.3183	0.1967	0.4068	0.3110	0.3653
Gen.	TIGER	0.3973	0.6742	0.4109	0.5107	0.1258	0.2287	0.1478	0.1795	0.2288	0.4133	0.3920	0.4090
	LC-Rec	0.5796	1.0047	0.5730	0.7271	0.2765	0.5185	0.3468	0.4141	0.1568	0.3637	0.2627	0.3202
OpenOneRec-1.7B	<i>ThinkOff</i>	1.7825	2.9611	1.7210	2.1437	1.9522	3.0202	2.4071	2.5966	1.1805	2.0120	1.9114	1.9161
	<i>ThinkOn</i>	1.1400	1.9996	1.0993	1.4090	1.4524	2.2759	1.7974	1.9510	1.1596	1.9994	1.9795	1.9749
	LICD	1.9104	3.1701	1.8791	2.3212	2.1307	3.3102	2.6949	2.8921	1.2068	2.0420	2.0170	2.0089
	Δ	+7.18	+7.06	+9.19	+8.28	+9.14	+9.60	+11.96	+11.38	+2.23	+1.49	+1.89	+1.72
OpenOneRec-8B	<i>ThinkOff</i>	2.3267	3.9311	2.2941	2.8560	2.4274	3.7792	3.0245	3.2798	1.5016	2.0420	2.9910	2.4629
	<i>ThinkOn</i>	1.2301	2.0836	1.1957	1.4965	1.7072	2.6095	2.1238	2.2785	1.4963	2.0184	2.9273	2.4194
	LICD	2.4891	4.0424	2.3764	2.9316	2.5423	3.9019	3.1035	3.3727	1.5417	2.0960	3.0785	2.5365
	Δ	+6.98	+2.83	+3.59	+2.65	+4.73	+3.25	+2.61	+2.83	+2.67	+2.64	+2.93	+2.99

Table 1: Main results on three datasets. $R@K$ and $N@K$ denote Recall@ K and NDCG@ K , respectively. All metric values are multiplied by 100. Δ reports the relative improvement (%) of **LICD** over OpenOneRec without thinking. Improvements are statistically significant at $p < 0.05$.

Variant	AD		Product	
	R@5	N@5	R@5	N@5
<i>OpenOneRec-1.7B</i>				
- <i>ThinkOn</i>	1.140	1.099	1.452	1.797
- <i>ThinkOff</i>	1.783	1.721	1.952	2.407
<i>CoT Processing Variants</i>				
(1) w/o Compression	1.721	1.700	2.028	2.511
(2) Short Thinking	1.768	1.706	2.031	2.572
<i>Decoding Variants</i>				
(a) w/o Contrastive	1.831	1.808	2.050	2.516
(b) Amateur Only	1.461	1.412	1.858	2.338
Full (LICD)	1.910	1.879	2.131	2.695

Table 2: Ablation study. Values are $\times 100$.

Amateur Only ranks candidates using only the Amateur score z_A . We report *ThinkOff* and *ThinkOn* of OpenOneRec-1.7B as reference points.

Table 2 reports the results. The CoT-processing variants show the necessity of explicit compression. Variant (1) underperforms Full, indicating that directly using raw CoT in the Expert view reintroduces linguistic inertia. Variant (2) also performs worse than Full, suggesting that simply shortening reasoning is insufficient; the CoT still needs to be distilled into a stable preference control for SID prediction.

The decoding variants verify the role of con-

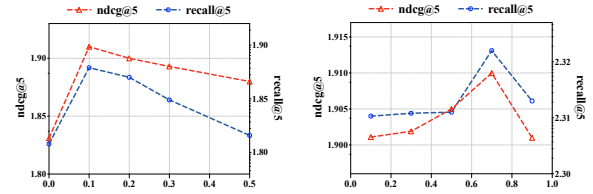


Figure 5: Sensitivity analysis on AD with respect to contrastive strength α and Amateur mask ratio ρ .

trastive calibration. Variant (a) improves over raw *ThinkOn* but remains below Full, showing compression alone cannot eliminate the text-dominant bias. Variant (b) substantially underperforms Full and *ThinkOff*, confirming the Amateur view exposes linguistic inertia under weakened-history conditions and should serve as a negative reference rather than a standalone predictor.

4.4 Hyperparameter Sensitivity

Our method mainly involves two inference-time hyperparameters: the contrastive strength α in Eq. 10 and the mask ratio ρ for constructing the Amateur context in Eq. 8. Each parameter is swept independently while keeping the others fixed.

Contrastive strength α . The left panel of Fig. 5 shows that a small positive α improves Recall@5 and NDCG@5, indicating the Amateur view pro-

vides a useful negative signal for correcting linguistic inertia. When α is too small, the text-dominant bias exposed by the Amateur context is insufficiently removed. As α increases to a moderate range, this bias is better calibrated, and the ranking becomes more reliable. However, large α gradually hurts performance, suggesting that the Amateur distribution may still contain useful coarse preference cues and should not be over-penalized. Overall, the method performs well within a broad small- α range, showing the contrastive strength does not require precise tuning.

Mask ratio ρ . The right panel of Fig. 5 shows that performance remains stable across a wide range of mask ratios. This suggests that the Amateur context does not rely on a specific masking level. Instead, it only needs to weaken the historical SID signal enough to reveal the model’s text-dominant tendency. The best performance appears around a moderate-to-large mask ratio, where the history signal is sufficiently weakened while the remaining context can still support meaningful CoT generation. These results indicate that our method is robust to the choice of ρ and can be applied without extensive per-domain tuning.

4.5 Layer-wise SID Decomposition

To further examine how Linguistic Inertia affects hierarchical SID generation, we analyze the impact of ThinkOn at different Semantic ID levels. Each item is represented as a three-level SID triplet $\langle s_a, s_b, s_c \rangle$, where s_a , s_b , and s_c capture coarse-, middle-, and fine-grained semantics, respectively. Since SIDs are generated autoregressively from coarse to fine, later levels require more precise SID-grounded evidence from user history to distinguish among similar items.

For each level $l \in \{a, b, c\}$, we compare the rank of the ground-truth SID token under *ThinkOff* and *ThinkOn*. An instance is counted as hurt if *ThinkOn* assigns a worse rank than *ThinkOff*, and we report the corresponding hurt rate.

As shown in Table 3, *ThinkOn* hurts all SID levels, but the degradation becomes stronger at finer levels. On AD, the hurt rate increases from 59.4% at the coarse level s_a to 68.8% at the fine level s_c . Product shows the same trend, increasing from 58.1% to 68.0%. This indicates that raw thinking does not uniformly affect hierarchical SID prediction; instead, its negative effect is amplified as generation moves from coarse semantic prediction to

Dataset	s_a (Coarse)	s_b (Middle)	s_c (Fine)
AD	59.4	64.4	68.8
Product	58.1	62.3	68.0

Table 3: Layer-wise hurt rate of *ThinkOn* compared with *ThinkOff*. Values are percentages.

fine-grained item matching.

This trend supports our interpretation that CoT mainly provides coarse-grained preference priors, such as category interests or general consumption intent. Such priors may help narrow the candidate space at the early SID level. However, accurate recommendation further requires selecting the correct fine-grained item within that broad preference direction, which depends more heavily on collaborative signals and item-level evidence encoded in the historical SID sequence. When the textual prior introduced by CoT is over-amplified, the model tends to follow natural-language preference descriptions and underuse the historical SID evidence needed for fine-grained discrimination. Therefore, the ground-truth fine-level token is more likely to be ranked lower under *ThinkOn*.

These findings further motivate LICD. Rather than discarding CoT, LICD calibrates its influence: reasoning-chain compression preserves useful coarse preference information, while bias-subtracted contrastive inference suppresses the text-dominant shift. In this way, CoT is used as an auxiliary coarse-level prior without overriding SID-grounded evidence for item-level prediction.

4.6 LICD Efficiency Analysis

Motivation. LICD adds compression and three-view scoring on top of the *ThinkOn* pipeline. We quantify the additional inference latency to assess its practical deployability.

Method. We estimate latency using the measured generation speed, i.e., 50 tokens/s for 1.7B and 20 tokens/s for 8B, and the measured prefill speed, i.e., 3000 tokens/s for 1.7B and 1500 tokens/s for 8B. We compare the latency of *ThinkOff*, *ThinkOn*, and our full method.

Analysis. Table 4 shows that our method introduces only moderate latency overhead over *ThinkOn*, ranging from 15% to 19% across datasets and model scales. This is because the dominant cost, CoT generation, is already incurred by the *ThinkOn* pipeline. Our method does not

Method	Product		AD	
	1.7B	8B	1.7B	8B
ThinkOff latency (s)	3.1	7.2	2.7	6.2
ThinkOn latency (s)	26.1	64.6	26.1	64.8
LICD latency (s)	31.0	74.9	30.9	74.4
Ours / ThinkOn	1.19×	1.16×	1.18×	1.15×

Table 4: Latency comparison. Ours/ThinkOn denotes the relative latency compared with ThinkOn.

generate longer reasoning chains; instead, it compresses the existing reasoning output and performs additional scoring passes for contrastive reranking. Therefore, the extra cost mainly comes from the compression step and the prefill computation for the Expert and Amateur views.

Conclusion. Our method achieves better recommendation performance than both *ThinkOn* and *ThinkOff* while adding less than 19% latency over *ThinkOn*.

5 Related Work

LLM-Based Recommendation Empowered by deep semantic understanding, Large Language Models (LLMs) have catalyzed a paradigm shift in recommendation systems. Early approaches directly elicited recommendation results via in-context learning or prompt engineering (Achiam et al., 2023; Gao et al., 2023; Sun et al., 2024; Liu et al., 2023). To better inject domain-specific collaborative knowledge, instruction tuning emerged as a prevailing paradigm (e.g., P5 (Geng et al., 2022), TALLRec (Bao et al., 2023)), aligning LLMs with recommendation tasks by reformulating user data into natural language sequences. However, verbose text representations struggle to efficiently encode collaborative signals. Consequently, a paradigm shift is occurring towards using discrete Semantic IDs to represent items (Zhou et al., 2025a; Liu et al., 2025; Rajput et al., 2023; Zheng et al., 2024).

Reasoning-Enhanced Recommendation In LLM-based recommendation, reasoning capabilities are pivotal for capturing user preferences and generating explainable outcomes (Kim et al., 2024b,a; Zhou et al., 2025b,a). While some explore implicit reasoning (Tang et al., 2025; Zhang et al., 2025), recent research predominantly prioritizes explicit Chain-of-Thought (CoT) processes. Since raw interaction data lacks reasoning ground truth, the dominant paradigm

utilizes capable LLMs as teachers to generate synthetic rationales (Fang et al., 2025; You et al., 2025; Sabouri et al., 2025; Tsai et al., 2024; Bismay et al., 2025). To mitigate teacher-induced noise, works like OneRec-think (Liu et al., 2025) incorporate retrieval mechanisms for autonomous logical chain construction.

6 Conclusion

In this paper, we investigated the failure of CoT reasoning in foundation recommender models. We found that textual reasoning before SID generation introduced *Linguistic Inertia*, which made the model depend more on natural-language context and less on historical SID evidence. We showed that this problem became more serious with less historical information and longer CoT segments. To mitigate it, we proposed LICD, a training-free decoding method that combined reasoning-chain compression with bias-subtracted contrastive inference. Experiments on three large-scale benchmarks demonstrated that LICD outperformed both no-thinking and original-thinking baselines.

Limitations

Although our proposed LICD offers a practical inference-time solution for mitigating reasoning-induced shifts in large recommendation models, several issues remain for future study. Our experiments are conducted on three representative datasets covering advertising, e-commerce, and video recommendation, and further work is needed to verify whether similar shift patterns hold across broader domains and more diverse item representations. Moreover, this study mainly focuses on text-based reasoning in sequential recommendation, while reasoning in multimodal or conversational settings may interact with richer signals such as visual content, user behavior, and dialogue context. We will continue to explore these directions to better understand the scope and characteristics of reasoning-induced shifts.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 62472394, 62441239, U23A20319, and 62441227), as well as the Anhui Province Science and Technology Innovation Project (Nos. 202423k09020010 and 202423k09020011).

References

- Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.
- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Keqin Bao, Jizhi Zhang, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2023. Tallrec: An effective and efficient tuning framework to align large language model with recommendation. In *Proceedings of the 17th ACM conference on recommender systems*, pages 1007–1014.
- Millennium Bismay, Xiangjue Dong, and James Caverlee. 2025. Reasoningrec: Bridging personalized recommendations and human-interpretable explanations through llm reasoning. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 8132–8148.
- Yi Fang, Wenjie Wang, Yang Zhang, Fengbin Zhu, Qifan Wang, Fuli Feng, and Xiangnan He. 2025. Reason4rec: Large language models for recommendation with deliberative user preference alignment. *arXiv preprint arXiv:2502.02061*.
- Yunfan Gao, Tao Sheng, Youlin Xiang, Yun Xiong, Haofen Wang, and Jiawei Zhang. 2023. Chatrec: Towards interactive and explainable llms-augmented recommender system. *arXiv preprint arXiv:2303.14524*.
- Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as language processing (rlp): A unified pretrain, personalized prompt & predict paradigm (p5). In *Proceedings of the 16th ACM conference on recommender systems*, pages 299–315.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.
- Jieyong Kim, Hyunseo Kim, Hyunjin Cho, SeongKu Kang, Buru Chang, Jinyoung Yeo, and Dongha Lee. 2024a. Review-driven personalized preference reasoning with large language models for recommendation. corr, abs/2408.06276, 2024. doi: 10.48550/arXiv:2408.06276.
- Minjin Kim, Minju Kim, Hana Kim, Beong-woo Kwak, Soyeon Chun, Hyunseo Kim, SeongKu Kang, Youngjae Yu, Jinyoung Yeo, and Dongha Lee. 2024b. Pearl: A review-driven persona-knowledge grounded conversational recommendation dataset. *arXiv preprint arXiv:2403.04460*.
- Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149*.
- Zhanyu Liu, Shiyao Wang, Xingmei Wang, Rongzhou Zhang, Jiaxin Deng, Honghui Bao, Jinghao Zhang, Wuchao Li, Pengfei Zheng, Xiangyu Wu, and 1 others. 2025. Onerec-think: In-text reasoning for generative recommendation. *arXiv preprint arXiv:2510.11639*.
- Andrzej Maćkiewicz and Waldemar Ratajczak. 1993. Principal components analysis (pca). *Computers & Geosciences*, 19(3):303–342.
- Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, and 1 others. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems*, 36:10299–10315.
- Milad Sabouri, Masoud Mansoury, Kun Lin, and Bamshad Mobasher. 2025. Towards explainable temporal user profiling with llms. In *Adjunct Proceedings of the 33rd ACM Conference on User Modeling, Adaptation and Personalization*, pages 219–227.
- Zhu Sun, Hongyang Liu, Xinghua Qu, Kaidong Feng, Yan Wang, and Yew Soon Ong. 2024. Large language models for intent-driven session recommendations. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 324–334.
- Jiakai Tang, Sunhao Dai, Teng Shi, Jun Xu, Xu Chen, Wen Chen, Jian Wu, and Yuning Jiang. 2025. Think before recommend: Unleashing the latent reasoning power for sequential recommendation. *arXiv preprint arXiv:2503.22675*.
- Alicia Tsai, Adam Kraft, Long Jin, Chenwei Cai, Anahita Hosseini, Taibai Xu, Zemin Zhang, Lichan Hong, Ed H Chi, and Xinyang Yi. 2024. Leveraging llm reasoning enhances personalized recommender systems. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13176–13188.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Runyang You, Yongqi Li, Xinyu Lin, Xin Zhang, Wenjie Wang, Wenjie Li, and Liqiang Nie. 2025. R²ec: Towards large recommender models with reasoning.

In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Jiaqi Zhai, Lucy Liao, Xing Liu, Yueming Wang, Rui Li, Xuan Cao, Leon Gao, Zhaojie Gong, Fangda Gu, Michael He, and 1 others. 2024. Actions speak louder than words: Trillion-parameter sequential transducers for generative recommendations. *arXiv preprint arXiv:2402.17152*.

Junjie Zhang, Beichen Zhang, Wenqi Sun, Hongyu Lu, Wayne Xin Zhao, Yu Chen, and Ji-Rong Wen. 2025. Slow thinking for sequential recommendation. *arXiv preprint arXiv:2504.09627*.

Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pages 1435–1448. IEEE.

Guorui Zhou, Honghui Bao, Jiaming Huang, Jiaxin Deng, Jinghao Zhang, Junda She, Kuo Cai, Lejian Ren, Lu Ren, Qiang Luo, and 1 others. 2025a. Openonerec technical report. *arXiv preprint arXiv:2512.24762*.

Jingyi Zhou, Cheng Chen, Kai Zuo, Manjie Xu, Zhendong Fu, Yibo Chen, Xu Tang, and Yao Hu. 2025b. Hymirec: A hybrid multi-interest learning framework for llm-based sequential recommendation. *arXiv preprint arXiv:2510.13738*.

A Detailed Experimental Setup

A.1 Evaluation Data Statistics

We evaluate on the official evaluation splits of the RecIF-Bench benchmark (Zhou et al., 2025a). The benchmark contains three industrial recommendation domains: **AD**, **Product**, and **Video**. All three domains encode each item as a hierarchical Semantic ID triplet $\langle s_a, s_b, s_c \rangle$, where each level has a vocabulary size of 8,192. For all methods, user interaction histories exceeding 100 interactions are truncated to the most recent 100 interactions. Table 5 summarizes the evaluation data statistics. The three domains differ substantially in catalog size and interaction density, ranging from 79K items in AD to over 1.5M items in Video, thus providing a diverse testbed for evaluating recommendation performance across domains.

A.2 Training Data Statistics

Table 6 reports the training data statistics for the per-dataset baselines, including discriminative recommender models and generative recommender models. Specifically, SASRec (Kang and McAuley, 2018), HSTU (Zhai et al., 2024),

ReaRec (Tang et al., 2025), TIGER (Rajput et al., 2023), and LC-Rec (Zheng et al., 2024) are independently trained on the corresponding domain-specific training split from RecIF-Bench (Zhou et al., 2025a). The training sets vary significantly in scale: AD contains 4.2M interactions over 161K items, Product contains 16.1M interactions over 1.79M items, and Video contains 75.7M interactions over 11.59M items.

OpenOneRec (Zhou et al., 2025a), categorized as an LLM-as-Rec model in our experiments, follows a different setting. It is pre-trained on the full cross-domain corpus released with RecIF-Bench, which contains approximately 120M interactions from 200K users across the AD, Product, and Video domains. We use the released OpenOneRec checkpoints directly and do not perform domain-specific fine-tuning.

A.3 Baseline Details

We compare against three groups of recommendation models.

Discriminative recommender models.

- **SASRec** (Kang and McAuley, 2018) is a self-attentive sequential recommendation model that predicts target items from users’ historical interaction sequences.
- **HSTU** (Zhai et al., 2024) is a sequential transduction model designed for large-scale recommendation.
- **ReaRec** (Tang et al., 2025) enhances sequential recommendation with reasoning-augmented user preference modeling.

Generative recommender models.

- **TIGER** (Rajput et al., 2023) formulates recommendation as generative retrieval over semantic item IDs.
- **LC-Rec** (Zheng et al., 2024) adapts LLMs for recommendation by incorporating collaborative semantics.

LLM-as-Rec models.

- **OpenOneRec** (Zhou et al., 2025a) is an open-source foundation recommendation model trained on the RecIF-Bench cross-domain corpus. Unlike per-domain baselines, OpenOneRec is pre-trained across multiple

Domain	#Users	#Items	#Inter.	Avg. Hist.	Avg. Tgt.
AD	27,677	79,185	1,050K	32.5	5.4
Product	27,910	365,228	1,890K	60.9	6.8
Video	38,781	1,577,337	4,224K	100.0	8.9

Table 5: Statistics of evaluation datasets. Histories exceeding 100 interactions are truncated to the most recent 100. Avg. Hist. denotes the average number of historical interactions per user, and Avg. Tgt. denotes the average number of target items per user.

Domain	#Users	#Items	#Inter.	Avg. Hist.	Avg. Tgt.
AD	110,858	161,305	4.2M	32.6	5.4
Product	113,210	1,792,738	16.1M	135.6	6.8
Video	156,245	11,589,098	75.7M	475.4	8.9

Table 6: Training data statistics for per-dataset baselines. Avg. Hist. denotes the average number of historical interactions per user, and Avg. Tgt. denotes the average number of target items per user.

recommendation domains and directly generates recommendations through hierarchical Semantic IDs. We evaluate OpenOneRec at both 1.7B and 8B scales.

For SASRec, HSTU, ReaRec, TIGER, and LC-Rec, we train one model per domain using the corresponding domain-specific training split. For OpenOneRec, we directly use the released pre-trained checkpoints without additional domain-specific fine-tuning.

A.4 Implementation Details and Hyperparameter Search

Our method is training-free and is applied to OpenOneRec-1.7B and OpenOneRec-8B. The reasoning-chain compressor is implemented with Qwen2.5-7B-Instruct and used off-the-shelf without fine-tuning. All evaluations are performed on GPUs.

Candidate generation uses beam search with beam width $B = 32$. We report Recall@ K and NDCG@ K for $K \in \{5, 10\}$.

For our method, we conduct a grid search over the following hyperparameters:

$$\alpha \in \{0.00, 0.05, 0.10, 0.15, 0.20, 0.30, 0.50, 0.70, 1.00\},$$

$$\rho \in \{0.10, 0.30, 0.50, 0.70, 0.90\}.$$

Here, α is the contrastive strength in Eq. 10, ρ is the mask ratio for constructing the amateur context.

B Details of the Empirical Analysis Experiments

This section provides full details of the empirical analysis experiments in Section 2 to ensure reproducibility.

Model and data. All experiments use the OpenOneRec-1.7B model (Qwen3ForCausalLM architecture) with 24,576 Semantic ID tokens organized in a three-level hierarchy ($\langle s_a_* \rangle$, $\langle s_b_* \rangle$, $\langle s_c_* \rangle$, 8,192 tokens each). We evaluate on two datasets: **Product** (e-commerce item recommendation) and **AD** (advertisement recommendation), using 300 randomly sampled test instances per dataset.

Attention extraction. We extract attention weights from the last Transformer layer at the prediction position (the token immediately following $\langle |sid_begin| \rangle$). Attention is averaged across all heads to obtain a single distribution over the input sequence. The SID Attention Ratio is computed as the sum of attention weights on all SID-related token positions (including $\langle |sid_begin| \rangle$, $\langle |sid_end| \rangle$, and all $\langle s_a/b/c_* \rangle$ tokens) divided by the total attention mass.

Drift-bias correlation. For each sample, we compute logits over all 8,192 first-level SID tokens ($\langle s_a_* \rangle$) under three conditions: (1) history only (x), (2) history + CoT (x, c), and (3) CoT only (history replaced with a generic instruction). The drift vector δ and bias vector $\mathbf{b}$ are computed in this 8,192-dimensional logit space, and Pearson r is computed per sample then averaged.

CoT generation. Reasoning chains are generated with temperature 0.7, do_sample=True, and a maximum length of 256 tokens. The $\langle \text{think} \rangle$ token marks the end of reasoning. If the model does not produce $\langle \text{think} \rangle$ within 256 tokens, we append it manually.

Masking protocol (Section 2.2). Token-level random masking replaces selected positions with the `<|endoftext|>` token (token ID 151643, the model’s EOS/padding token). Masking is applied at 10 ratios: $\{0\%, 10\%, 20\%, \dots, 90\%\}$.

- **Mask History:** For each sample, we identify all SID token positions in the user message (before the assistant turn). A random subset of these positions (determined by the mask ratio) is replaced with the padding token. The CoT is generated once from the full history and remains fixed across all mask ratios.
- **Mask CoT:** The full CoT is generated from the unmasked history. A random subset of CoT token positions (excluding the final `</think>` token) is replaced with the padding token. The history remains intact.

Both conditions preserve the original sequence length and positional encodings. A fixed random seed (42) ensures reproducibility, with a separate RNG instance for masking operations.

CoT length experiment (Section 2.2, Finding 3). For each sample, we generate the full reasoning chain (up to 256 tokens) from the unmasked history. We then truncate the CoT at 8 lengths: $\{32, 64, 96, 128, 160, 192, 224, 256\}$ tokens. At each truncation point, the last token is replaced with `</think>` to maintain a well-formed reasoning boundary. The history remains intact across all truncation lengths. We measure the SID Attention Ratio and drift–bias correlation at each length, using the same attention extraction and logit computation procedures described above.

Statistical significance. All reported trends are verified with Wilcoxon signed-rank tests comparing the 0% and 75% mask conditions ($p < 0.01$ for all primary metrics on the Product dataset).

C Subspace Visualization

To provide intuition for the partial alignment between the Semantic ID Subspace and the General Subspace, we perform Principal Component Analysis (PCA) (Maćkiewicz and Ratajczak, 1993; Abdi and Williams, 2010) on the token embeddings from the model’s embedding layer. As shown in Fig. 6, the two subspace distributions exhibit partial overlap—indicating that pre-training achieves some degree of alignment—yet

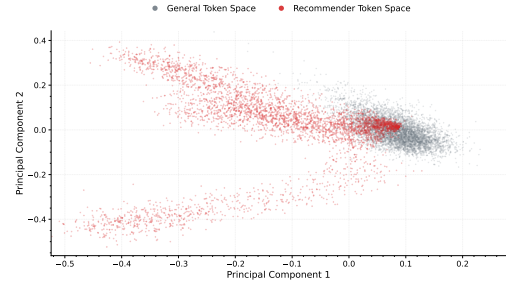


Figure 6: PCA visualization of token embeddings from the Semantic ID Subspace and the General Subspace. The two distributions are partially aligned but remain structurally distinct.

remain semantically distinct. This structural separation explains why relying primarily on general-subspace tokens (e.g., extended CoT) may reduce access to semantic-ID-specific evidence during decoding.