KNIGHT: Key-attended Controlled Decoding with Topological Enhancement for Uncertainty Quantification in LLM-based Recommendation

Anonymous ACL submission

Abstract

This paper studies the problem of uncertainty quantification in LLM-based recommendation, which aims to measure the reliability of LLMs for recommender systems. Previous works typically measure the uncertainty based on predictive distributions of output tokens. Despite the progress, their performance is far from satisfactory due to biased token distributions across different prompts and the neglect of the semantic structures of items. Towards this end, we propose a novel approach named Key-attended Controlled Decoding with Topological Enhancement (KNIGHT) for uncertainty quantification in LLM-based recommendation. The core of our KNIGHT is to utilize controlled decoding to naturally generate multiple outputs with diverse key-attended emphases for topology-enhanced uncertainty quantification. In particular, our KNIGHT first debiases target token logits with candidate-label prior and mean token logits under key-attended prompts. More importantly, we train a lightweight probe via contrastive learning to provide key-agnostic scores, which are combined with key-attended scores for candidate item ranking. Then, we not only measure the ranking differences across keys, but also calculate their average distance between corresponding item-title embeddings. The joint consistency-aware scores from two views would be fused with calibrated entropy for final scores. Extensive experiments across multiple datasets validate the effectiveness of the proposed KNIGHT.

1 Introduction

Large language models (LLMs) are increasingly used as candidate-conditioned list-wise recommendation rankers (Wu et al., 2024; Geng et al., 2022; Bao et al., 2023; Hou et al., 2024; Lyu et al., 2024). Given a user's chronological history and candidate set, the LLM outputs a candidate index or ranking. This candidate-index formulation exploits item semantics while avoiding unconstrained item generation (Wang and Lim, 2023).

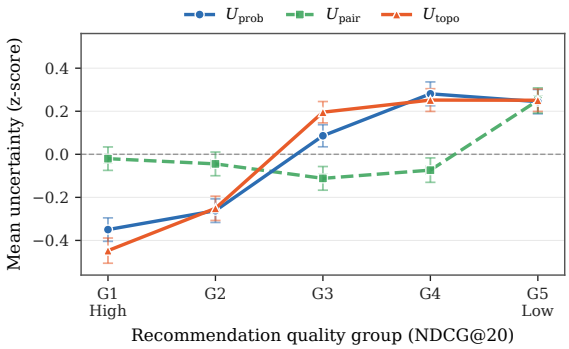


Figure 1: **Uncertainty versus recommendation quality.** Users are grouped by final NDCG@20 from high-quality (G1) to low-quality (G5). All uncertainty scores are z-scored across users.

Despite this promise, LLM-based recommendation remains unreliable. Small changes in candidate order (Hou et al., 2024; Bito et al., 2025), prompt wording (Sclar et al., 2024), history context, or recommendation criteria (Lyu et al., 2024) can alter the ranking. Thus, beyond asking *what* the model recommends, we must ask *how reliable* the recommendation is. Figure 1 illustrates this problem: as users move from high- to low-NDCG groups, U_{prob} and U_{topo} rise overall, while U_{pair} is noisier. This suggests that unreliable recommendations often show broader semantic dispersion across perturbed views, motivating structural uncertainty signals beyond static candidate-label confidence.

Uncertainty quantification has been effective for LLMs in classification (Ling et al., 2024; Tian et al., 2023; Xiao et al., 2022), question answering (Kuhn et al., 2023; Wang et al., 2022), and open-ended generation (Kang et al., 2026; Xiong et al., 2024). However, it is much less developed for LLM-based recommendation. General-purpose uncertainty quantification methods, such as verbalized confidence, semantic entropy, or entailment-graph consistency, are not tailored to candidate-

conditioned ranking. The closest recommendation-specific framework, Kweon et al. (2025), estimates ranking uncertainty from candidate top-1 probabilities with a Plackett–Luce approximation and evaluates whether lower uncertainty predicts higher NDCG. This is effective and efficient, but remains largely observational: it measures variability under sampled prompts and candidate orders without actively probing the model's local candidate-index decision boundary.

Two challenges make uncertainty quantification for recommendations structurally different from the standard one. First, recommendation outputs are rankings over mutually exclusive items rather than independent textual answers or semantically equivalent responses; the output space grows factorially with the number of candidates, and reliability is tied to the rank of a held-out item (Wu et al., 2024; Hou et al., 2024; Yue et al., 2023). Second, candidate-index prompting introduces label-position artifacts: alphabet labels can receive strong prior logits unrelated to item content, contaminating both NDCG and uncertainty estimates (Zhao et al., 2021; Hou et al., 2024). A reliable uncertainty quantification protocol should therefore keep the final recommendation ranking fixed, correct candidate-label bias, and measure instability in a space appropriate for competing item choices rather than textual entailment claims.

We propose a novel approach named Key-attended Controlled Decoding with Topological Enhancement (KNIGHT) for uncertainty quantification in LLM-based Recommendation. We use *Controlled Decoding* (CD) (Mudgal et al., 2023) as an inference-time reliability stress test: a lightweight auxiliary scorer perturbs candidate-index logits under recommendation keys such as genre, popularity, novelty, and similarity. Crucially, CD does not change the final recommendation ranking; the final ranking is computed from calibrated base probabilities, while CD-generated views are used only to measure reliability. From these views, KNIGHT estimates three uncertainty signals: probability entropy U_{prob}, pairwise rank instability U_{pair}, and latent semantic topology dispersion U_{topo}. The topology signal replaces textual entailment graphs with dispersion over perturbed top-item modes in latent item space, aligning uncertainty with the structure of recommendation. Overall, the results show that KNIGHT delivers consistent reliability gains across datasets and backbones.

Our contributions are as follows:

❶ *New Perspective.* We view controlled decoding as an inference-time reliability stress test for LLM-based recommendation: it probes a frozen candidate-index ranker through key-attended controlled views while keeping the final recommendation ranking fixed.

❷ *New methodology.* We propose KNIGHT, a validation-calibrated uncertainty estimator that combines calibrated probability entropy with controlled-decoding structural signals, including pairwise rank instability and latent semantic dispersion over perturbed top-item modes.

❸ *Empirical Analysis.* We conduct extensive experiments on three datasets and three LLM backbones under a fixed-ranking evaluation protocol, showing that KNIGHT consistently outperforms representative uncertainty baselines in reliability prediction.

2 Related Work

LLM-based Recommendation. Large language models (LLMs) have recently been adopted as recommendation rankers due to their strong ability to understand user histories, item texts, and open-domain semantic knowledge (Geng et al., 2022; Bao et al., 2023; Hou et al., 2024). Early studies adapt language-model architectures or prompting schemes to recommendation, while later work explores zero-shot, few-shot, and instruction-tuned LLM recommenders (Geng et al., 2022; Bao et al., 2023). A particularly relevant line of work formulates recommendation as list-wise candidate ranking, where an LLM ranks or selects items from a retrieved candidate set rather than generating unconstrained item names (Hou et al., 2024; Kweon et al., 2025). This candidate-conditioned formulation reduces hallucination and enables efficient re-ranking (Deldjoo et al., 2024), but it also exposes LLM recommenders to prompt sensitivity, candidate-order effects, and candidate-label bias (Lyu et al., 2024; Bito et al., 2025). Our work follows the list-wise candidate-ranking setting, but focuses on estimating the reliability of the frozen LLM recommender rather than improving the recommendation list itself.

Uncertainty Quantification for LLMs. Uncertainty quantification has demonstrated remarkable performance in measuring LLM knowledge boundaries (Kuhn et al., 2023; Chen et al., 2025) and guiding downstream adaptive behaviors (Jiang et al., 2023; Komorowski et al., 2025). Conventionally,

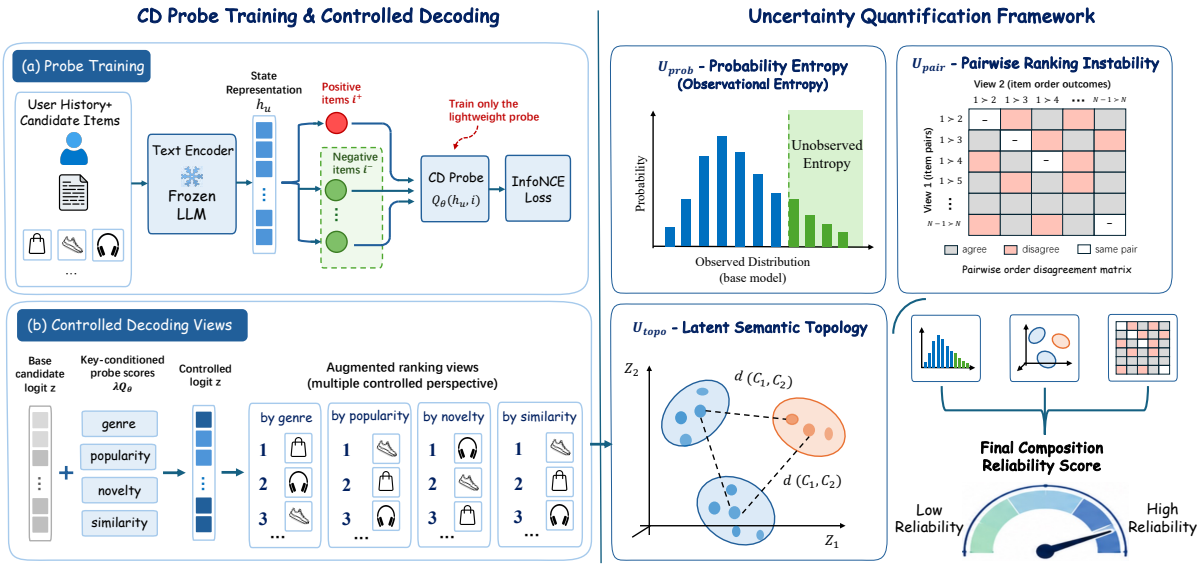


Figure 2: **Overview of KNIGHT**. The final recommendation ranking is computed from calibrated candidate-index logits and remains fixed throughout evaluation. A lightweight CD probe then produces key-conditioned augmented ranking views, from which U_{pair} and U_{topo} are computed. The final reliability score is obtained by validation-fitted fusion with U_{prob}.

it has been evaluated through token probabilities, verbalized confidence, sampling consistency, semantic entropy, and graph-based response consistency (Geng et al., 2022; Kuhn et al., 2023; Farquhar et al., 2024; Lin et al., 2023; Da et al., 2024). Most of these methods are developed for classification (Li et al., 2025; Tao et al., 2025) or question-answering (Kuhn et al., 2023; Bakman et al., 2025; Vazhentsev et al., 2025), where the output space is relatively constrained or where multiple generated responses can be grouped by semantic equivalence. In recommendation, however, the output is a ranking over candidate items, whose permutation space grows factorially with the candidate set size. To address this issue, Kweon et al. (2025) approximate the ranking distribution using candidate top-1 probabilities under a Plackett–Luce model and decompose uncertainty into recommendation and prompt uncertainty. Different from traditional approaches, we use controlled-decoding views as stress tests of the frozen recommender and measures uncertainty from calibrated probability entropy, pairwise ranking instability, and latent semantic topology dispersion.

3 The Proposed KNIGHT

Problem Formulation. Let u denote a user with chronological interaction history

$$H_u = [i_{u,1}, i_{u,2}, \ldots, i_{u,T}], \tag{1}$$

where each item i is associated with a textual representation t_i, such as a movie title, game title, or product name. In the candidate-conditioned setting, the user is evaluated with a candidate set

$$C_u = \{i_{\text{gt}}\} \cup \{i_1^-, \ldots, i_{M-1}^-\}, \tag{2}$$

where i_{gt} is the held-out next item and the remaining items are negative candidates. The candidate set is randomly ordered and assigned candidate labels $A, \ldots, T$. Given a prompt containing H_u and C_u, the frozen LLM produces next-token logits over candidate labels. We evaluate uncertainty as a reliability estimator: lower uncertainty should correspond to higher user-level recommendation quality, measured by NDCG. Throughout this paper, the final recommendation ranking is fixed by the calibrated base candidate probabilities; all controlled views are used only to estimate uncertainty.

3.1 Framework Overview

KNIGHT estimates the reliability of a frozen LLM recommender without changing its final recommendation ranking. The framework begins by deriving a calibrated base ranking from candidate-label logits. Since alphabetic candidate labels can carry strong position-dependent priors, we debias label logits on validation prompts before computing base probabilities. Built on this calibrated base scorer, KNIGHT constructs diverse key-attended ranking views by appending recommendation-key instructions, such as genre, popularity, novelty,

recency, brand, or similarity, and applying controlled decoding (CD) to stress-test the LLM's candidate-index decision boundary. These diagnostic views, together with the calibrated base distribution, are then used to compute distribution-aware and consistency-aware uncertainty signals, which are combined through validation-fitted fusion. Importantly, KNIGHT follows a protocol only for uncertainty quantification. The final ranking π_u is always obtained from base calibrated probabilities. CD views are diagnostic views: they expose local instability of the frozen ranker but are not used to re-rank candidates. Thus, all uncertainty variants share the same NDCG within each model-dataset block.

3.2 Key-attended Controlled Decoding for Diverse Candidate Ranking

A calibrated base scorer provides a fixed recommendation ranking, but reliability estimation also requires testing how stable this ranking is under plausible recommendation criteria. For each user, KNIGHT constructs key-attended views by emphasizing criteria and applies controlled decoding to perturb the candidate-index decision boundary. These augmented rankings just serve as diagnostic views for measuring whether candidate orderings and top-item semantic modes remain stable.

Token Logit Debiasing for Key-attended Scores. For each user, we draw P base prompt variants by changing candidate order, and let $z_{u,p,j}$ be the raw next-token logit of candidate label j under base prompt variant p. We estimate a candidate-label bias vector on validation prompts. For each logit vector, we remove its prompt-level mean:

$$\bar{z}_{u,p,j} = z_{u,p,j} - \frac{1}{M}\sum_{m=1}^{M} z_{u,p,m}. \tag{3}$$

The centered logits are then used to estimate a validation-fitted label bias:

$$b_j = \mathbb{E}_{(u,p)\in\mathcal{D}_{\text{val}}}[\bar{z}_{u,p,j}]. \tag{4}$$

At inference time, any base or key-attended candidate-label logit is debiased as:

$$\tilde{z}_{u,p,j} = z_{u,p,j} - \frac{1}{M}\sum_{m=1}^{M} z_{u,p,m} - b_j. \tag{5}$$

The calibrated candidate probability is then obtained by applying softmax over the debiased vector $\tilde{z}_{u,p}$:

$$q_{u,p,j} = \text{softmax}(\tilde{z}_{u,p})_j. \tag{6}$$

For the base recommender, the final candidate score is averaged over base prompt variants:

$$s_{\text{base}}(i) = \frac{1}{P}\sum_{p=1}^{P} q_{u,p}(i), \tag{7}$$

and the final ranking π_u is obtained by sorting s_{base} in descending order. The label-bias vector is fitted only from validation logits; it does not use validation NDCG or test targets. We then generate key-attended views. Let $\mathcal{K}$ denote the set of recommendation keys. For each key $k \in \mathcal{K}$, we append a key instruction to the prompt and obtain key-attended logits $z_{u,p,k,j}$. These logits are debiased by the same validation-fitted label-bias correction:

$$\tilde{z}_{u,p,k,j} = z_{u,p,k,j} - \frac{1}{M}\sum_{m=1}^{M} z_{u,p,k,m} - b_j. \tag{8}$$

This gives a key-specific view of the LLM's candidate preference while keeping candidate-label artifacts controlled.

Lightweight Probe for Key-agnostic Scores. To further stress-test the ranker, we train a lightweight item-value probe $Q_\theta(h_u, i)$ that scores the compatibility between the user state and a candidate item. The dependence on the recommendation key is provided by the key-attended prompt logits $\tilde{z}_{u,p,k,j}$. The probe itself is key-agnostic and supplies a general candidate plausibility shift. This design keeps the probe reusable across keys and LLM backbones. We obtain the user state h_u by encoding the recent item titles in H_u with a frozen sentence encoder, and encode each candidate title t_i in the same latent space. The probe is a small MLP over the concatenation of the user-state embedding and candidate-item embedding. It is trained with a listwise contrastive objective. For each training tuple, let i^+ be the observed next item and $\{i_r^-\}_{r=1}^{R}$ be sampled negatives. The InfoNCE loss is:

$$\mathcal{L}_{\text{NCE}}(u) = -Q_\theta(h_u, i^+)/\tau + \log\Big[\exp\big(Q_\theta(h_u, i^+)/\tau\big) + \sum_{r=1}^{R} \exp\big(Q_\theta(h_u, i_r^-)/\tau\big)\Big]. \tag{9}$$

We normalize the probe scores within each candidate set to obtain $\widehat{Q}_\theta(h_u, i_j)$. For key k, prompt variant p, and candidate i_j, controlled decoding combines the debiased key-attended logit with the normalized probe score:

$$\tilde{z}^{\text{CD}}_{u,p,k,j} = \tilde{z}_{u,p,k,j} + \lambda\widehat{Q}_\theta(h_u, i_j), \tag{10}$$

where λ controls the CD perturbation strength in logit space.

3.3 Fused Uncertainty Quantification with Topological Enhancement

The augmented views reveal whether the base recommendation is stable under controlled key-attended perturbations. We compute two families of uncertainty signals, a distribution-aware signal from calibrated base probabilities and consistency-aware signals from base and CD ranking views.

Distribution-aware Uncertainty. The first signal measures the entropy of the calibrated base candidate distribution. We average over the P base prompt variants to obtain:

$$\bar{q}_u(i) = \frac{1}{P}\sum_{p=1}^{P} q_{u,p}(i). \tag{11}$$

We define normalized probability uncertainty as

$$U_{\text{prob}}(u) = -\frac{1}{\log M}\sum_{i\in C_u} \bar{q}_u(i)\log(\bar{q}_u(i)+\epsilon). \tag{12}$$

This signal captures observational uncertainty in the final base distribution. It is strong when calibrated candidate probabilities are diffuse and weak when the model places most mass on a small set of candidates.

Consistency-aware Uncertainty. The second family measures instability across base and CD-augmented ranking views. Let $\mathcal{V}_u$ be the set of ranking views for user u, including the P base views and all key-attended CD views. For a candidate pair (i_a, i_b), define:

$$\rho_{ab} = \frac{1}{|\mathcal{V}_u|}\sum_{\pi\in\mathcal{V}_u} \mathbb{I}[\text{rank}_\pi(i_a) < \text{rank}_\pi(i_b)]. \tag{13}$$

The pairwise ranking instability is the average binary entropy over candidate pairs:

$$U_{\text{pair}}(u) = \frac{2}{M(M-1)}\sum_{a<b} H_2(\rho_{ab}), \tag{14}$$

where $H_2(p) = -p\log_2 p - (1-p)\log_2(1-p)$. This signal is high when pairwise orders frequently flip across controlled views.

We further compute a topological uncertainty signal in latent item space. Let $\mathcal{T}_u$ be the multiset of Top-1 items selected by views in $\mathcal{V}_u$, and let $e(i)$ be the normalized sentence-encoder embedding of item title t_i. The latent semantic topology dispersion is:

$$U_{\text{topo}}(u) = (2/M_u)\sum_{i_a,i_b\in\mathcal{T}_u,\, a<b} \big(1 - e(i_a)^\top e(i_b)\big), \tag{15}$$

where $M_u = |\mathcal{T}_u|(|\mathcal{T}_u|-1)$. If all views select the same item, $U_{\text{topo}} = 0$. If controlled views jump across semantically distant items, U_{topo} becomes large. This replaces textual entailment graphs with a latent semantic topology over item choices, which is better matched to recommendation because candidate items are competing alternatives rather than mutually entailing claims.

Finally, we fuse the three uncertainty components using statistics computed only on validation users. For each component $m \in \{\text{prob}, \text{pair}, \text{topo}\}$, we compute the validation mean μ_m and standard deviation σ_m, and standardize

$$\widehat{U}_m(u) = \frac{U_m(u) - \mu_m}{\sigma_m + \epsilon}. \tag{16}$$

The standardized components are now comparable across scales, allowing a validation-fitted linear fusion:

$$U_{\text{final}}(u) = \omega_p\widehat{U}_{\text{prob}}(u) + \omega_r\widehat{U}_{\text{pair}}(u) + \omega_t\widehat{U}_{\text{topo}}(u), \tag{17}$$

where the weights are selected on validation users and then fixed for test evaluation. In our main setting, we use model-level weights: one set of weights is selected for each backbone model and shared across datasets.

3.4 Summarization

`KNIGHT` is designed to estimate recommendation reliability without changing the recommendation itself. The calibrated base probabilities define the final ranking and the shared NDCG. Key-attended CD views provide controlled stress tests of the frozen LLM's candidate-index decision boundary. The final uncertainty combines three complementary signals: U_{prob} captures uncertainty in the calibrated base distribution, U_{pair} captures rank-space inconsistency, and U_{topo} captures semantic dispersion among perturbed top-item modes. Since calibration statistics and fusion weights are fitted only on validation users, test evaluation measures whether the resulting uncertainty score predicts recommendation reliability under a fixed-ranking protocol.

Table 1: **Reliability estimation performance under the fixed-ranking UQ setting.** Best results within each model block are in bold, and second-best results are underlined. The shared base N@20 row reports the recommendation quality of the common frozen-backbone candidate-ranking backbone.

Model	Method	MovieLens 1M				Amazon Grocery				Steam			
		τ@5	τ@20	C@5	C@20	τ@5	τ@20	C@5	C@20	τ@5	τ@20	C@5	C@20
Llama-3-8B	*Shared base N@20*	*0.5727*				*0.4864*				*0.5616*			
	Label Prob.	0.1476	0.1373	0.5844	0.5734	0.1510	0.1257	0.5911	0.5654	0.1960	0.1748	0.6147	0.5932
	Semantic Unc.	0.0808	0.0763	0.5506	0.5447	0.0963	0.0788	0.5632	0.5447	0.1267	0.1125	0.5807	0.5653
	Verb. 1S top-1	0.1285	0.1197	0.6082	0.5961	0.1234	0.1047	0.6087	0.5827	0.0732	0.0658	0.5645	0.5538
	Pred. Unc.	0.1498	0.1398	0.5857	0.5748	0.1570	0.1311	0.5948	0.5683	0.1948	0.1739	0.6140	0.5927
	KNIGHT (Ours)	**0.2006**	**0.1904**	**0.6148**	**0.6018**	**0.2159**	**0.1878**	**0.6303**	**0.5978**	**0.2953**	**0.2748**	**0.6728**	**0.6465**
Gemma-2-9B	*Shared base N@20*	*0.6030*				*0.5252*				*0.5992*			
	Label Prob.	0.1948	0.1834	0.6114	0.5994	0.1967	0.1767	0.6160	0.5929	0.2036	0.1919	0.6173	0.6039
	Semantic Unc.	0.1188	0.1116	0.5753	0.5670	0.1047	0.0889	0.5684	0.5517	0.1276	0.1193	0.5815	0.5714
	Verb. 1S top-1	0.2037	0.1961	**0.6309**	**0.6200**	0.1586	0.1447	0.6026	0.5842	0.1347	0.1292	0.5905	0.5819
	Pred. Unc.	0.1931	0.1819	0.6105	0.5986	0.1922	0.1725	0.6133	0.5908	0.2053	0.1935	0.6183	0.6048
	KNIGHT (Ours)	**0.2267**	**0.2167**	0.6297	0.6174	**0.2207**	**0.2006**	**0.6301**	**0.6055**	**0.2662**	**0.2517**	**0.6534**	**0.6363**
Mistral-v0.3-7B	*Shared base N@20*	*0.5299*				*0.4790*				*0.5728*			
	Label Prob.	0.1224	0.1122	0.5711	0.5591	0.1741	0.1517	0.6056	0.5788	0.1894	0.1757	0.6094	0.5940
	Semantic Unc.	0.0311	0.0270	0.5198	0.5156	0.0984	0.0807	0.5653	0.5461	0.0870	0.0806	0.5550	0.5471
	Verb. 1S top-1	0.1602	0.1505	0.5988	0.5839	0.1461	0.1306	0.5990	0.5761	0.1292	0.1207	0.5816	0.5704
	Pred. Unc.	0.1227	0.1128	0.5713	0.5594	0.1816	0.1606	0.6102	0.5835	0.1901	0.1769	0.6098	0.5947
	KNIGHT (Ours)	**0.1980**	**0.1877**	**0.6151**	**0.5989**	**0.2357**	**0.2140**	**0.6430**	**0.6113**	**0.2727**	**0.2567**	**0.6576**	**0.6374**

4 Experiments

4.1 Experimental Settings

Datasets and Candidate Caches. We evaluate on MovieLens 1M (Harper and Konstan, 2015), Amazon Grocery (Ni et al., 2019), and Steam (Kang and McAuley, 2018), covering media, e-commerce, and gaming domains. We use 20-core filtering for MovieLens 1M and Steam, and 10-core filtering for Amazon Grocery. For each user, the last item is held out for testing and the penultimate item for validation. Each candidate set contains 20 items with the ground-truth item included. We construct fixed validation and test candidate caches so that all methods share the same users, histories, targets, negatives, and candidate orders. Validation caches are used to fit label bias and fusion weights, while test caches are reserved for final reporting.

Models. We use three open-weight LLM backbones: Llama-3-8B (Grattafiori et al., 2024), Gemma-2-9B (Team et al., 2024), and Mistral-v0.3-7B (Jiang et al., 2024). All backbones are kept frozen. The final recommendation ranking is computed from calibrated base logits, while controlled-decoding views are used only for uncertainty estimation.

Baselines. We consider a broad range of baselines. Label Prob. (Guo et al., 2017) uses the probability of the predicted candidate label as a confidence estimate, where lower confidence indicates higher uncertainty. Semantic Unc. (Kuhn et al., 2023) follows semantic uncertainty, estimating uncertainty from the empirical distribution over semantic equivalence classes of generated outputs. Verb. 1S top-1 (Tian et al., 2023) follows verbalized confidence estimation, asking the LLM to report how likely its top-1 answer is correct and converting the reported confidence into uncertainty. Pred. Unc. (Kweon et al., 2025) denotes the predictive uncertainty method, which uses candidate top-1 probabilities and a Plackett–Luce approximation to estimate uncertainty over ranking lists.

Evaluation Metrics. Recommendation quality is measured by NDCG@K (N@K). Uncertainty reliability is measured by Kendall's $\tau@K$ (Sen, 1968) and Concordance Index $C@K$ (Uno et al., 2011), where lower uncertainty should correspond to higher user-level N@K. We report $K \in \{5, 20\}$ to evaluate whether uncertainty predicts both top-ranked quality and full-list quality.

4.2 Main Results

Evaluation Settings. Table 1 reports reliability estimation under the fixed-ranking protocol defined in Section 4.1. Within each model–dataset block, all uncertainty estimators score the same calibrated base ranking, so the shared base N@20 is fixed and methods differ only in their uncertainty scores. We report $\tau@K$ and $C@K$ for $K \in \{5, 20\}$; higher values indicate that lower uncertainty better predicts higher NDCG. KNIGHT uses model-level fu-

sion weights selected on validation users and fixed for test evaluation. Additional dataset construction, cache construction, and inference details are provided in Appendix A.4.

Overall Performance. From Table 1, we make the following observations. ❶ KNIGHT consistently improves reliability estimation. Across all model–dataset blocks, KNIGHT outperforms the strongest same-backbone probability baseline, Pred. Unc., on every reported reliability metric. This shows that static predictive entropy is informative but insufficient for identifying unreliable recommendations. ❷ Controlled-decoding topology provides complementary reliability information. The improvement is especially clear on domains where recommendation quality varies more strongly across users, such as Steam. In these settings, CD-induced semantic dispersion captures failure modes that probability confidence alone misses. Even when probability-based uncertainty is already competitive, as in the Gemma-2-9B results, KNIGHT still yields more reliable uncertainty rankings. ❸ The gains come from uncertainty estimation rather than re-ranking. All methods within the same model–dataset block share the same calibrated base ranking and therefore the same recommendation quality. Thus, the improvement of KNIGHT indicates that controlled-decoding views expose additional reliability information about a fixed frozen-backbone recommendation, rather than improving the recommendation list itself.

Further Analysis. The baselines reveal two important patterns. First, Semantic Unc. is generally weak and unstable. This is expected because semantic entropy was designed for free-form NLG, where different strings may express the same meaning. However, in recommendation, the outputs are mutually exclusive item choices, and with only five prompt samples, top-1 semantic-set entropy is a coarse signal. Second, verbal confidence (Verb. 1S top-1) can be competitive when the model follows chat-template confidence elicitation, especially on ML-1M, but it is interface-dependent and requires an additional generation pass. Its performance also varies substantially across datasets and models, suggesting that "asking the model how confident it is" is not a stable same-backbone reliability estimator. In contrast, KNIGHT is not designed to improve the shared base ranking, but it provides the most consistent reliability improvement over Pred. Unc. across model-dataset blocks. This supports our main claim: controlled-decoding topology acts as a robust structural reliability signal that complements predictive uncertainty under a fixed candidate-ranking protocol.

4.3 Ablation Studies

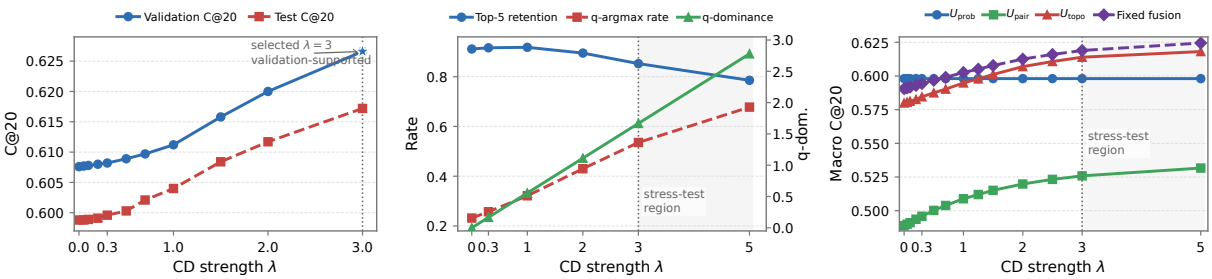
Table 2: **Ablation study of uncertainty components.** Macro scores are averaged over datasets. $\Delta C@20$ is computed against the best single component.

Variant	Macro $C@20$	Macro $\tau@20$	$\Delta C@20$
U_{prob}	0.5786	0.1483	-0.0288
U_{pair}	0.5545	0.1030	-0.0529
U_{topo}	0.6074	0.2021	0.0000
$U_{\text{prob}} + U_{\text{pair}}$	0.5905	0.1709	-0.0169
$U_{\text{pair}} + U_{\text{topo}}$	0.6143	0.2156	$+0.0069$
$U_{\text{prob}} + U_{\text{topo}}$	0.6084	0.2044	$+0.0010$
Full	**0.6154**	**0.2177**	$\mathbf{+0.0080}$

We conduct this component ablation on Llama-3-8B across the three datasets to examine the contribution of each uncertainty component. All variants are evaluated on the same fixed recommendation rankings, so the comparison isolates the quality of uncertainty estimation rather than recommendation accuracy. We report macro C@20 and macro τ@20, and compute ΔC@20 against the best single component. Table 2 shows that U_{topo} becomes the strongest single component under stronger controlled decoding. This substantially outperforms U_{prob}, indicating that latent semantic dispersion of CD-augmented views provides a stronger reliability signal than calibrated probability entropy alone. In contrast, U_{pair} remains weaker as a standalone signal, suggesting that pairwise rank instability is noisy when used without semantic or probability information. Combining components further improves performance. The best two-component variant is $U_{\text{pair}} + U_{\text{topo}}$, while the full fusion yields a 0.0080 absolute $C@20$ gain over the best single component. Although U_{pair} is weak individually, its nonzero weight in the full model suggests that it contributes complementary residual information when combined with probability and topology uncertainty.

4.4 Effect of Controlled Decoding

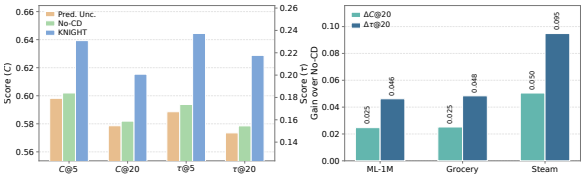
We examine whether the trained CD probe is necessary beyond key-conditioned prompt perturbations. For efficiency, this ablation is conducted on Llama-3-8B across ML-1M, Amazon Grocery, and Steam. All variants use the same candidate caches, validation-fitted label calibration, and calibrated base ranking, so NDCG is fixed and only



(a) Strict selection (b) Drift diagnostic (c) Component sensitivity

Figure 3: **Analysis on choice of CD strength λ.** (a) Strict validation-tuned ML-1M sweep for selecting λ. (b) Drift diagnostics measuring how CD views move away from the base-ranking neighborhood. (c) Component sensitivity showing that λ mainly strengthens CD-derived U_{pair} and U_{topo}, while U_{prob} remains nearly unchanged.

the uncertainty score changes. We compare ① Pred. Unc., the strongest overall non-CD baseline, ② No-CD, which uses key-conditioned prompts without the CD probe, and ③ KNIGHT, which uses the full trained probe. Figure 4 shows that key-conditioned views already provide a competitive perturbation-based reliability signal, but the trained CD probe is crucial for making these views substantially more diagnostic. KNIGHT clearly improves over No-CD, with dataset-wise gains of $\Delta C@20 = 0.025$, 0.025, and 0.050 on ML-1M, Amazon Grocery, and Steam, respectively. The corresponding $\Delta\tau@20$ gains are 0.046, 0.048, and 0.095. Thus, the CD probe does not merely add more views. With stronger control strength, it more effectively exposes local decision-boundary instability and semantic mode shifts that are predictive of recommendation reliability.



(a) Macro comparison (b) CD gain over No-CD

Figure 4: **Effect of Controlled Decoding.** (a) macro-level comparison among Pred. Unc., No-CD, and KNIGHT. (b) dataset-wise gains of CD over No-CD.

4.5 Choice of CD Strength

The CD strength λ controls how strongly the normalized probe score perturbs the calibrated key-conditioned logits. It therefore governs a trade-off between informativeness and drift. When λ is too small, CD views remain close to the base ranking and provide limited stress-test information. When λ is too large, CD views can become probe-dominant and move away from the base-ranking neighborhood. We choose λ using validation users under this drift-aware protocol. Figure 3 reports the resulting analysis. The strict ML-1M sweep re-selects fusion weights on validation users for each λ and evaluates them on held-out test users. Both validation and test reliability improve up to $\lambda = 3$, supporting it as a validation-selected operating point rather than a test-only choice. The component sensitivity plot explains where the gain comes from, with U_{prob} nearly unchanged in λ, while CD-derived U_{pair} and U_{topo} become increasingly informative. The drift diagnostic then shows why we do not keep increasing λ. Beyond the selected strong-CD setting, CD views show lower Top-5 retention and higher probe dominance. Thus, we use $\lambda = 3$ as the default setting and reserve larger values for stress-test diagnostics.

5 Conclusion

In this work, we proposed KNIGHT, a UQ-only framework for LLM-based recommendation. KNIGHT calibrates candidate-index logits to remove label-position bias and uses key-attended controlled decoding to stress-test a frozen LLM ranker without changing its final recommendation ranking. From the controlled views, it estimates probability entropy, pairwise ranking instability, and latent semantic topology dispersion. Experiments across three datasets and three LLM backbones show that KNIGHT consistently improves reliability prediction over strong uncertainty baselines, with especially strong gains when controlled decoding is used as a sharper reliability stress test.

Limitations

This work applies KNIGHT to LLM-based recommendation, where the output is a fixed candidate ranking and reliability can be evaluated

by user-level recommendation quality. This setting provides a natural testbed for controlled-decoding uncertainty quantification, since controlled views can probe the same frozen ranker under different recommendation criteria without changing the final ranking. Future work can extend KNIGHT to broader LLM-based decision-making tasks, such as decision support, planning, tool selection, retrieval-augmented generation, and other candidate-selection scenarios. In these settings, controlled decoding may also expose reliability-relevant instability, and structural dispersion across controlled views could provide uncertainty signals beyond static confidence.

References

Yavuz Bakman, Sungmin Kang, Zhiqi Huang, Duygu Nur Yaldiz, Catarina G Belém, Chenyang Zhu, Anoop Kumar, Alfy Samuel, Salman Avestimehr, Daben Liu, and 1 others. 2025. Uncertainty as feature gaps: Epistemic uncertainty quantification of llms in contextual question-answering. *arXiv preprint arXiv:2510.02671*.

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.

Ethan Bito, Yongli Ren, and Estrid He. 2025. Evaluating position bias in large language model recommendations. *arXiv preprint arXiv:2508.02020*.

Lihu Chen, Gerard de Melo, Fabian M Suchanek, and GaÃĢl Varoquaux. 2025. Query-level uncertainty in large language models. *arXiv preprint arXiv:2506.09669*.

Longchao Da, Tiejin Chen, Lu Cheng, and Hua Wei. 2024. Llm uncertainty quantification through directional entailment graph and claim level response augmentation. *arXiv preprint arXiv:2407.00994*.

Yashar Deldjoo, Zhankui He, Julian McAuley, Anton Korikov, Scott Sanner, Arnau Ramisa, René Vidal, Maheswaran Sathiamoorthy, Atoosa Kasirzadeh, and Silvia Milano. 2024. A review of modern recommender systems using generative models (gen-recsys). In *Proceedings of the 30th ACM SIGKDD conference on Knowledge Discovery and Data Mining*, pages 6448–6458.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

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.

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*.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, 5(4):1–19.

Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European conference on information retrieval*, pages 364–381. Springer.

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, and 1 others. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pages 7969–7992.

Sungmin Kang, Yavuz Faruk Bakman, Duygu Nur Yaldiz, Baturalp Buyukates, and Salman Avestimehr. 2026. Uncertainty quantification for hallucination detection in large language models: Foundations, methodology, and future directions. *IEEE BITS the Information Theory Magazine*.

Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pages 197–206. IEEE.

Piotr Komorowski, Elena Golimblevskaia, Reduan Achtibat, Thomas Wiegand, Sebastian Lapuschkin, and Wojciech Samek. 2025. Attribution-guided decoding. *arXiv preprint arXiv:2509.26307*.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Wonbin Kweon, Sanghwan Jang, SeongKu Kang, and Hwanjo Yu. 2025. Uncertainty quantification and decomposition for llm-based recommendation. In *Proceedings of the ACM on Web Conference 2025*, pages 4889–4901.

Yawei Li, David Rügamer, Bernd Bischl, and Mina Rezaei. 2025. Calibrating llms with information-theoretic evidential deep learning. *arXiv preprint arXiv:2502.06351*.

Zhen Lin, Shubhendu Trivedi, and Jimeng Sun. 2023. Generating with confidence: Uncertainty quantification for black-box large language models. *arXiv preprint arXiv:2305.19187*.

Chen Ling, Xujiang Zhao, Xuchao Zhang, Wei Cheng, Yanchi Liu, Yiyou Sun, Mika Oishi, Takao Osaki, Katsushi Matsuda, Jie Ji, and 1 others. 2024. Uncertainty quantification for in-context learning of large language models. 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 3357–3370.

Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Chris Leung, Jiajie Tang, and Jiebo Luo. 2024. Llm-rec: Personalized recommendation via prompting large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 583–612.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, and 1 others. 2023. Controlled decoding from language models. *arXiv preprint arXiv:2310.17022*.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, volume 2024, pages 25055–25083.

Pranab Kumar Sen. 1968. Estimates of the regression coefficient based on kendall's tau. *Journal of the American statistical association*, 63(324):1379–1389.

Linwei Tao, Yi-Fan Yeh, Minjing Dong, Tao Huang, Philip Torr, and Chang Xu. 2025. Revisiting uncertainty estimation and calibration of large language models. *arXiv preprint arXiv:2505.23854*.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, and 1 others. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher D Manning. 2023. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5433–5442.

Hajime Uno, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and Lee-Jen Wei. 2011. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, 30(10):1105–1117.

Artem Vazhentsev, Lyudmila Rvanova, Ivan Lazichny, Alexander Panchenko, Maxim Panov, Timothy Baldwin, and Artem Shelmanov. 2025. Token-level density-based uncertainty quantification methods for eliciting truthfulness of large language models. 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 2246–2262.

Lei Wang and Ee-Peng Lim. 2023. Zero-shot next-item recommendation using large pretrained language models. *arXiv preprint arXiv:2304.03153*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, and 1 others. 2024. A survey on large language models for recommendation. *World Wide Web*, 27(5):60.

Yuxin Xiao, Paul Pu Liang, Umang Bhatt, Willie Neiswanger, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2022. Uncertainty quantification with pre-trained language models: A large-scale empirical analysis. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7273–7284.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2024. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. In *International Conference on Learning Representations*, volume 2024, pages 23650–23678.

Zhenrui Yue, Sara Rabhi, Gabriel de Souza Pereira Moreira, Dong Wang, and Even Oldridge. 2023. Llamarec: Two-stage recommendation using large

language models for ranking. *arXiv preprint arXiv:2311.02089*.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*, pages 12697–12706. Pmlr.

A Experimental Details

A.1 Dataset and Candidate Construction

We evaluate KNIGHT under a candidate-conditioned sequential recommendation protocol. For ML-1M, we use the processed 20-core split from the public benchmark. For Steam and Amazon Grocery, we preprocess raw interactions with iterative k-core filtering: Steam uses 20-core filtering and Grocery uses 10-core filtering. Amazon Grocery is substantially more long-tailed than the other datasets; under strict alternating user/item filtering, removing low-frequency items also makes some users fall below the threshold, which in turn removes more items. This cascading shrinkage leads to a smaller but cleaner strict 10-core Grocery split. We therefore report the actual post-filtering statistics in Table 3 and use the same fixed-cache protocol for all methods.

For each user, interactions are sorted chronologically. The last item is held out for test, the penultimate item for validation, and the remaining sequence is used as training history. The maximum history length is 20 for ML-1M and Steam and 10 for Grocery. Each evaluation instance contains 20 candidates, consisting of the held-out item and 19 negatives sampled from items the user has never interacted with. We pre-construct fixed validation and test candidate caches, including users, histories, targets, negative candidates, base candidate orders, and key-attended candidate orders. Thus, all uncertainty methods are evaluated on exactly the same candidate sets and final base rankings.

A.2 Prompting and Candidate-Index Scoring

We use a candidate-index prompting protocol. For each user, we use five base prompt variants produced by fixed candidate-order shuffling. Key-attended prompts use the same format but insert an additional recommendation criterion, such as genre, popularity, novelty, recency, brand, developer, price, or similarity. These key-attended prompts are used only to construct diagnostic ranking views for uncertainty estimation. All validation and test prompts are generated from fixed candidate caches.

Candidate-index prompt template

User history, in chronological order:
- <history item 1>
- <history item 2>
...
- <history item L>

The 20 candidate items are:
Candidate A: <candidate title>
Candidate B: <candidate title>
...
Candidate T: <candidate title>

[Optional key-attended line]
Additional recommendation criterion: <focus on genre / popularity / novelty / similarity / ...>

Which <domain item> would the user most likely want to <action> next?
Only output one candidate index from this set: A, B, . . ., T.
Do not output any title, reason, punctuation, or extra words.
Answer:

A.3 CD Probe Training

The CD probe is trained only from the training interactions. For each user, we construct one contrastive training instance: the last item in the training sequence is used as the positive item, the preceding items form the user state, and five non-interacted items are sampled as negatives. This probe-training construction does not use validation or test held-out items.

To capture item semantics without relying on domain-specific collaborative filtering IDs, we employ a frozen all-MiniLM-L6-v2 sentence-transformer to encode raw item titles and user chronological histories into 384-dimensional dense representations. The CD probe is a lightweight MLP that concatenates the user-state vector and the candidate-item vector and maps them to a scalar compatibility score. Table 4 summarizes the architecture and training hyperparameters.

A.4 Inference and UQ Hyperparameters

Unless otherwise specified, we use five base prompt variants, five recommendation keys, one prompt per key, and CD strength $\lambda = 3$. The CD strength is selected on validation users, as detailed in Section 4.5. Probe scores are normalized within each candidate set before being added to calibrated key-attended logits. For latent topology, item-title embeddings

Table 3: **Dataset statistics after preprocessing.** #Interactions denotes the total number of interactions after filtering and before leave-one-out evaluation split. Eval Users denotes the number of users in the fixed test cache.

Dataset	Core	#Users	#Items	#Interactions	Hist. Len.	Eval Users
ML-1M	20-core	6040	3883	1000207	20	6040
Grocery	10-core	16450	8036	278,243	10	16450
Steam	20-core	38503	6267	1,722,038	20	38503

Table 4: Architectural details and training hyperparameters of the CD probe.

Architecture & feature extraction	
Text encoder	`all-MiniLM-L6-v2`
Text embedding dimension	384
Probe input dimension	768
MLP hidden layers	$768 \to 1024 \to 256 \to 1$
Activation / dropout	ReLU / 0.1
Probe output	Scalar score
Training hyperparameters	
Loss function	InfoNCE loss
Temperature τ	0.1
Negative samples K	5
Batch size	256
Optimizer	AdamW
Learning rate	1×10^{-4}
Epochs	10

Table 5: Inference hyperparameters for `KNIGHT`.

Hyperparameter	Value
Candidate size M	20
Base prompt variants P	5
Number of keys	5
CD prompts per key	1
Total CD views/user	5
CD strength λ	3
LLM batch size	4/2/4 for Llama/Gemma/Mistral
Encoder batch size	512
Sentence encoder	all-MiniLM-L6-v2
Item embedding cache	enabled
Seed	42

are cached and normalized before computing average pairwise cosine distances among view-level Top-1 items. Table 5 summarizes the inference hyperparameters.

A.5 Computational Cost

Table 6 summarizes the approximate inference cost of `KNIGHT` relative to the main uncertainty baselines. Pred. Unc. uses the five base prompt variants only, while No-CD and `KNIGHT` add five key-conditioned diagnostic views. The main additional cost of `KNIGHT` is the lightweight CD probe and the sentence encoder used for item-title embeddings; it does not require rationale generation, NLI scoring, CrossEncoder reranking, or graph eigensolvers in the final inference path.

Table 6: **Approximate inference cost.** Costs are reported relative to the five-base-view Pred. Unc. protocol.

Method	LLM forwards/user	Extra model	Relative cost
Pred. Unc.	5	none	1.0×
No-CD	10	none	∼2.0×
`KNIGHT`	10	CD probe + sentence encoder	∼2.0× + light encoder
Semantic Unc.	multiple generations	clustering/NLI	higher

Table 7: **Robustness of shared fusion weights.** Different rows vary the validation pool used to select fusion weights. All results are evaluated on held-out test users.

Fusion granularity	C@5	C@20	τ@5	τ@20
Cell-specific	0.6388	0.6167	0.2373	0.2196
Model-level	0.6385	0.6170	0.2369	0.2200
Universal	0.6382	0.6163	0.2363	0.2188
Dataset-level	0.6373	0.6153	0.2349	0.2169

B Additional Experimental Results

B.1 Cross-Model Robustness

We evaluate whether fusion weights selected on one LLM backbone transfer to other backbones. Figure 5 shows the macro model-to-model transfer results. All source-to-target transfers remain close to the target cell-specific fusion, with ΔC@20 between -0.0012 and $+0.0004$. Thus, model-level weights selected from one backbone can be applied to another backbone with negligible degradation.

Table 7 further compares different levels of weight sharing. Model-level fusion nearly matches cell-specific tuning, while universal fusion also remains competitive. This indicates that the learned fusion rule is not over-specialized to a particular model–dataset cell. The selected shared weights remain topology-dominant, supporting U_{topo} as a stable reliability signal across backbones and domains.

B.2 Qualitative Analysis

We visualize the latent topology of view-level Top-1 choices from Gemma-2-9B on ML-1M. For each user, the final ranking is fixed by calibrated base probabilities, while five base views and five CD-conditioned views are used only for uncertainty

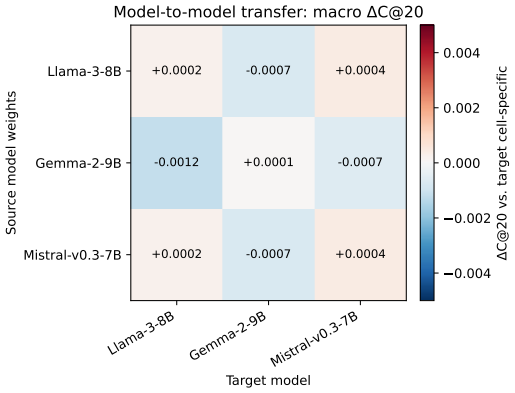


Figure 5: **Model-to-model transfer of fusion weights.** Rows denote the source LLM used to select model-level fusion weights on validation users, and columns denote the target LLM evaluated on test users. Each cell reports ΔC@20 relative to the target cell-specific fusion, averaged over datasets. All values are close to zero, indicating that fusion weights transfer robustly across LLM backbones.

estimation. We embed the unique Top-1 titles selected by these views and project them with PCA.

Figure 6 shows three representative cases. In the stable case, all base and CD views select *Armageddon*, which is also the final Top-1 and the ground-truth item; the topology collapses to a single point with $U_{\text{topo}} = 0$ and $NDCG@20 = 1.000$. In the unstable case, views scatter across semantically diverse movies, leading to high $U_{\text{topo}} = 0.816$ and low $NDCG@20 = 0.250$. The hidden-risk case is especially informative: all base views agree on *Kalifornia*, but CD-conditioned views split into several distant semantic regions, while the ground-truth item is *Heartbreak Ridge*. This exposes instability that base-view agreement alone would miss. Overall, these examples show that U_{topo} measures semantic coherence of controlled views, not just rank variance or Top-1 switching.

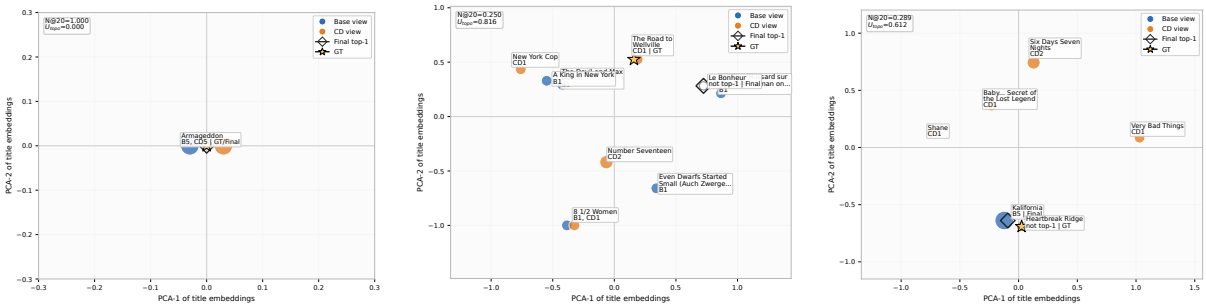
B.3 Supervision-matched Probe Controls

Because KNIGHT trains a lightweight CD probe, we evaluate whether the trained probe alone explains the observed reliability gains. We construct probe-only uncertainty scores directly from candidate-level probe values $Q_\theta(h_u, i)$, including probe entropy, top-1 confidence, and top-1 margin. These scores use the same trained probe as KNIGHT, but do not generate controlled-decoding ranking views. We further evaluate a stronger Probe+Prob baseline that fuses the probe-only scores with U_{prob} using the same validation-fitted model-level protocol.

Table 8: **Supervision-matched probe-only controls.** Probe-only uses uncertainty scores derived directly from the trained CD probe, without controlled-decoding views. Probe+Prob additionally fuses probe-only scores with U_{prob} using the same validation-fitted model-level protocol. KNIGHT uses the full CD-induced view signals. All results are **macro-averaged** over the nine blocks.

Method	C@5	C@20	τ@5	τ@20
Probe-only	0.5562	0.5471	0.0956	0.0889
Probe+Prob	0.6119	0.5930	0.1912	0.1750
KNIGHT	**0.6385**	**0.6170**	**0.2369**	**0.2200**

Table 8 shows that Probe-only is substantially weaker than KNIGHT. Adding U_{prob} produces a stronger supervision-matched baseline, but Probe+Prob still trails KNIGHT by 0.0240 Macro C@20 and 0.0450 Macro τ@20. Thus, the improvement of KNIGHT is not merely due to supervised probe training. The trained probe becomes most useful when used through controlled decoding to induce diagnostic ranking views, from which U_{pair} and U_{topo} capture rank and semantic-mode instability.



(a) Stable (b) Unstable (c) Hidden Risk

Figure 6: **CD views reveal latent semantic topology.** Base and CD-conditioned views are shown with different colors. The stable case (a) collapses to a single semantic point, while the unstable (b) and hidden-risk (c) cases show dispersed CD-induced semantic modes. GT and final Top-1 items are explicitly marked.