Multimodal Item Scoring for Natural Language Recommendation via Gaussian Process Regression with LLM Relevance Judgments

Anonymous ACL submission

Abstract

Natural Language Recommendation (NLRec) generates item suggestions based on the relevance between user-issued NL requests and NL item description passages. Existing NLRec approaches often use Dense Retrieval (DR) to compute item relevance scores from aggregation of inner products between user request embeddings and relevant passage embeddings. However, DR views the request as the sole relevance label, thus leading to a unimodal scoring function centered on the query embedding that is often a weak proxy for query relevance. To better capture the potential multimodal distribution of the relevance scoring function that may arise from complex NLRec data, we propose **GPR-LLM** that uses Gaussian Process Regression (GPR) with LLM relevance judgments for a subset of candidate passages. Experiments on four NLRec datasets and two LLM backbones demonstrate that GPR-LLM with an RBF kernel, capable of modeling multimodal relevance scoring functions, consistently outperforms simpler unimodal kernels (dot product, cosine similarity), as well as baseline methods including DR, cross-encoder, and pointwise LLM-based relevance scoring by up to 65%. Overall, GPR-LLM provides an efficient and effective approach to NLRec within a minimal LLM labeling budget.

1 Introduction

Natural Language Recommendation (NLRec) (Kang et al., 2017) aimsGuassianto generate item suggestions based on user-issued free-form textual NL requests. Unlike traditional recommender systems that rely on historical interaction data, NLRec assumes the request itself encodes the user's preferences and intent (Kang et al., 2017; Bogers and Koolen, 2017). Each item is typically associated with multiple descriptive passages such as summaries, reviews, or menus. To score item relevance, existing approaches commonly use Dense Retrieval

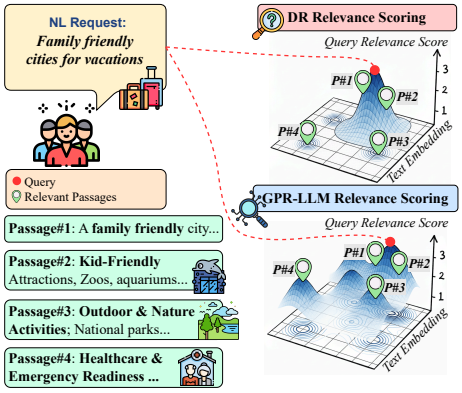

Figure 1: **(Top)** Dense retrieval (DR) methods assume a unimodal relevance scoring function concentrated near the user's NL request within the embedding space. **(Bottom)** In practice, the true relevance function is often multimodal, with relevant passages dispersed across distinct regions. The relevance scoring function for NL request "*Family friendly cities for vacations*" is multimodal as it covers multiple theme passages, including "*Kid-friendly Attractions*," "*Outdoor and Nature Activities*," and "*Healthcare and Emergency Readiness*."

(DR; Karpukhin et al., 2020), which first estimates passage relevance scores based on cosine similarity or inner product between NL request and passage embeddings in a dense embedding space, and then aggregates these passage-level scores into item-level scores. The standard DR approach implicitly assumes that the relevance scoring function is unimodal with a peak centered around the NL request (Figure 1, top). However, in practice, relevance is often complex and multimodal: the NL request alone is insufficient as the sole positive relevance label. Passages may require complex reasoning to determine relevance and may thus be distributed across multiple distant regions (modes) within the embedding space (Figure 1, bottom). As a result, the standard DR relevance scoring function is a weak proxy for passage-level relevance, and hence aggregate item relevance.

Large Language Models (LLMs) have recently

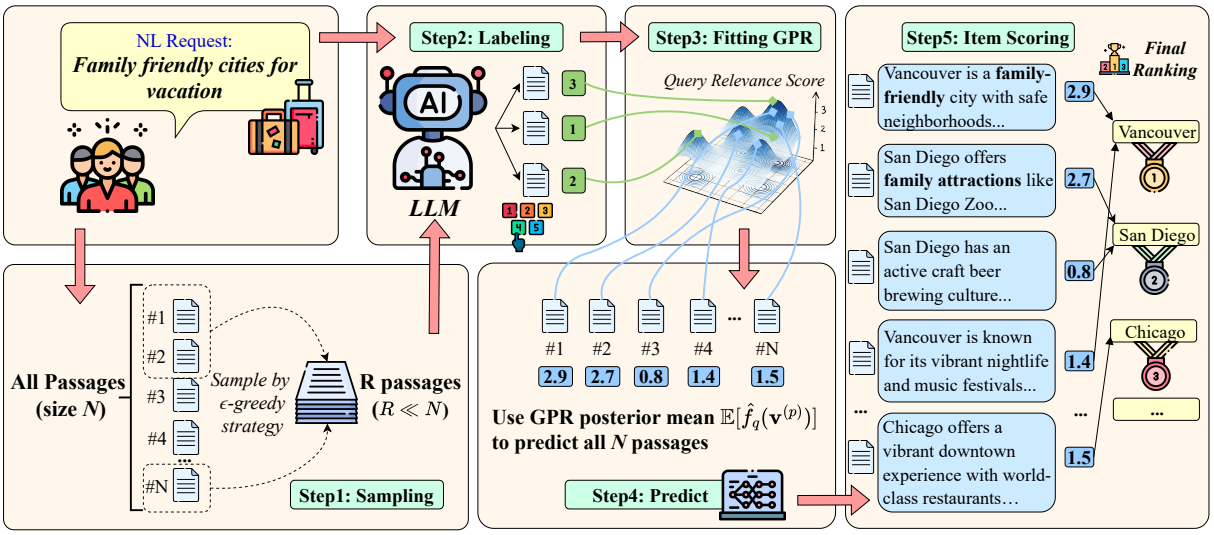

Figure 2: Overview of GPR-LLM. A small subset of $R \ll N$ passages is sampled from the full set of N passages using an ϵ-greedy sampling strategy and labeled using an LLM. A query-specific GPR model is then trained on this labeled subset to estimate relevance scores for *all* passages. Finally, item-level relevance scores are aggregated from the passage-level scores using Equation 3, and items are ranked accordingly to generate the recommendation list.

emerged as a promising resource for reasoning about relevance between items or passages and NL queries, which offer more reliable relevance scoring than can be achieved with DR (Sachan et al., 2022; Qin et al., 2024; Sun et al., 2023; Ma et al., 2023; Zhuang et al., 2024; Shen et al., 2023). However, exhaustively prompting LLMs for relevance judgments of all passages associated with an item is expensive, especially with large item collections and numerous NL description passages.

In order to improve the reliability of the dense scoring function for NLRec while working within a minimal budget of LLM labeling calls, we propose **GPR-LLM** that uses Gaussian Process Regression (GPR) (Rasmussen and Williams, 2006; Gramacy, 2020) with LLM relevance judgements for a sampled subset of candidate passages.

A key insight of GPR-LLM is that the similarity or distance functions from DR can serve directly as *kernel functions* with GPR. In the special case where the kernel function is identical to the similarity function used in DR, and assuming that the NL request itself is the only relevant label, GPR exactly reduces to standard DR. However, GPR can also critically leverage *additional relevance labels* obtained from LLM passage relevance judgments to estimate the multimodal relevance scoring function (cf. Figure 1, bottom) when used with an RBF kernel over dense embeddings. As we show empirically, GPR-LLM is an efficient and effective relevance scoring approach for NLRec that outperforms state-of-the-art methods under a minimal LLM labeling budget. In summary, we make the following key contributions:

- We propose GPR-LLM (cf. Figure 2) to estimate the multimodal relevance scoring function in NLRec using GPR with LLM relevance judgements to address the limitations of the unimodal relevance scoring function in DR.
- We empirically show that the RBF kernel, which is designed to capture the multimodal relevance scoring function, consistently outperforms unimodal dot product and cosine similarity kernels across four benchmarks and two LLM backbones.
- GPR-LLM consistently outperforms all baselines, including BM25, DR, Cross-encoder, and LLM-based Relevance Scoring under the same LLM labeling budget, and achieves comparable performance to the best baselines even with a significantly smaller budget.

2 Problem Definition

We consider the problem of ranking a set of items in response to a user-issued NL request, which we simply refer to as a query $q \in \mathcal{Q}$. Let $\mathcal{I}$ represent a collection of total M items. Each item $i \in \mathcal{I}$ is associated with a set of textual passages. We denote the complete collection of N passages as $\mathcal{P}$, and their corresponding text embeddings as $\mathcal{V}$, where

$\mathbf{v}^{(p_j)} \in \mathbb{R}^D$ represents the embedding of passage $p_j \in \mathcal{P}$. Each passage $p_j \in \mathcal{P}$ is uniquely linked to item i. We define the set of item i's passages as

$$\mathcal{P}^{(i)} = \{p_j \in \mathcal{P} \mid p_j \text{ is associated with item } i\}. \tag{1}$$

The objective is to recommend the item $i \in \mathcal{I}$ that is most relevant to q based on its associated passages $\mathcal{P}^{(i)}$. Thus, we decompose the task into two subproblems: passage-level relevance estimation and item-level relevance aggregation.

Passage-Level Relevance Let $f_q^* : \mathcal{P} \to \mathbb{R}$ be a query-specific passage relevance scoring function that assigns a ground truth real-valued relevance score to passage $p_j \in \mathcal{P}$ with respect to query q. let $\mathcal{S}^* \in \mathbb{R}^{M \times N}$ be the ground truth relevance score of the j-th passage of item $i$as:

$$\mathcal{S}^*_{i,j} = f_q^*(p_j) \quad \forall p_j \in \mathcal{P}^{(i)}, i \in \mathcal{I} \tag{2}$$

However, in real-world applications, $f_q^*(p_j)$ is often unknown; thus, we employ $\hat{f}_q : \mathcal{P} \to \mathbb{R}$ to estimate $\mathcal{S}^*$ as $\mathcal{S}$ where $\mathcal{S}_{i,j} = \hat{f}_q(p_j) \forall p_j \in \mathcal{P}^{(i)}$.

Item-Level Relevance Given the estimated passage-level relevance score $\mathcal{S}_{:,:}$, item-level scores can be derived by aggregating $\mathcal{S}_{:,:}$ by item. Specifically, for each item i, we first select the top-T passages from $\mathcal{P}^{(i)}$ with highest $\mathcal{S}_{i,:}$. We denote this list as $\mathcal{P}^{(i)}_{\text{top-T}}$. The item-level relevance score $\mathcal{S}_i$ for an item i is computed as follows:

$$\mathcal{S}_i = \phi\left(\left[\mathcal{S}_{i,j} \mid p_j \in \mathcal{P}^{(i)}_{\text{top-T}}\right]\right) \quad \forall i \in \mathcal{I}, \tag{3}$$

where $\phi : \mathbb{R}^T \to \mathbb{R}$ is an aggregation function (e.g., mean or max). The top-K NLRec items are ranked by $\mathcal{S}_i$ in descending order.

The effectiveness of NLRec relies heavily on the quality of estimated passage-level scores. Thus, we focus on the formulation of $\hat{f}_q$, aiming to closely approximate the true relevance function f_q^* by effectively capturing the multimodal relevance scoring function, thereby enabling better item-level relevance scoring estimation for NLRec.

3 GPR-LLM: Gaussian Process Regression with LLM Relevance Judgments

To capture the multimodal structure of $\mathcal{S}^*$ (cf. Figure 1, bottom) and ensure generalizability across different applications, we aim to approximate the true (possibly multimodal) relevance function f_q^* with a query-specific GPR scorer $\hat{f}_q : \mathcal{V} \to \mathbb{R}$ that leverages both (i) LLM relevance judgments on a small, carefully sampled subset of passages and (ii) the geometry of NL embedding spaces (as in DR). At a high level:

1. *Candidate Sampling:* from all N passages, construct a small labeled set $\mathcal{P}^{\text{GP}}$ of size $R \ll N$ via an ϵ-greedy strategy that mixes top DR-ranked items and uniform random exploration (cf. Subsection 3.4).

2. *LLM Relevance Judgments:* Obtain $\mathcal{S}^{\text{LLM}}_{\cdot,j}$ from an LLM backbone for each sampled passage $p_j \in \mathcal{P}^{\text{GP}}$ (cf. Subsection 3.3).

3. *Fitting GPR:* fit a query-specific GP prior/posterior over $\hat{f}_q$ with a kernel defined in the embedding space (cf. Subsection 3.1 and Subsection 3.2).

4. *Scoring all passages:* use the GPR posterior mean $\mathbb{E}[\hat{f}_q(\mathbf{v}^{(p)})]$ to produce passage-level scores for all $p \in \mathcal{P}$, followed by standard item-level aggregation.

3.1 Query-specific GPR.

Gaussian Process Regression (GPR) (Williams and Rasmussen, 1996) is a non-parametric, kernel-based regression model that has been widely adopted for estimating multimodal functions (Lawrence and Urtasun, 2009).

GPR places a GP prior over $\hat{f}_q$:

$$\hat{f}_q(\mathcal{V}) \sim \mathcal{GP}(0, k(\mathcal{V}, \mathcal{V}')), \tag{4}$$

where $k(x, x')$ is a covariance function (kernel).

Given R passages with observed scores $\mathcal{S}_{\cdot,j}$ and the query with max possible score, we collect scores and embeddings as

$$\mathcal{D} = \{(\mathbf{v}^{(q)}, \mathcal{S}_{\max})\} \cup \{(\mathbf{v}^{(p_j)}, \mathcal{S}_{\cdot,j})\}_{j=1}^{R}. \tag{5}$$

For a passage p_* with embedding $\mathbf{v}^{(p_*)}$, the noisy-observation posterior is $p(\hat{f}_q(\mathbf{v}^{(p_*)})|\mathcal{D}) \sim \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Sigma}_*)$ with

$$\boldsymbol{\mu}_* = \mathbf{k}_*^\top (\mathbf{K} + \alpha \mathbf{I})^{-1} \{\mathcal{S}_{\cdot,j}\}_{j=1}^{R}, \tag{6}$$

$$\boldsymbol{\Sigma}_* = k(\mathbf{v}^{(p_*)}, \mathbf{v}^{(p_*)}) - \mathbf{k}_*^\top (\mathbf{K} + \alpha \mathbf{I})^{-1} \mathbf{k}_*, \tag{7}$$

where

- $\mathbf{K} \in \mathbb{R}^{R \times R} = k(\{\mathbf{v}^{(p_j)}\}_{j=1}^{R}, \{\mathbf{v}^{(p_j)}\}_{j=1}^{R})$ is the kernel matrix of the labeled passages,

- $\mathbf{k}_* \in \mathbb{R}^R = k(\{\mathbf{v}^{(p_*)}, \{\mathbf{v}^{(p_j)}\}_{j=1}^R)$ is the cross-kernel vector, and
- α is the variance of the Gaussian noise (label noise) of observed $\{\mathcal{S}_{\cdot,j}\}_{j=1}^R$ since the true relevance is usually unknown.

The predictive mean $\mathbb{E}[\hat{f}_q(\mathbf{v}^{(p_*)})]$ is used as the estimated passage relevance, which we later aggregate to the item level.

3.2 Kernels for GPR

We can conveniently repurpose the distance or similarity function used in DR as a kernel function for GPR to leverage the intuition of DR that semantically similar NL texts are likely to be located close together in the embedding space. We propose the following dense kernels:

- **Dot Product (Linear)**: $k(\mathcal{V}, \mathcal{V}') = \mathcal{V}^\top \mathcal{V}'$
- **Cosine Similarity**: $k(\mathcal{V}, \mathcal{V}') = \frac{\mathcal{V}^\top \mathcal{V}'}{\|\mathcal{V}\| \cdot \|\mathcal{V}'\|}$
- **Radial Basis Function (RBF)**: $k(\mathcal{V}, \mathcal{V}') = \exp\left(-\frac{\|\mathcal{V}-\mathcal{V}'\|^2}{2\ell^2}\right)$

Both the Dot Product and Cosine Similarity kernels are consistent with the relevance measures used in DR. They are *non-stationary* kernels capable of modeling only unimodal distributions (Williams and Rasmussen, 2006). In contrast, RBF is a *stationary* kernel that measures similarity by proximity in the embedding space and capture local variations across regions. Due to this locality (Steinwart, 2001; Williams and Rasmussen, 2006), it is effective for capturing the potentially multimodal relevance function (cf. Figure 1, bottom).

3.3 LLM Relevance Judgments for GPR

Another crucial consideration for GPR-LLM is the source of relevance judgements $\mathcal{S}$ in $\mathcal{D}$. LLMs are capable of modeling complex relationships between queries and passages that require contextual reasoning, inference, or multi-step understanding (Sun et al., 2023; Qin et al., 2024). This makes LLM relevance judgments a natural choice to estimate S^*.

We use the commonly adopted UMBRELA `prompt` (Upadhyay et al., 2024) and follow the procedure of Zhuang et al. (2024) to obtain the LLM relevance judgment between a query q and a passage p_j as follows:

$$\mathbf{z} = \mathrm{LLM}(q, p_j, \texttt{prompt}) \tag{8}$$

where $\mathbf{z} = [z_0, z_1, \ldots, z_{K-1}]$ corresponds to the logit of a predefined discrete relevance label $r_k \in \{0, 1, \ldots, K-1\}$. We then compute the LLM-based relevance score using the *expected relevance*:

$$\mathcal{S}_{i,j}^{\mathrm{LLM}} = \sum_{k=0}^{K-1} \left(\frac{e^{z_k}}{\sum_{j=0}^{K-1} e^{z_j}} \right) \cdot r_k. \tag{9}$$

3.4 Sampling for GPR

Since obtaining LLM relevance judgments of the entire N passages is expensive, we instead sample a subset of passages (i.e., $R \ll N$), denoted as $\mathcal{P}^{\mathrm{GP}}$. In an ideal situation, $\mathcal{P}^{\mathrm{GP}}$ should include all relevant passages. DR offers an efficient mechanism for identifying potentially relevant passages. However, its underlying unimodal assumption often causes relevant passages far from the query in the embedding space to be overlooked.

To mitigate this limitation, we propose the ϵ-*Greedy Sampling* method, which leverages DR efficiency while reducing its inherent bias. Specifically, we define the sampling set as:

$$\mathcal{P}^{\mathrm{GP}} = \mathcal{P}^{\mathrm{DenseRanking}} \cup \mathcal{P}^{\mathrm{ExploratorySampling}},$$

where $\mathcal{P}^{\mathrm{DenseRanking}}$ consists of the top-$\lfloor(1-\epsilon)R\rfloor$ passages ranked by DR. The remaining $\lceil \epsilon R \rceil$ passages are sampled uniformly at random from the top-η DR-ranked passages not already selected, where $\eta \gg \lfloor(1-\epsilon)R\rfloor$. Formally,

$$\mathcal{P}^{\mathrm{ExploratorySampling}} \subseteq \mathcal{P}^{\eta} \setminus \mathcal{P}^{\mathrm{DenseRanking}},$$

with $\mathcal{P}^{\eta}$ denoting a set of top-η DR-ranked passages. The method thus involves two parameters:

$\epsilon \in [0, 1]$ controls the exploration–exploitation trade-off. Setting $\epsilon = 0$ corresponds to *greedy sampling*, relying on DR rankings, whereas $\epsilon = 1$ selects passages uniformly at random.

η constraints exploratory sampling to the top-η ranked passages, preventing the inclusion of irrelevant passages ranked too low by DR.

$\mathcal{D}$ combines the LLM relevance judgments $\mathcal{S}_{\cdot,j}^{\mathrm{LLM}}$ and the dense embedding $\mathbf{v}^{(p_j)}$ of all sampled $p_j \in \mathcal{P}^{\mathrm{GP}}$ using ϵ-Greedy Sampling. We then estimates the passage-level scores for all $\mathcal{P}$ given $\mathcal{D}$ (cf. Equation 6) and aggregates for item recommendation (cf. Equation 3). Figure 2 illustrates the overall GPR-LLM pipeline.

Method	Per-query Complexity	Latency (sec)
DR	$\mathcal{O}(ND)$	0.165 (0.161, 0.168)
	LLM-based Scoring	
Point.	$\mathcal{O}(ND + R\,C_{\text{LLM}})$	0.678 (0.671, 0.685)
	GPR–LLM	
Dot	$\mathcal{O}(ND + NRD + R\,C_{\text{LLM}} + R^2D + R^3)$	0.782 (0.769, 0.795)
Cosine		0.774 (0.762, 0.787)
RBF		0.754 (0.730, 0.780)

Table 1: Per-query time complexity and latency (seconds) under R=50, N=100,000, and D=368. Latencies show 95% CIs in [·]. All computations were performed on an NVIDIA GeForce RTX 4070 GPU.

3.5 Complexity Analysis

Per query, candidate sampling via ϵ-greedy selection costs $\mathcal{O}(ND)$, corresponding to a dense-retrieval pass over N passages with D-dimensional embeddings and negligible additional sampling overhead. LLM relevance judgments for R passages incur a cost of $\mathcal{O}(R\,C_{\text{LLM}})$, where C_{LLM} denotes the per-call inference cost. GPR fitting then includes kernel construction of $\mathcal{O}(R^2D)$ and an exact Cholesky decomposition of $\mathcal{O}(R^3)$. Finally, scoring all N passages requires $\mathcal{O}(NRD)$ to compute the cross-kernel $K_{N\times R}$ and posterior mean. The total per-query runtime is therefore:

$$\mathcal{O}\big(ND + R\,C_{\text{LLM}} + R^2D + R^3 + NRD\big).$$

Since $R \ll N$, the dominant costs are typically the DR pass and GP inference over all passages, while GPR fitting remains inexpensive. In practice, the computational overhead of GPR-LLM relative to pointwise LLM-based relevance scoring with the same R is minor (cf. Table 1).

4 Experiments

4.1 Experimental Setup

We compare GPR-LLM against the following baseline relevance scoring methods[1][2]:

- BM25 (Robertson et al., 1994): Computes relevance scores using BM25.
- Dense Retrieval (DR) (Karpukhin et al., 2020): Computes relevance scores using inner products between query and passage embeddings.
- Cross-Encoder (CE) (Nogueira et al., 2019): Computes relevance scores by jointly encoding query-passage pairs.

[1]Code for reproducing our experiments is available at `https://anonymous.4open.science/r/Bandit_Retrieval-0352/README.md`.

[2]See Appendix B for detailed baseline implementations.

Dataset	# Queries	# Items	# Passages	# Qrel
TravelDest	50	775	126,400	3,721
POINTREC	28	59,553	592,818	498
TripAdvisor Hotel	100	589	133,759	4,887
Yelp Restaurant	100	1,152	283,658	11,726

Table 2: Statistics of our benchmark datasets. Qrels indicates the total number of relevant items summed over all queries.

- Pointwise LLM-based Relevance Scoring (Point.) (Zhuang et al., 2024): Computes relevance scores by prompting an LLM with query–passage pairs and computing the expected relevance score from the model's predicted label distribution.

We conduct experiments on four publicly available benchmark datasets:

- `POINTREC` (Afzali et al., 2021): Point-of-interest recommendation using reviews and structured descriptions.
- `TravelDest` (Wen et al., 2024): Travel city recommendation using city-level descriptions.
- `TripAdvisor Hotel` (Wen et al., 2025): Hotel recommendation using user reviews.
- `Yelp Restaurant` (Wen et al., 2025): Restaurant recommendation using user reviews.

These datasets allow us to assess the generalizability of our approach across diverse NLRec scenarios. See Table 2 for dataset statistics.

We experiment with two LLM backbone models:

- GPT-4o (OpenAI, 2023)
- Qwen3-Next-80B-A3B as an open-source model (Yang et al., 2025)

We use the UMBRELA prompt (Upadhyay et al., 2024) to obtain LLM relevance judgments across various LLM budgets R. For embeddings, we use the `all-MiniLM-L6-v2` (Karpukhin et al., 2020) and the `msmarco-distilbert-base-tas-b` (Hofstätter et al., 2021; see Appendix D) We set the number of top passages to aggregate as $T = 3$ and define the aggregation function ϕ (cf. Equation 3) as the mean $\text{mean}(\cdot)$ over passage scores. The observation noise variance α is set by default to 10^{-3}. Finally, the length scale ℓ in the RBF kernel is optimized using the L-BFGS-B (Zhu et al., 1997)

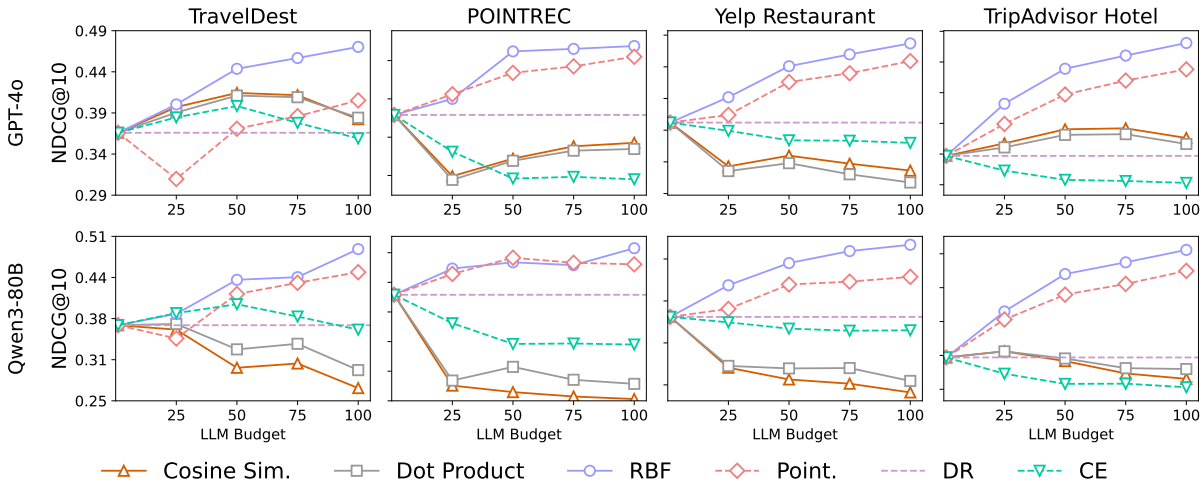

Figure 3: **RQ1:** Kernel choices comparison with Greedy Sampling ($\epsilon = 0$) under varying LLM labeling budget.

algorithm.[3] Evaluation is conducted using NDCG and Precision @10 and @30.

We address the following research questions:

RQ1 (Kernel Choice): Does the stationary RBF kernel outperform non-stationary kernels in relevance scoring?

RQ2 (Sampling Strategy): Does including a fraction ϵ of randomly sampled exploratory passages with a cap τ lead to better relevance scoring compared to only selecting top-ranked passages from DR?

RQ3 (Performance Comparison): Does GPR-LLM consistently outperform baseline methods across different LLM backbones given the same labeling budget?

RQ4 (Multimodal Relevance Scoring): Do empirical results support that GPR-LLM with the RBF kernel more effectively captures multimodal relevance compared to other methods?

4.2 Results

RQ1 (Kernel Choice) To address **RQ1**, Figure 3 compares different kernel functions (cf. Subsection 3.2) under greedy sampling ($\epsilon = 0$) with τ set to include all passages. The stationary RBF kernel consistently and significantly outperforms the non-stationary dot product and cosine similarity kernels across all datasets and LLM backbones. These results confirm the advantage of stationary kernel for GPR-LLM, likely due to its capability to capture the multimodal passage-level relevance scoring function. This is further examined in **RQ4**. Additionally, Figure 3 demonstrates that GPR-LLM with the RBF kernel outperforms all baseline methods under the same labeling budget, even without tuning sampling parameters.

RQ2 (Sampling Strategy). Figure 4 shows the effect of different sampling configurations that mix a fraction ϵ of randomly selected passages from the top-τ non-DR set with $(1 - \epsilon)$ passages drawn from the top DR rankings. Across all datasets and LLM backbones, a small exploratory fraction ($\epsilon = 0.3$) consistently improves GPR-LLM performance when the sampling range is sufficiently large ($\tau \geq 5000$), whereas no such gain is observed for smaller τ. Under most sampling configurations, GPR-LLM also outperforms the pointwise LLM-based scoring baseline (dot–dash line). These results suggest that limited exploration enables GPR-LLM to better capture the multimodal relevance distribution by incorporating a more diverse subset of samples, but only when the exploration remains balanced and the sampled passages are sufficiently distinct from the top DR-ranking.

RQ3 (Performance Comparison) Table 3 compares GPR-LLM (with the best-performing RBF kernel, $\epsilon = 0.1$, and τ including all passages) against all baseline methods across various LLM backbones and labeling budgets. GPR-LLM consistently outperforms baselines at same LLM budgets with only a few exceptions, and specifically achieves up to 65% improvements over the pointwise LLM relevance scoring. Also, GPR-LLM is

[3] Additional experiments analyzing the impact of different embeddings and hyperparameter choices are provided in Appendix D, Appendix E, and Appendix F

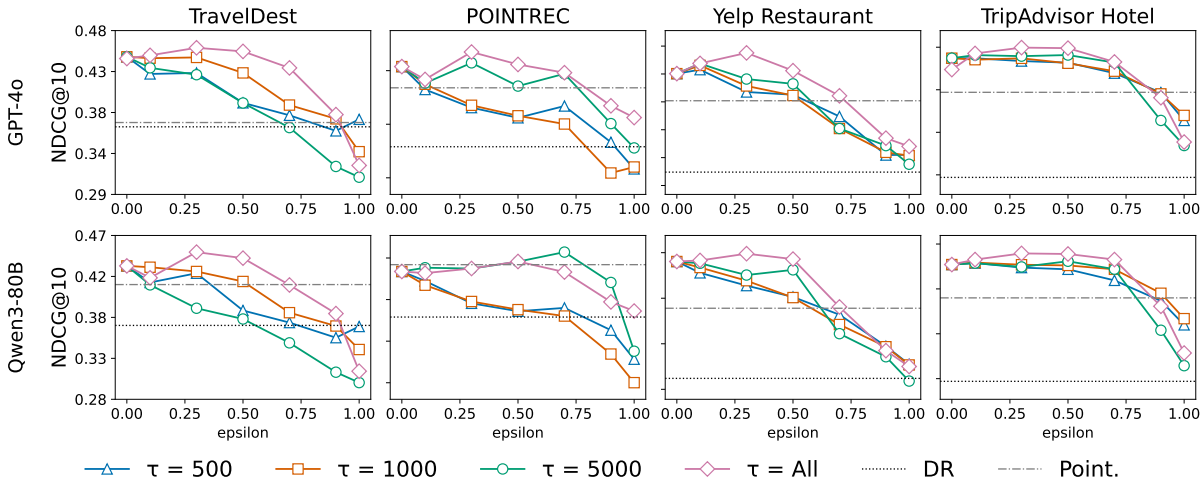

Figure 4: **RQ2:** Performance of GPR-LLM under $R = 50$ with varying sampling sets. Each configuration labels an ϵ fraction of random passages from the top-τ non-DR set and a $(1 - \epsilon)$ fraction from top DR-ranked passages.

Budget	Backbone	Method	TravelDest P@10	TravelDest N@10	TravelDest P@30	TravelDest N@30	POINTREC P@10	POINTREC N@10	POINTREC P@30	POINTREC N@30	Yelp Restaurant P@10	Yelp Restaurant N@10	Yelp Restaurant P@30	Yelp Restaurant N@30	TripAdvisor Hotel P@10	TripAdvisor Hotel N@10	TripAdvisor Hotel P@30	TripAdvisor Hotel N@30
N/A	N/A	BM25	0.234	0.238	0.237	0.239	0.025	0.032	0.025	0.038	0.309	0.327	0.236	0.283	0.205	0.257	0.153	0.325
	N/A	DR	0.360	0.366	0.314	0.332	0.164	0.179	0.104	0.182	0.346	0.362	0.282	0.331	0.231	0.297	0.166	0.365
25	N/A	CE	0.384	0.385	0.316	0.336	0.121	0.131	0.087	0.137	0.332	0.349	0.275	0.321	0.227	0.273	0.168	0.336
	GPT-4o	Point.	0.294	0.309	0.238	0.219	**0.175**	**0.206**	0.106	0.158	0.324	0.374	0.237	0.260	0.251	0.349	0.178	0.332
		GPR-LLM	**0.376***	**0.401***	**0.340***	**0.364***	0.157	0.200	**0.106**	**0.158**	**0.360**	**0.402***	**0.273**	**0.340***	**0.282***	**0.382***	**0.182**	**0.426***
	Qwen3-80B	Point.	0.308	0.345	0.286	0.309	0.175	0.214	**0.111**	**0.206**	0.336	0.381	0.266	0.331	0.258	0.353	0.165	0.398
		GPR-LLM	**0.366***	**0.384***	**0.285**	**0.316**	**0.179**	**0.223**	0.082	0.177	**0.395***	**0.437***	**0.277**	**0.353***	**0.270**	**0.365**	**0.176**	**0.418**
50	N/A	CE	0.390	0.399	0.325	0.348	0.093	0.096	0.077	0.110	0.314	0.334	0.272	0.315	0.216	0.258	0.167	0.329
	GPT-4o	Point.	0.356	0.371	0.248	0.281	0.196	0.234	0.105	0.208	0.386	0.426	0.254	0.332	0.289	0.397	0.169	0.417
		GPR-LLM	**0.432***	**0.445***	**0.362***	**0.389***	**0.236***	**0.262***	**0.113**	**0.222**	**0.408**	**0.451***	**0.304***	**0.377***	**0.324***	**0.439***	**0.197***	**0.482***
	Qwen3-80B	Point.	0.392	0.415	0.273	0.315	**0.211**	**0.242**	**0.108**	**0.213**	0.401	0.439	0.266	0.347	0.296	0.390	0.169	0.417
		GPR-LLM	**0.414***	**0.437***	**0.327***	**0.363***	0.200	0.234	0.102	0.202	**0.439***	**0.490***	**0.308***	**0.398***	**0.314**	**0.420***	**0.196***	**0.472***
100	N/A	CE	0.364	0.359	0.312	0.324	0.089	0.095	0.069	0.104	0.320	0.330	0.270	0.309	0.204	0.253	0.161	0.322
	GPT-4o	Point.	0.398	0.406	0.296	0.328	0.225	0.255	0.124	0.231	0.418	0.459	0.298	0.379	0.322	0.438	0.189	0.472
		GPR-LLM	**0.448***	**0.472***	**0.380***	**0.412***	**0.246***	**0.269**	**0.130**	**0.239**	**0.444***	**0.487***	**0.334***	**0.413***	**0.358***	**0.481***	**0.218***	**0.529***
	Qwen3-80B	Point.	0.418	0.449	0.323	0.366	0.207	0.230	**0.121**	**0.218**	0.424	0.457	0.303	0.381	0.319	0.425	0.192	0.467
		GPR-LLM	**0.472***	**0.486***	**0.362***	**0.399***	**0.225***	**0.258***	0.123	0.226	**0.491***	**0.534***	**0.355***	**0.443***	**0.345***	**0.456***	**0.220***	**0.520***

Table 3: **RQ3**: Comparison of DR, Cross-Encoder (CE), Pointwise LLM-based Relevance Scoring (Point.) and GPR-LLM (RBF kernel, $\epsilon = 0.1$, τ set to include all passages) across four datasets at varying LLM label budgets. Metrics: Precision@10 (P@10), NDCG@10 (N@10), Precision@30 (P@30), NDCG@30 (N@30). Bold values indicate the best-performing method between Point. and GPR-LLM for each backbone. Statistically significant improvements over Point. (paired t-test, $p < 0.05$) are indicated by an asterisk (*).

often able to outperform baselines that use twice as many labels. The performance is consistent across different NLRec datasets and LLM backbones, which highlights the robustness and effectiveness of our proposed method.

RQ4 (Multimodal Relevance Scoring) Next, we investigate the underlying factors contributing to these performance gains. Figure 5 visualizes the distribution of top-ranked passages for a representative query from each dataset using both DR and GPR-LLM in the reduced embedding space using t-SNE (van der Maaten and Hinton, 2008).

DR's top-ranked passages tend to cluster tightly around the query across all datasets, reflecting its unimodal relevance assumption with a single peak at the query (i.e., passages closer to the query are inherently more relevant). In contrast, GPR-LLM retrieves passages distributed across multiple distinct regions in the embedding space. This distribution aligns naturally with our multimodal relevance hypothesis illustrated in Figure 1, where each region represents a local relevance peak within the overall multimodal relevance surface. By capturing distant yet relevant passages, GPR-LLM overcomes a key limitation of DR. This multimodal retrieval pattern is consistently observed across all four benchmark datasets, providing a compelling explanation for GPR-LLM's performance advantage.

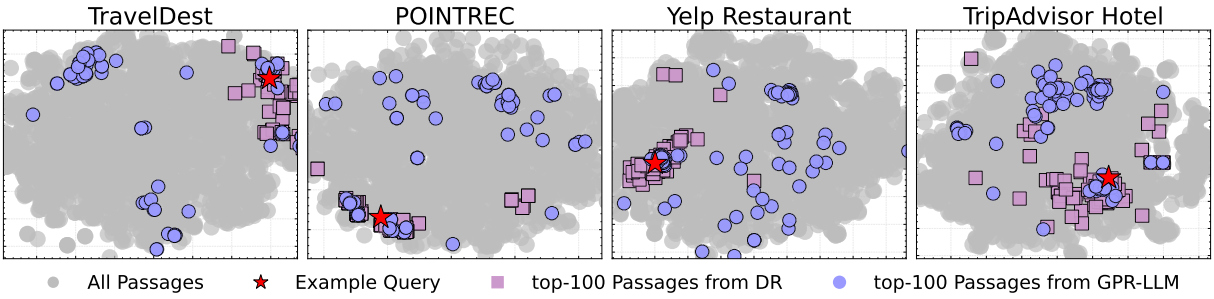

Figure 5: **RQ4:** Distribution of the top-100 passages from DR and GPR-LLM for example queries across all datasets. The embedding space is reduced to two dimensions using t-SNE for visualization. Passages ranked highest by DR (purple squares) cluster tightly around the query (red star), reflecting DR's unimodal assumption. In contrast, GPR-LLM (blue circles) identifies passages dispersed across multiple regions, highlighting its ability to capture the multimodal relevance distribution hypothesized in Figure 1.

5 Related Work

5.1 Natural Language Recommendation

Natural Language Recommendation (NLRec) aims to generate item recommendations based on user-issued free-form textual requests (Kang et al., 2017). NLRec systems typically exhibit two key properties. First, they utilize natural language requests to encode user intent and preferences, contrasting with traditional recommender systems that mainly rely on structured interaction history (e.g., clicks, ratings) as their primary source of preference signals (Bogers and Koolen, 2017; Bogers et al., 2018, 2019; Afzali et al., 2021). Second, NLRec leverages collections of textual sources, such as item descriptions, reviews, or menus, to represent items and aggregate them into item-level representations (Afzali et al., 2021; Wen et al., 2025; Zhang et al., 2023).

As NLRec tasks grow in complexity, they often exhibit *multimodal relevance*, characterized by multiple regions of high relevance scores in the dense embedding space (cf. Figure 1; Wen et al., 2025). To effectively address this multimodal relevance, we propose GPR-LLM, which aims to model the multimodal relevance scoring function using Gaussian Process Regression with a stationary kernel.

5.2 LLM Relevance Judgement

Recently, Large Language Models (LLMs) have emerged as promising resources for determining the relevance between items or passages and natural language queries due to their strong contextual understanding and reasoning capabilities (Sachan et al., 2022; Qin et al., 2024; Sun et al., 2023; Ma et al., 2023; Zhuang et al., 2024; Shen et al., 2023).

LLM-based relevance judgments generally fall into three categories: pointwise, listwise, and pairwise. Pointwise methods independently compute relevance scores for each query–passage pair (Sachan et al., 2022; Zhuang et al., 2024; Upadhyay et al., 2024). Listwise methods prompt LLMs with a query and multiple candidate passages simultaneously, often using a sliding window approach to facilitate passage comparisons and directly generate a ranked list (Sun et al., 2023; Ma et al., 2023; Shen et al., 2023). Pairwise methods prompt LLMs to compare two passages, determine their relative relevance, and aggregate these pairwise preferences into final rankings (Qin et al., 2024).

To obtain passage-level relevance scores for aggregation into item-level scores in NLRec tasks, we primarily focus on pointwise LLM-based relevance scoring, which we compare against our proposed GPR-LLM method.

6 Conclusion

We introduced GPR-LLM, a method that models multimodal relevance scoring for NLRec using Gaussian Process Regression with a small subset of LLM-based relevance judgments. GPR-LLM addresses limitations of the unimodal scoring assumption in Dense Retrieval (DR) and reduces the reliance on exhaustive LLM labeling. Experiments across multiple datasets and LLM backbones demonstrate that GPR-LLM consistently outperforms baseline methods, including DR, Cross Encoder, and Pointwise LLM Relevance Scoring at same labeling budgets, and achieves comparable performance with substantially fewer LLM labels. These results establish GPR-LLM as both an efficient and effective approach for NLRec tasks.

Limitations

While GPR-LLM achieves consistent improvement over baselines, several limitations remain. First, the performance of GPR-LLM critically depends on the quality and consistency of the LLM relevance judgments. Variations in LLM prompting or labeling criteria can impact accuracy and reliability, potentially influencing the quality of the GPR.

Second, we use an ϵ-greedy sampling strategy for exploration that depends on uniform random sampling. However, alternative sampling methods that explicitly leverage uncertainty-aware active learning or diversity-maximization exploration may align better with the exploration goal. Although such methods could further enhance performance, investigating them is beyond the scope of this paper and is thus left for future study.

Third, we aggregate item-level scores using a fixed aggregation function ϕ (mean or max) over the top-T passages. Alternative aggregation methods, such as harmonic mean or learned fusion networks, could be explored to potentially enhance performance. However, as our primary focus in this paper is on improving passage-level relevance scoring, we leave the investigation of more advanced aggregation methods for future work.

Ethical Considerations

In deploying GPR-LLM, it is important to consider potential ethical implications. The reliance on LLMs introduces risks related to bias and fairness, as LLM relevance judgments may inherit or amplify biases present in training data. Thus, careful evaluation and monitoring of the generated labels and recommendation results are necessary to mitigate potential discriminatory impacts.

References

Jafar Afzali, Aleksander Mark Drzewiecki, and Krisztian Balog. 2021. Pointrec: A test collection for narrative-driven point-of-interest recommendation. In *Proceedings of the 44th ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '21)*, pages 2478–2484.

Toine Bogers, Matthias Gäde, Marijn Koolen, Vojtěch Petras, and Mads Skov. 2018. "what was this movie about again?": combining narrative cues and example items for narrative-driven movie recommendation. In *Proceedings of the iConference 2018 (KaRS workshop)*, volume 10766 of *Lecture Notes in Computer Science*, pages 323–334.

Toine Bogers, Matthias Gäde, Marijn Koolen, Vojtěch Petras, and Mads Skov. 2019. "looking for an amazing game i can relax and sink hours into . . . ": a study of relevance aspects in video game discovery. In *Proceedings of the iConference 2019*, volume 11420 of *Lecture Notes in Computer Science*, pages 503–515.

Toine Bogers and Marijn Koolen. 2017. Defining and supporting narrative-driven recommendation. In *Proceedings of the Eleventh ACM Conference on Recommender Systems (RecSys '17)*, pages 238–242, New York, NY, USA. Association for Computing Machinery.

Robert B. Gramacy. 2020. *Surrogates: Gaussian Process Modeling, Design, and Optimization for the Applied Sciences*. Chapman & Hall/CRC.

Sebastian Hofstätter, Sheng-Chieh Lin, Jheng-Hong Yang, Jimmy Lin, and Allan Hanbury. 2021. Efficiently teaching an effective dense retriever with balanced topic aware sampling. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 113–122. ACM.

Jie Kang, Kyle Condiff, Shuo Chang, Joseph A. Konstan, Loren Terveen, and F. Maxwell Harper. 2017. Understanding how people use natural language to ask for recommendations. In *Proceedings of the 11th ACM Conference on Recommender Systems (RecSys '17)*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Neil D. Lawrence and Raquel Urtasun. 2009. Non-linear matrix factorization with gaussian processes. In *Proceedings of the 26th International Conference on Machine Learning (ICML '09)*, pages 601–608. ACM.

Xueguang Ma, Xinyu Zhang, Ronak Pradeep, and Jimmy Lin. 2023. Zero-shot listwise document reranking with a large language model. *arXiv preprint arXiv:2305.02156*.

Rodrigo Nogueira, Wei Yang, Kyunghyun Cho, and Jimmy Lin. 2019. Multi-stage document ranking with bert. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '19)*, pages 1129–1132, Paris, France. ACM.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, Xuanhui Wang, and Michael Bendersky. 2024. Large language models are effective text rankers with pairwise ranking prompting.

In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1504–1518. Association for Computational Linguistics.

Carl E. Rasmussen and Christopher K. I. Williams. 2006. *Gaussian Processes for Machine Learning*. MIT Press.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. 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 3982–3992. Association for Computational Linguistics.

Stephen E. Robertson, Steve Walker, Susan Jones, Micheline Hancock-Beaulieu, and Mike Gatford. 1994. Okapi at trec-3. In *Text Retrieval Conference*.

Devendra Singh Sachan, Mike Lewis, Mandar Joshi, Armen Aghajanyan, Wen-tau Yih, Joelle Pineau, and Luke Zettlemoyer. 2022. Improving passage retrieval with zero-shot question generation. *arXiv preprint arXiv:2204.07496*.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations (ICLR)*.

Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. 2023. Large language models are strong zero-shot retriever. *arXiv preprint arXiv:2304.14233*.

Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias Seeger. 2010. Gaussian process optimization in the bandit setting: No regret and experimental design. In *Proceedings of the 27th International Conference on Machine Learning (ICML)*, pages 1015–1022. Omnipress.

Ingo Steinwart. 2001. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2:67–93.

Weiwei Sun, Lingyong Yan, Xinyu Ma, Shuaiqiang Wang, Pengjie Ren, Zhumin Chen, Dawei Yin, and Zhaochun Ren. 2023. Is chatgpt good at search? investigating large language models as re-ranking agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14918–14937. Association for Computational Linguistics.

Shivani Upadhyay, Ronak Pradeep, Nandan Thakur, Nick Craswell, and Jimmy Lin. 2024. UMBRELA: The open-source reproduction of the bing relevance assessor. *arXiv preprint arXiv:2406.06519*.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Qianfeng Wen, Yifan Liu, Justin Cui, Joshua Zhang, Anton Korikov, George-Kirollos Saad, and Scott Sanner. 2025. A simple but effective elaborative query reformulation approach for natural language recommendation. *Preprint*, arXiv:2510.02656.

Qianfeng Wen, Yifan Liu, Joshua Zhang, George Saad, Anton Korikov, Yury Sambale, and Scott Sanner. 2024. Elaborative subtopic query reformulation for broad and indirect queries in travel destination recommendation. In *Proceedings of the 1st Workshop on ROEGEN (RecSys 2024 Workshop on Generative Models in RecSys)*. ArXiv:2410.01598.

Christopher K. I. Williams and Carl Edward Rasmussen. 1996. Gaussian processes for regression. In *Advances in Neural Information Processing Systems 8 (NIPS 1995)*, pages 514–520.

Christopher KI Williams and Carl Edward Rasmussen. 2006. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

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

Haochen Zhang, Anton Korikov, Parsa Farinneya, Mohammad Mahdi Abdollah Pour, Manasa Bharadwaj, Ali Pesaranghader, Xi Yu Huang, Yi Xin Lok, Zhaoqi Wang, Nathan Jones, and 1 others. 2023. Recipe-mpr: A test collection for evaluating multi-aspect preference-based natural language retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2744–2753.

Ciyou Zhu, Richard H. Byrd, Peihuang Lu, and Jorge Nocedal. 1997. Algorithm 778: L-bfgs-b, fortran routines for large scale bound constrained optimization. *ACM Transactions on Mathematical Software*, 23(4):550–560.

Honglei Zhuang, Zhen Qin, Kai Hui, Junru Wu, Le Yan, Xuanhui Wang, and Michael Bendersky. 2024. Beyond yes and no: Improving zero-shot llm rankers via scoring fine-grained relevance labels. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 358–370. Association for Computational Linguistics.

A UMBRELA Prompt for LLM Relevance Judgement

We show the `UMBRELLA` prompt used to judge the relevance between a query and passages:

UMBRELLA prompt

Given a query and a list of passages, assign each passage a relevance score from 0 to 3:

0 = Unrelated to the query.
1 = Related but does not answer the query.
2 = Partially answers the query but is unclear or mixed with extra information.
3 = Fully dedicated to answering the query.

Instructions:
- Assign 1 if the passage is somewhat related but incomplete.
- Assign 2 if it provides key information but includes unrelated content.
- Assign 3 if it solely and fully addresses the query.
- Otherwise, assign 0.

Query: `{query}`
Passages: `{passages}`

Steps:
1. Analyze the search intent.
2. Measure content alignment (M).
3. Assess passage trustworthiness (T).
4. Decide on the final score (O).

Return a JSON object mapping passage indices (starting from 0) to relevance scores. Do not include any code in your response.

B Implementation Details

We implement the following baseline methods for relevance scoring in Natural Language Recommendation (NLRec).

BM25. We employ the Okapi BM25 algorithm (Robertson et al., 1994) as implemented in the `Pyserini` toolkit, using default hyperparameters ($k_1 = 0.9$, $b = 0.4$).

Dense Retrieval (DR). We use the `all-MiniLM-L6-v2` embedding model (Karpukhin et al., 2020) with $D = 384$ and the `msmarco-distilbert-base-tas-b` model (Hofstätter et al., 2021) with $D = 768$ from Hugging Face. Candidate relevance scores are computed as the inner product between query and passage embeddings.

Cross-Encoder (CE). We use the `cross-encoder/ms-marco-MiniLM-L-6-v2` model (Reimers and Gurevych, 2019; Wolf et al., 2020) from Hugging Face, which jointly encodes query–passage pairs and outputs a fine-grained scalar relevance score. The CE is applied to the top-R passages retrieved by DR to maintain comparability with GPR-LLM and pointwise LLM-based scoring.

Pointwise LLM-based Relevance Scoring (Point.). We use the commonly adopted UMBRELA `prompt` (Upadhyay et al., 2024) and follow the procedure of Zhuang et al. (2024) to obtain LLM-based passage relevance scores. For each query q and passage p_j, the LLM outputs a vector of logits:

$$\mathbf{z} = \mathrm{LLM}(q, p_j, \texttt{prompt}) = [z_0, z_1, \ldots, z_{K-1}], \tag{10}$$

where each z_k corresponds to the logit of a predefined discrete relevance label $r_k \in \{0, 1, \ldots, K-1\}$. We compute the scalar LLM-based relevance score using the *expected relevance* (ER) formulation:

$$\mathcal{S}^{\mathrm{LLM}}_{i,j} = \sum_{k=0}^{K-1} \left(\frac{e^{z_k}}{\sum_{j=0}^{K-1} e^{z_j}} \right) \cdot r_k. \tag{11}$$

We use both a closed-source LLM (`GPT-4o`; OpenAI, 2023) and an open-source LLM (`Qwen-Next-80B`; Yang et al., 2025) for generating relevance judgments. Pointwise LLM-based scoring is applied to the top-R passages retrieved by DR to ensure fair comparability with GPR-LLM.

C How Sampling Quality Affects GPR-LLM

To better understand how the quality of the small labeled subset of passages influences model performance, we conduct a controlled experiment to examine whether augmenting the sampled subset with additional high-relevance passages can further improve GPR-LLM and how different kernel choices respond to such changes. The goal is to test a simple hypothesis: *if the sampling process were improved, could GPR-LLM achieve higher performance?*

Experimental Setup. We use the `TravelDest` dataset and begin with a baseline of 100 passages selected greedily by dense-retrieval scores. We then incrementally augment this subset by adding 1–20 additional passages drawn from two distinct supervision pools:

1. **High-relevance augmentation:** passages assigned high LLM relevance scores (2–3);

2. **Low-relevance augmentation:** passages assigned low or partial relevance scores (1–2).

Results. Results are shown in Figure 6. Adding low-relevance passages yields negligible or inconsistent changes across all kernels, suggesting that weakly informative samples provide limited benefit. In contrast, augmenting with high-relevance passages leads to clear gains *only* for the RBF kernel, which consistently improves as more high-quality samples are included. The stationary dot product and cosine kernels remain largely unaffected.

These results indicate that the stationary RBF kernel can better capture multimodal relevance scoring functions when provided with higher-quality labeled samples. This finding motivates us to move beyond purely greedy sampling based on DR rankings toward the proposed ϵ-greedy strategy, which introduces exploration to identify additional relevant passages. It also suggests promising future directions such as *uncertainty-aware active sampling* (e.g., GP-UCB (Srinivas et al., 2010)) and *diversity-optimized sampling* (e.g., core-set selection (Sener and Savarese, 2018)), which could further enhance sampling efficiency and model performance.

D Impact of Different Embeddings

Table 4 presents the results obtained using the `msmarco-distilbert-base-tas-b` encoder with GPT-4o as the LLM backbone. The results follow the same overall trend observed with the `MiniLM-L6-v2` encoder, where GPR-LLM consistently outperforms both pointwise LLM-based scoring and other baseline methods across all datasets and LLM labeling budgets. The performance gains are most pronounced at smaller labeling budgets, indicating that GPR-LLM effectively leverages limited high-quality supervision to model the underlying multimodal relevance function. This consistent pattern across distinct embedding models demonstrates that the improvements of GPR-LLM stem from its core modeling design rather than encoder-specific characteristics.

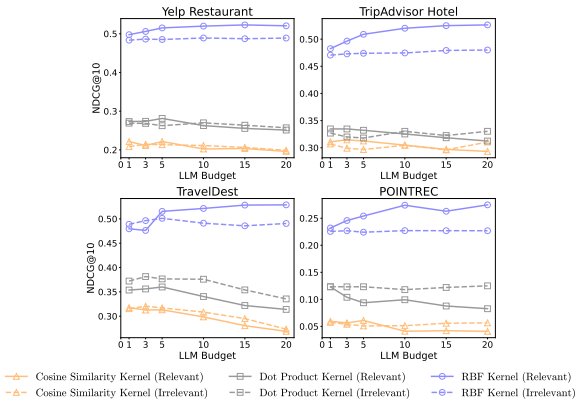

Figure 6: Effect of adding high-relevance vs. low-relevance passages to the sampling subset on `TravelDest`. We report NDCG@10 as additional passages (1–20) are added. Only the RBF kernel shows consistent gains when augmented with high-relevance labels, while dot product and cosine kernels remain insensitive to either augmentation type.

E Hyperparameter Settings for GPR

E.1 Observation Noise Variance α.

In a noisy observation setting (i.e., the ground truth relevance is unknown), the performance of GPR can be affected by the variance of the Gaussian observation noises α. Thus, we examine the difference in GPR-LLM performance under various values of α. In Figure 7, Figure 8, Figure 9 and Figure 10, we presents the performance of GPR-LLM under nDCG@10 using different value of α.

Overall, we observe a non-monotonic relationship between α and performance. Extremely low noise levels (e.g., $\alpha = 10^{-4}$, effectively assuming near-perfect LLM relevance judgments) tend to *underperform*, likely due to overfitting to the few LLM judgments. Conversely, very high noise settings (e.g., $\alpha = 10$, treating LLM relevance judgments as extremely noisy) also degrade performance by underfitting the relevance signal.

In all cases, moderate noise variance yields the best results: performance generally peaks at intermediate α values (around 10^{-2} to 10^{-1} in our experiments) and remains fairly stable across a broad mid-range. This trend is consistent across datasets, though the optimal α can vary slightly by dataset (each dataset's curve achieves its maximum nDCG at a slightly different α). Crucially, adding a reasonable amount of observation noise improves generalization, but too much noise or none at all is

Budget	Method	TravelDest				POINTREC				Yelp Restaurant				TripAdvisor Hotel			
		P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30	P@10	N@10	P@30	N@30
N/A	DPR	0.358	0.365	0.318	0.333	0.143	0.161	0.094	0.165	0.363	0.385	0.294	0.351	0.224	0.275	0.165	0.349
	BM25	0.234	0.238	0.237	0.239	0.025	0.032	0.025	0.038	0.309	0.327	0.236	0.283	0.205	0.257	0.153	0.325
25	Point.	0.344	0.379	0.313	0.339	0.154	0.188	0.094	0.178	0.340	0.382	**0.289**	**0.355**	0.227	0.302	0.165	0.369
	GPR-LLM	**0.370**	**0.402**	**0.325**	**0.356**	**0.159**	**0.192**	**0.098**	**0.183**	**0.359**	**0.397**	0.272	0.343	**0.286**	**0.378**	**0.177**	**0.419**
50	Point.	0.380	0.419	0.298	0.339	0.171	0.215	0.098	0.193	0.368	0.410	0.282	0.355	0.251	0.336	0.166	0.389
	GPR-LLM	**0.416**	**0.453**	**0.348**	**0.386**	**0.176**	**0.217**	**0.103**	**0.197**	**0.376**	**0.414**	**0.286**	**0.361**	**0.325**	**0.199**	**0.470**	
100	Point.	0.428	0.467	0.323	0.371	0.214	0.245	0.102	0.200	0.391	0.432	0.285	0.366	0.273	0.366	0.170	0.408
	GPR-LLM	**0.444**	**0.479**	**0.344**	**0.389**	**0.215**	**0.246**	**0.121**	**0.227**	**0.430**	**0.472**	**0.311**	**0.398**	**0.360**	**0.476**	**0.225**	**0.535**

Table 4: Comparison of BM25, DPR, Pointwise LLM-based Scoring (Point.), and GPR-LLM (RBF kernel, $\epsilon = 0.1$, τ includes all passages) using the `msmarco-distilbert-base-tas-b` encoder with GPT-4o as the LLM backbone across four datasets: `TravelDest`, `POINTREC`, `Yelp Restaurant`, and `TripAdvisor Hotel`, under varying LLM labeling budgets. Metrics reported are Precision@10 (P@10), NDCG@10 (N@10), Precision@30 (P@30), and NDCG@30 (N@30). Bold values indicate the best performance in each column.

detrimental.

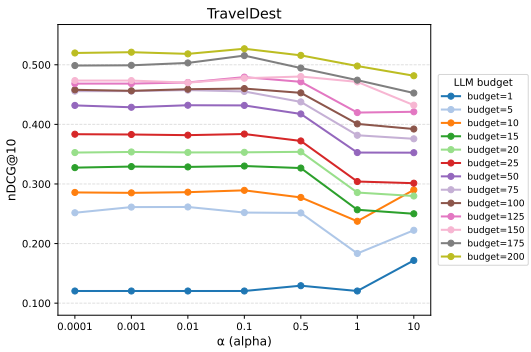

Figure 7: Performance of GPR-LLM in TravelDest with different values of α of GPR using LLM budget of 1 to 200 labels.

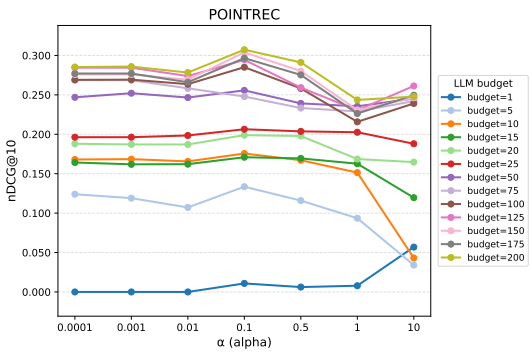

Figure 8: Performance of GPR-LLM in POINTREC with different values of α of GPR using LLM budget of 1 to 200 labels.

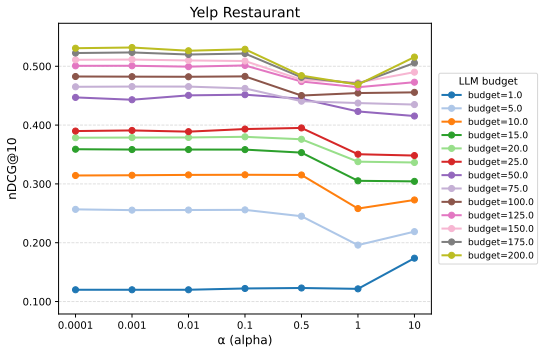

Figure 9: Performance of GPR-LLM in Yelp Restaurant with different values of α of GPR using LLM budget of 1 to 200 labels.

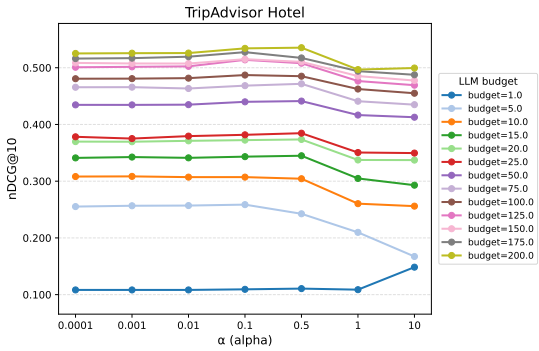

Figure 10: Performance of GPR-LLM in TripAdvisor Hotel with different values of α of GPR using LLM budget of 1 to 200 labels.

E.2 Length Scale of the RBF Kernel.

The RBF kernel in GPR is defined as

$$k(\mathbf{v}, \mathbf{v}') = \exp\left(-\frac{\|\mathbf{v} - \mathbf{v}'\|^2}{2\ell^2}\right) \qquad (12)$$

where ℓ is the length scale hyperparameter that controls the smoothness of the learned function.

Smaller values of ℓ yield more flexible, highly localized functions, while larger values impose smoother, more global behavior.

While ℓ is optimized using L-BFGS-B algorithm, we empirically evaluate the impact of varying the initial value of ℓ over the range $[0.001, 0.1, 1, 10, 100, 1000]$. As shown in Figure 11, Figure 12, Figure 13 and Figure 14, we can conclude that our algorithm is robust to different initializations of length scale values except for very small budget of LLM labels, where noise could impact the results.

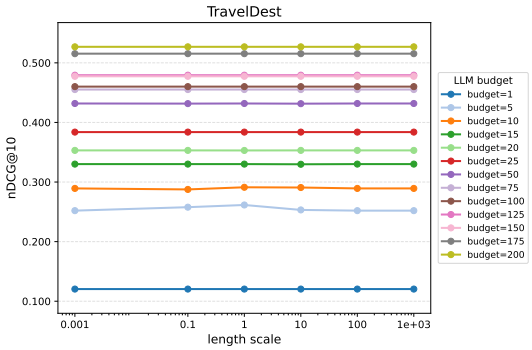

Figure 11: Performance of GPR-LLM in TravelDest across different value ofℓ using LLM budget of 1 to 200 labels with L-BFGS-B algorithm.

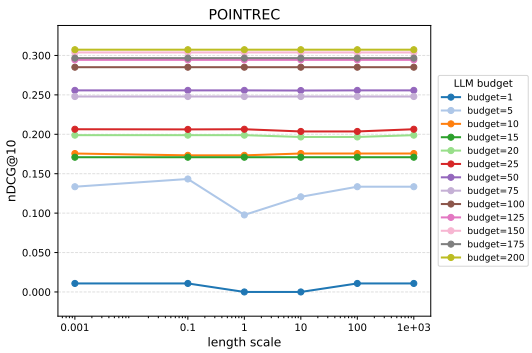

Figure 12: Performance of GPR-LLM in POINTREC across different values of ℓ using LLM budget of 1 to 200 labels with L-BFGS-B algorithm.

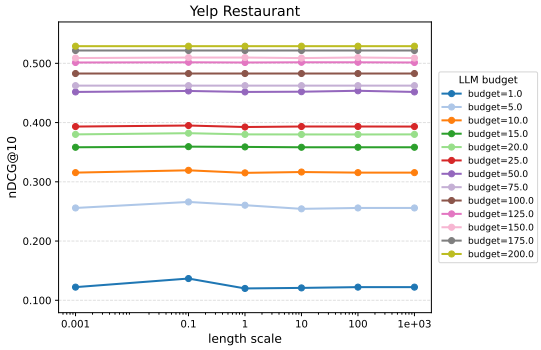

Figure 13: Performance of GPR-LLM in Yelp Restaurant across different values of ℓ using LLM budget of 1 to 200 labels with L-BFGS-B algorithm.

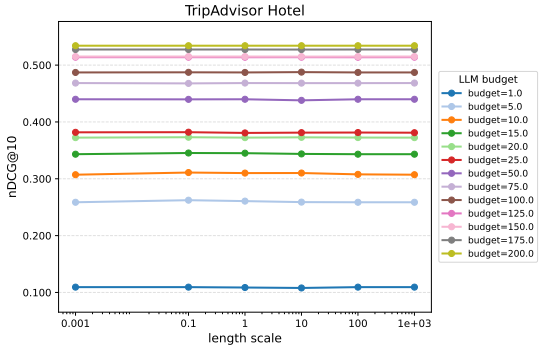

Figure 14: Performance of GPR-LLM in TripAdvisor Hotel across different values of ℓ using LLM budget of 1 to 200 labels with L-BFGS-B algorithm.

We also evaluate performance across different values of ℓ without applying the L-BFGS-B algorithm. As shown in Figure 15, Figure 16, Figure 17, and Figure 18, under a fixed α, the best performance achieved without L-BFGS-B is comparable to that with L-BFGS-B. This suggests that L-BFGS-B provides an efficient and effective method for kernel parameter selection in GP-LLM.

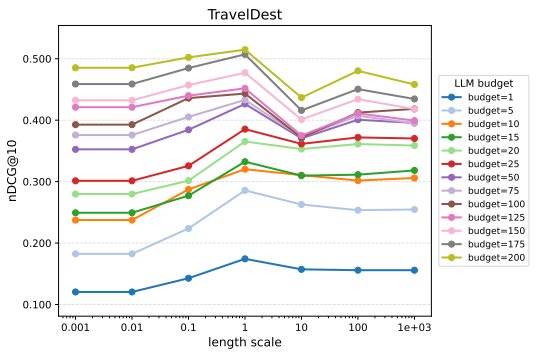

Figure 15: Performance of GPR-LLM in TravelDest across different values of ℓ using LLM budget of 1 to 200 labels without L-BFGS-B algorithm.

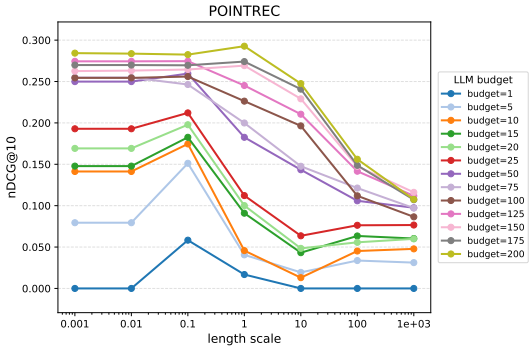

Figure 16: Performance of GPR-LLM in POINTREC across different values of ℓ using LLM budget of 1 to 200 labels without L-BFGS-B algorithm.

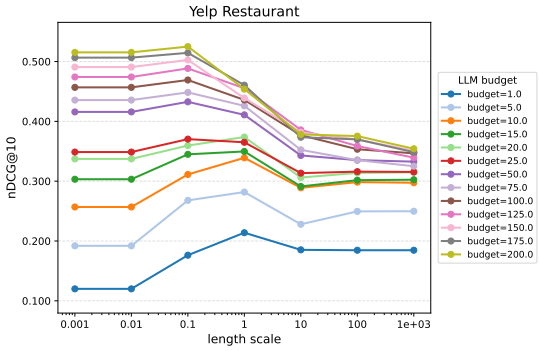

Figure 17: Performance of GPR-LLM in Yelp Restaurant across different values of ℓ using LLM budget of 1 to 200 labels without L-BFGS-B algorithm.

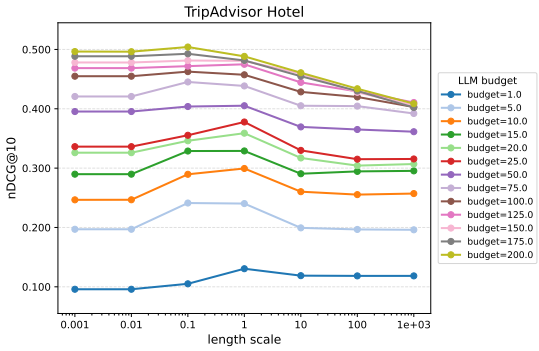

Figure 18: Performance of GPR-LLM in TripAdvisor Hotel across different value ofℓ using LLM budget of 1 to 200 labels without L-BFGS-B algorithm.

E.3 Scaling of Relevance Scores.

Figure 19 illustrates the effect of varying the rating scale used by the LLM to provide relevance judgments, with results shown for all four datasets (nDCG@10 across different LLM budgets of labels). Across all four datasets, applying extreme scaling to the LLM relevance scores consistently degrades GP-LLM's nDCG@10 performance, whereas keeping the labels at or near their original scale yields the best results.

In particular, using the original rating range (scale factor = 1) produces the highest nDCG@10 on TravelDest, POINTREC, Yelp Restaurant, and TripAdvisor Hotel, for both low and high LLM budgets. Any substantial compression of the label range (e.g., a 0.1× factor) or expansion (e.g., a 10× factor) leads to a notable drop in ranking effectiveness across all LLM budgets.

This suggests that over-compressing the relevance scores blurs meaningful differences between items, while over-expanding them amplifies noise and overemphasizes minor relevance distinctions, in both cases hurting the GP-LLM's ability to accurately rank items. Consequently, maintaining the original scale preserves the proper balance of signal to noise in the LLM labels and achieves the strongest overall nDCG@10 performance in the GP-LLM framework.

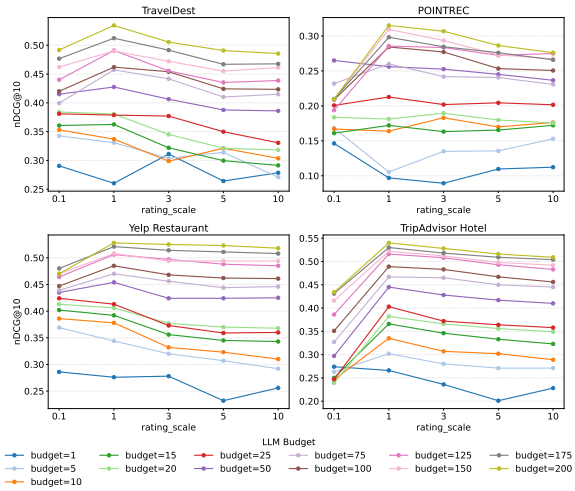

Figure 19: TravelDest, POINTREC, Yelp Restaurant, and TripAdvisor Hotel under various rating scale using LLM budget of 1 to 200 labels without L-BFGS-B algorithm.

F Hyperparameter Settings for Item-Level Relevance Aggregation

The final item-level score $\mathcal{S}_i$ for item $i \in \mathcal{I}$ is computed by aggregating the top-T passage-level scores:

$$\mathcal{S}_i = \phi\left(\left\{ \hat{f}_q(\mathbf{v}_{p_j}) \mid p_j \in \mathrm{top}_T(\mathcal{P}_i, \hat{f}_q) \right\}\right), \tag{13}$$

where ϕ is the aggregation function (e.g., mean, max) and $\mathrm{top}_T(\mathcal{P}_i, \hat{f}_q)$ returns the top-T passages

for item i ranked by the GPR relevance scoring function $\hat{f}_q$.

We vary the number T of top-ranked passages used in aggregation and evaluate its effect on item-level ranking performance. A small T may ignore informative passages, while a large T may include irrelevant noise. We vary it by [1, 3, 5, 10, 25, 50] and set LLM budget and epsilon to 100 and 0 respectively.

As shown in Figure 20, we can see that for mean aggregation method ndcg@10 increases when the number of top-k passages increases and achieves the best results when top-3 passages or top-10 passages are used. Then we observe a sharp decrease as more top-k passages are being aggregated. Our results suggest that a moderate choice of number of top-k passages (around 3-10) best balances these competing effects.

We compare different choices of aggregation functions to assess how the fusion of passage scores affects the final item recommendation quality. We test over [mean, max]. We can see that using mean as the aggregation function greatly outperforms the max method as we use more than one top-k passage for aggregation.

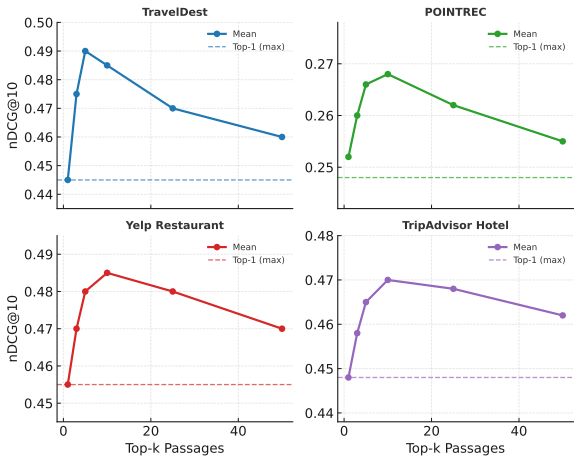

Figure 20: Performance of GPR-LLM under varying numbers of Top-T passages and different aggregation functions for item-level relevance scoring.