# SeCor: Aligning Semantic and Collaborative Representations by Large Language Models for Next-Point-of-Interest Recommendations

Shirui Wang
sr_wang@tongji.edu.cn
Tongji University
Shanghai, China

Bohan Xie
hey_bohan@tongji.edu.cn
Tongji University
Shanghai, China

Ling Ding
dling@tongji.edu.cn
Tongji University
Shanghai, China

Xiaoying Gao
gxying@tongji.edu.cn
Tongji University
Shanghai, China

Jianting Chen
tj_chenjt@tongji.edu.cn
Tongji University
Shanghai, China

Yang Xiang*
shxiangyang@tongji.edu.cn
Tongji University
Shanghai, China

## ABSTRACT

The widespread adoption of location-based applications has created a growing demand for point-of-interest (POI) recommendation, which aims to predict a user's next POI based on their historical check-in data and current location. However, existing methods often struggle to capture the intricate relationships within check-in data. This is largely due to their limitations in representing temporal and spatial information and underutilizing rich semantic features. While large language models (LLMs) offer powerful semantic comprehension to solve them, they are limited by hallucination and the inability to incorporate global collaborative information. To address these issues, we propose a novel method SeCor, which treats POI recommendation as a multi-modal task and integrates semantic and collaborative representations to form an efficient hybrid encoding. SeCor first employs a basic collaborative filtering model to mine interaction features. These embeddings, as one modal information, are fed into LLM to align with semantic representation, leading to efficient hybrid embeddings. To mitigate the hallucination, SeCor recommends based on the hybrid embeddings rather than directly using the LLM's output text. Extensive experiments on three public real-world datasets show that SeCor outperforms all baselines, achieving improved recommendation performance by effectively integrating collaborative and semantic information through LLMs.

## CCS CONCEPTS

• **Information systems → Recommender systems**.

## KEYWORDS

Recommendation System, Large Language Model, Collaborative Information

*The corresponding author.



*RecSys '24, October 14–18, 2024, Bari, Italy*

ACM ISBN 979-8-4007-0505-2/24/10
https://doi.org/10.1145/3640457.3688124

**ACM Reference Format:**
Shirui Wang, Bohan Xie, Ling Ding, Xiaoying Gao, Jianting Chen, and Yang Xiang. 2024. SeCor: Aligning Semantic and Collaborative Representations by Large Language Models for Next-Point-of-Interest Recommendations. In *18th ACM Conference on Recommender Systems (RecSys '24), October 14–18, 2024, Bari, Italy*. ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3640457.3688124

## 1 INTRODUCTION

The proliferation of mobile devices and location-based social networks has driven the growth of location-based services, including advertising, socialization, and food delivery. Point-of-interest (POI) recommendation systems play a pivotal role in these popular apps, including Foursquare, Gowalla, Meituan, and Gaode. Their goal is to suggest the next POI for a user, based on their previous check-ins and current location.

In the existing literature, most of the next POI recommenders adopt various deep neural networks to capture the temporal and sequential relationships, including RNN[15], GNN[26] and Attention[3]. These models typically rely on collaborative filtering (CF), encoding on users' and POIs' IDs. Inspired by the Large Language Models (LLMs) excellent abilities, some researchers[24, 27] also try to enhance modeling by leveraging content information, such as user profiles and reviews. They aim to utilize LLMs' strong comprehension to capture the textual semantics within descriptions. However, the above methods cannot effectively work in Next POI recommendation tasks. The reason can be summarized as follows:

(1) Traditional methods could capture global interaction signals and encode collaborative semantics. However, their modeling is always inefficient and fails to effectively capture the complex relationships inherent in check-in data[29]. Specifically, there are three main reasons. **i) Neglecting in-depth relationships**. Most previous work focused on temporal and spatial correlations, which only allow for a superficial understanding of POI, unable to capture essential information, such as cultural connotations and geographical characteristics, that may provide valuable insights. **ii) Noisy and crude connections**. The topology of the pre-defined connections is mostly generated based on statistical metrics, such as physical distance [2, 36]. However, the choice of these metrics is subjective,

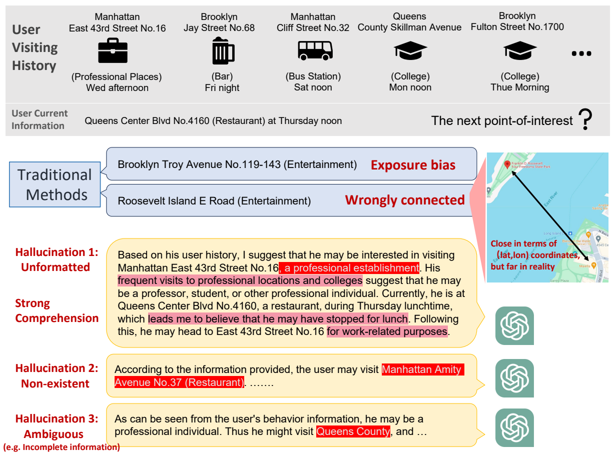


**Figure 1: An example where both traditional methods and LLM fail.**

tending to obtain missing, redundant, and erroneous POI connections, as illustrated in the second answer in Figure 1, leading to suboptimal performance.

(2) LLMs could provide strong comprehension to solve the above dilemma where traditional methods fail. They can utilize language information and uncover the intrinsic characteristics of POIs and users. Nevertheless, they also have significant limitations in POI recommendations. The most prominent issue is **hallucination**[11, 25]. As shown in Figure 1, LLMs tend to generate non-existent POIs or ambiguous answers when directly providing recommendation text. This issue is extremely severe in POI recommendation due to the quite long POI descriptions. Therefore, many LLM-based methods using output text fail in POI recommendation due to hallucination (The detailed results are shown in Section 4.6). Additionally, most LLMs cannot concretely **acknowledge overall POIs and users** in the dataset, even if input the complete information to instructions[24]. It makes LLMs unaware of the global collaborative relationship. This lack of information leads to LLMs' poor performance in re-ranking recommendation[27].

To address the challenges mentioned above, we aim to leverage both collaborative information and textual semantics to balance their strengths and weaknesses. To be noticed, we utilize LLMs' representations instead of using LLMs' output text as recommendations directly. Therefore, our recommendation results come from vector calculation, which avoids hallucination influence. In summary, We propose a next POI recommendation method that aligns **Se**mantic representations and **Co**llaborative **r**epresentations (SeCor). Collaborative representations are inherent latent intention embeddings, therefore conveying special semantics. SeCor views collaborative embeddings as one modal information, extracts it from a CF-based model and maps the initial representations to a collaborative semantics space. The collaborative semantics are prepended to the prompts. SeCor leverages the advanced representing and transfer capabilities of LLMs[17, 18] to combine collaborative and linguistic information, forming efficient hybrid representations. To take full advantage of LLM's capabilities, SeCor adopts Low-rank Adaptation (LoRA) to fine-tune it. We also design a two-stage tuning paradigm for maximizing SeCor's effectiveness. We summarize our main contributions as follows:

- We review POI recommendation systems from a novel perspective: it's a multi-modal (collaborative and semantic) information cooperation task. Simultaneously, we set templates for location descriptions and user visitation histories, and then construct instructions. This facilitates LLMs' thorough understanding of the task, thereby enhancing representation capabilities.
- We propose a new architecture, SeCor, which leverages the transfer and alignment ability of LLMs to obtain an efficient hybrid encoding of collaborative and semantic embeddings. This approach effectively addresses representation deficiencies in traditional methods while avoiding the hallucination issues of LLMs.
- Extensive experiments on three public real-world datasets show the superiority of the proposed SeCor over all baselines in Recall and NDCG, which demonstrates the effectiveness of our model.

## 2 PRELIMINARY

This work focuses on the next POI recommendation task with a set of users $\mathcal{U} = \{u_1, u_2, \cdots, u_M\}$ and a set of POIs $\mathcal{P} = \{p_1, p_2, \cdots, p_N\}$, where $M$ and $N$ denote the number of users and POIs respectively. The trajectory $\mathcal{T}_u = \{p_k\}_{k=1}^{n}$ of a user $u$ records her check-in histories, where $n$ refers to the number of POIs she visited. The overall trajectory data is $\mathcal{T} = \{\mathcal{T}_u\}_{u=1}^{M}$. We have the following definitions:

**Definition 1. Point-of-interest (POI)**. While previous work treated each POI as a (lat, lon) coordinate, in our work each POI $p_i$ is an identifier to its description text. The specific text description will be described in section 3.2.1.

**Definition 2. Next POI Recommendation**. Given the trajectory $\mathcal{T}_{u_i}$ of the user $u_i$, the objective of the next POI recommendation is to recommend $u_i$ a list of top-ranked POIs that she may have interests at the next timestamp.

## 3 METHOD

The architecture of SeCor is illustrated in Figure 2. SeCor treats the POI recommendation as a multi-modal alignment task and disregards the text output of LLM. The collaborative semantics extractor (CSE) aims to capture and initially align collaborative semantics (Section 3.1). The LLM subsequently co-encodes collaborative semantics and linguistic semantics to extract hybrid features of POIs and users (Section 3.2). We design a two-stage tuning paradigm for better optimizing (Section 3.3), enabling clearer functionality of each component of SeCor.

In this paper, we choose the Llama2-7b[30] language model and use LightGCN[13] or DirectAU[31] as CF method in CSE, while the framework is flexible, illustrated in subsection 4.5.

### 3.1 Collaborative Semantics as Prefix

LLMs often overlook global collaborative signals when solely relying on a single user visit sequence. Some research[23, 49] show that directly incorporating global interaction into prompt text does not help LLM, but reduces its performance. Therefore, it is crucial to

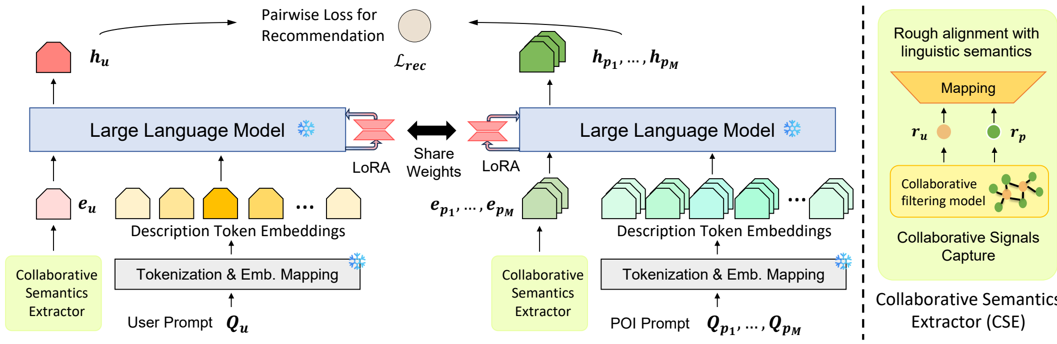


**Figure 2: The overall architecture of SeCor. The CSE learns the collaborative information and maps it to collaborative semantics. The overall information is transmitted by the prefix embeddings through LLM.**

represent global collaborative information in a non-textual manner to effectively leverage LLMs as Recommenders (LLMRec)[39].

SeCor employs a collaborative semantics extractor to integrate collaborative semantics into the LLM. The CF-based model within CSE learns collaborative representations of users and POIs, establishing a representation space. Subsequently, a mapping process is employed to linearly transform the collaborative space into collaborative semantics space, making it more accessible for the LLM to utilize.

*3.1.1 Collaborative Representation Space.* Collaborative filtering employs various methods, such as neighbor aggregation and matrix factorization, to learn interactions between users and POIs, represent collaborative signals, and make predictions[10, 29]. By leveraging the overall trajectory information $\mathcal{T}$ , we can obtain a global interaction matrix $\mathcal{I} \in \mathbb{Z}^{n \times m}$ or a global interaction graph $\mathcal{G} = (\mathcal{V}, \mathcal{I})$ $w.r.t.$ $\mathcal{V} = \{\mathcal{U}, \mathcal{L}\}$ between users and POIs. The interaction graph $\mathcal{G}$ treats each interaction as an edge and each user and POI as a node, while the interaction matrix $\mathcal{I}$ uses the interaction frequency between users and POIs as entries. CF-based models utilize either $\mathcal{G}$ or $\mathcal{I}$ to capture collaborative signals. Taking an example of a CF model utilizing the neighbor aggregation approach, it employs the interaction graph $\mathcal{G}$ for modeling. Given any $u \in \mathcal{U}$ and $p \in \mathcal{P}$, its encoding can be formulated as follows:

$$\mathbf{r_u} = f_\phi(u; \mathcal{G}); \quad \mathbf{r_p} = f_\phi(p; \mathcal{G}) \tag{1}$$

where $\mathbf{r_u}, \mathbf{r_p} \in \mathbb{R}^{1 \times d_{\mathrm{cf}}}$ convey collaborative signals, representing inherent preference of users and POIs; $d_{\mathrm{cf}}$ representing the embedding dimensions of users and POIs; $f_\phi$ denotes the CF-based method, and $\phi$ represents the trainable parameters within the model.

*3.1.2 Mapping for Initial Alignment.* To enhance the encoding efficiency of LLM, the collaborative semantics extractor maps the raw collaborative representations for preliminary alignment with textual features. The LLM then integrates and aligns the mapped collaborative semantics with textual semantics to construct new hybrid representations. This mapping phase offers flexibility in subsequent alignment, which can be adjusted based on the LLM hybrid encoding result.

The mapping transforms the representations from the original dimension $d_{\mathrm{cf}}$ to the dimension $d_{\mathrm{llm}}$, with typically $d_{\mathrm{cf}} < d_{\mathrm{llm}}$ due to discrepancy in information density (collaborative space usually contains less information). It can be formulated as follows:

$$\mathbf{e_u} = g_\eta(\mathbf{r_u}); \quad \mathbf{e_p} = g_\eta(\mathbf{r_p}) \tag{2}$$

where $\mathbf{e_u}, \mathbf{e_p} \in \mathbb{R}^{1 \times d_{\mathrm{llm}}}$ represent the mapped embedding, containing collaborative semantics; $g_\eta$ denotes the mapping function which is implemented as one layer of multiple perceptual machines in SeCor, and $\eta$ represents the trainable parameters within it.

## 3.2 Heterogeneous Representation Alignment

Collaborative semantics representations captured from CSE, are essentially special sequences that encode latent preference. SeCor regards this collaborative information as one modal, aiming to align it with another modal information, linguistic semantics. Recently, LLMs have demonstrated impressive transfer and alignment abilities in heterogeneous representation[18, 19, 47]. As a result, LLMs are well-equipped to effectively utilize and integrate collaborative representations with linguistic representations. In order to make better use of LLM, SeCor avoids inserting collaborative embeddings into prompts in a rough manner. Instead, it adds the collaborative semantics to the beginning of the sequence, similar to the "[CLS]" token in ViT[9], which represents all the information in the sequence after the complete encoding of LLM. For higher efficiency, SeCor leverages LoRA[14] to fine-tune LLM with fewer costs.

*3.2.1 Construction of POI Instruction and User Instruction.* To better convey key points in prompts, we leveraged the geopy[1] library to convert the original latitude and longitude coordinates into formatted text, adhering to a standardized template: "<City> <District/County> <Street/Road> <Number> (<Category>)". This design deliberately omits the POI's name, instead incorporating its category, to maintain data privacy while still conveying its essence.

"<Category>" descriptions empower the model to grasp the location's role and interpret user visit intentions, thereby minimizing potential noise that might introduced by the full location name. Meanwhile, the "<District/County> <Street/Road> <Number>" component enables SeCor to perceive where the POI is and how far it will be between two POIs.

[1]https://geopy.readthedocs.io/en/latest/

**Table 1: The construction of instructions.**

| | |
|---|---|
| User Instruction $Q_u$ | |
| Instruction | Given a user's POI visit history, think about his preference and recommend the next POI to visit. |
| Input | User Visiting History: <POI description> at <time>, <POI description> at <time>, ... |
| POI Instruction $Q_p$ | |
| Instruction | Given a POI's description, think about its characteristics. |
| Input | <City> <District/County> <Street/Road> <Number> (<Category>) |

The user instruction is comprised of users' interaction history, augmented with visit timestamps based on POI descriptions. The overall instruction structure is shown in Table 1. We refrained from performing additional inference or profiling on users to avoid introducing noise. Furthermore, considering the significant impact of input length, we only considered the 20 most recent visitations. This setting will be discussed in detail in section 4.3.2. The final user prompt input to LLM is denoted as $Q_u$, and the POI prompt is denoted as $Q_p$.

*3.2.2 Alignment for Hybrid Representation.* LoRA[14], a famous parameter-efficient fine-tuning method, employs low-rank matrices to optimize the original model by spatial transform. The effectiveness of such low-rank adapters in domain adaptation has been demonstrated in numerous studies[21, 44]. Notably, semantic embeddings and collaborative embeddings can be viewed as two distinct representations of the same items, belonging to separate domains. Therefore, the use of low-rank matrices provides an elegant solution for integrating and adapting these two representation spaces. The final user/POI representation can be formulated as follows:

$$\begin{aligned} \mathbf{h_u} &= \mathrm{LLM}^{e}_{\mu+\gamma}(\mathbf{e_u} \oplus \mathrm{Tokenizer}(Q_u)) \\ \mathbf{h_p} &= \mathrm{LLM}^{e}_{\mu+\gamma}(\mathbf{e_p} \oplus \mathrm{Tokenizer}(Q_p)) \end{aligned} \tag{3}$$

where $\mu$ denotes the original overall parameters of LLM, which remain frozen throughout the entire tuning process; $\gamma$ denotes the new trainable parameters introduced by LoRA; $\oplus$ denotes the concatenation operation. After getting the token embeddings from the tokenizer, SeCor inserts collaborative embeddings as prefixes in the input layer.

The resulting hybrid representations $\mathbf{h_u}$ and $\mathbf{h_p}$ are extracted from the last layer of LLM, not mapped to any token. They contain both collaborative and semantic information and would be fed into the output layer of SeCor. SeCor's output layer is set to dot product, computing the product score $\mathbf{h_{u_i}} \cdot \mathbf{h_{p_j}}$ as the preference score of user $u_i$ to the POI $p_j$.

## 3.3 Two-stage Tuning Paradigm

The efficient and effective optimization of LLMs has always been a challenge [48]. To accelerate the training process and reduce training costs, only the LoRA module and the CSE module in SeCor could be trainable. A naive approach would be training both modules simultaneously. However, it would lead to increased optimization difficulty, as the creation of the collaborative representation space and the alignment process optimized concurrently. Therefore, tuning through two or more stages is necessary for SeCor. We design a two-stage tuning paradigm as illustrated in Fig 3, and discuss the rationality of this setting in detail in section 4.3.3.

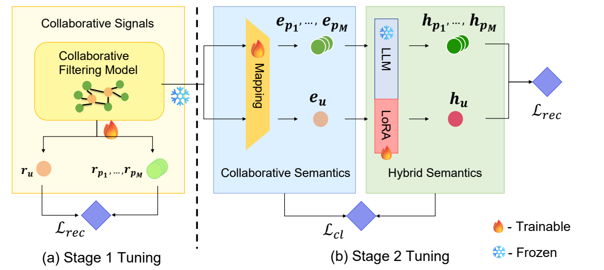


**Figure 3: The two-stage tuning paradigm in SeCor.**

*3.3.1 Stage 1: Tuning the CF Model for Capturing Collaborative Signals.* There is no LLM involvement at this stage. The choice of the CF model is flexible with the only requirement of capturing global collaborative information. The objective of this tuning stage is the same as traditional recommendation algorithms, and the loss function is also flexible as long as it urges the model to learn more robust representations. We use Bayesian Personalized Ranking (BPR)[28] in our implemented version, which can be formulated as follows:

$$\mathcal{L}_{s1} = -\sum_{(u,p_{\mathrm{pos}},p_{\mathrm{neg}})\in\mathcal{G}} \log \sigma\left(s\left(\mathbf{r_u}, \mathbf{r_{p_{pos}}}\right) - s\left(\mathbf{r_u}, \mathbf{r_{p_{neg}}}\right)\right) \tag{4}$$

where $\sigma$ is the sigmoid activation function; $s\left(\mathbf{r_u}, \mathbf{r_p}\right)$ is the estimated interaction probability between user $u$ and POI $p$, generally computed as the dot-product $\mathbf{r_u}^\top \mathbf{r_p}$; $p_{\mathrm{pos}}$ is the positive sample (preferred by the user $u$), and $p_{\mathrm{neg}}$ is the negative sample (disliked by the user $u$). This loss function ensures spatial relationships between collaborative embeddings, creating a certain order relationship within the collaborative subspace that the user is interested in.

*3.3.2 Stage 2: Tuning the LoRA Module and the Mapping Process for Alignment.* In the second stage, we aim to train SeCor for better alignment and integration.

SeCor treats collaborative representations and semantic representations as different views of the same user/POI, effectively two modalities. Therefore, we consider the same samples in the hybrid space and the collaborative space as positive pairs, and different samples in the hybrid space and the collaborative space as negative pairs, and thus use the contrastive loss InfoNCE[5] to measure consistency between the two views:

$$\mathcal{L}_{\mathrm{cl}} = \sum_{v\in\mathcal{V}} -\log \frac{\exp\left(sim\left(\mathbf{h_v}, \mathbf{e_v}/\tau\right)\right)}{\sum_{v'\in\mathcal{V}, v'\neq v} \exp\left(sim\left(\mathbf{h_v}, \mathbf{e_{v'}}/\tau\right)\right)} \tag{5}$$

where $\mathcal{V} = \{\mathcal{U}, \mathcal{L}\}$ denotes the sets of all users and POIs, $\tau$ is the temperature parameter, and $sim(\cdot,\cdot)$ is the cosine function to calculate the similarity between pairs. This loss facilitates SeCor to learn the relationship between the two views, ultimately achieving agreement between positive pairs and distinguishing them from negative ones.

**Table 2: Statistics of experimental datasets.**

| Dataset | #Users | #POIs | #Check-ins | #Avg.Seq. |
|---|---|---|---|---|
| TKY | 2,293 | 61,858 | 573,703 | 250.20 |
| NYC | 1,083 | 38,336 | 227,428 | 210.00 |
| Gowalla | 7,698 | 18,703 | 850,010 | 110.42 |

We further couple BPR loss to evaluate the final embeddings:

$$\mathcal{L}_{\text{rec}} = -\sum_{(u,p_{\text{pos}},p_{\text{neg}})\in\mathcal{G}} \log\sigma(s\left(\mathbf{h_u},\mathbf{h_{p_{pos}}}\right) - s\left(\mathbf{h_u},\mathbf{h_{p_{neg}}}\right)) \tag{6}$$

where denotations are same as Equation 4. Given the above definitions, the integrative optimization loss of stage two is as follows:

$$\mathcal{L}_{s2} = \mathcal{L}_{\text{rec}} + \lambda_1\mathcal{L}_{\text{cl}} + \lambda_2\|\eta\|_2^2 \tag{7}$$

where $\lambda_1$ denotes the strength of alignment urgency, $\lambda_2$ denotes the regularization strength, and $\eta$ is the parameters of the mapping process. Minimizing the loss makes SeCor learn to solve both recommendation and representation alignment tasks.

# 4 EXPERIMENTS

To evaluate the performance of SeCor, we conduct extensive experiments to address the following research questions: RQ1: Does SeCor effectively utilize both collaborative and semantic information to improve? RQ2-5: What impact do our design choices have on the performance of SeCor, including the CSE (RQ2), user prompts (RQ3), the tuning choice (RQ4), and the other hyper-parameters (RQ5)? RQ6: What is the computational overhead of SeCor? RQ7: How scalable is SeCor? RQ8: Why SeCor doesn't directly utilize LLM's answers?

## 4.1 Experimental Setup

*4.1.1 Datasets.* We conduct experiments on three public real-world datasets collected from two platforms, Foursquare[43] and Gowalla[6]. The detailed statistics of the three datasets are presented in Table 2.

(1) Foursquare [2]: It contains two subsets, which are records collected in Tokyo (TKY) and New York City (NYC) respectively from April 2012 to February 2013.

(2) Gowalla[3]: In order to validate that SeCor is generic, we also conduct experiments using a dataset from another LBSN service, Gowalla, which were collected from February 2009 to October 2010.

Following [34], each user's check-ins are sorted in chronological order based on the check-in timestamps and we filtered out check-in data of length less than 5.

*4.1.2 Evaluation Metrics.* To ensure comprehensive evaluation and mitigate bias, we adopt the all-rank protocol across all items to accurately assess recommendations. We employ two commonly used metrics to assess the performance of studied methods: Recall@K and NDCG@K with $K \in \{5, 10\}$.

[2]https://sites.google.com/site/yangdingqi/home/foursquare-dataset
[3]http://snap.stanford.edu/data/loc-gowalla.html

*4.1.3 Baselines.* We compare SeCor against two types of methods: traditional methods and LLM-based methods, specified as follows.

- DirectAU[31] directly optimizes alignment and uniformity of representation hypersphere space for recommendations.
- LightGCN[13] uses lightweight GCN to learn users' preferences on POIs based on a pairwise ranking loss.
- SGL[38] exploits an auxiliary self-supervised task to reinforce node representation for recommendations.
- SGRec[22] designs a Seq2Graph augmentation mechanism for mining collaborative signals among POIs.
- CAPE[3] enhances traditional CF-based methods with textual content to capture users' interests.
- STaTRL[35] view POI recommendation as a spatiotemporal and textual representation learning task, exploring multiple contexts.
- ICL[7, 32] is short for In-Context Learning, which utilizes in-context demonstrations and elaborate prompts to inspire LLM for recommendations.
- Soft-Prompting[50] is a state-of-the-art LLMRec method that uses both discrete and continuous prompts. We extend this method to the LLM llama2-7b for fair comparisons.
- POD[20] distills the discrete prompt to a set of continuous prompt vectors to bridge IDs and words and unleash the power of LLM for recommendation.

*4.1.4 Implementation Details.* SeCor is implemented with PyTorch, while most baselines are evaluated based on the unified recommendation library RecBole[42]. We employ the AdamW optimizer and warm-up strategy with the cosine scheduler for LLM-based models and the Adam optimizer for others. Each experiment of LLM-based methods is trained for a maximum of 20 epochs with a batch size of 64, while other methods train on an early-stopping strategy of 100 epochs. We investigate the learning rate within the scope of $[1e-4, 2e-4, \ldots, 1e-3]$ across all methods, and fix weight decay at 0 for all methods. In our SeCor, we search $\tau$ in the range of $[0.1, 0.2, \ldots, 1.0]$, $\lambda_1$ in the range of $[0.5, 0.1, \ldots, 1e-3]$ and $\lambda_2$ in the set 1e-3, 1e-4, 1e-5. We set the LoRA rank as 8, LoRA alpha as 16, and LoRA dropout as 0.05 following previous works[1, 14]. To ensure a fair comparison, the embedding size was fixed at 4096 for all LLM-based methods, while for CF-based methods we explored a range of 64, 128, 256. The specific hyper-parameters of each baseline followed their original settings. Our code is available at https://github.com/siri-ya/SeCor.

## 4.2 Performance Comparison (RQ1)

The comparison results between SeCor and baselines are presented in Table 3. Our model demonstrates superior performance across all datasets, indicating that SeCor can leverage both LLM and CSE to obtain more valuable and richer representations. Additionally, we have the following findings:

- Compared to methods based solely on LLM or CF, except for ICL, LLM-based methods show a slight advantage. This suggests that LLM, with reinforced understanding and reasoning ability after tuning, can perform recommendation tasks relatively well. However, LLM-based methods are limited by hallucinations, text length limits, and global information capture difficulties, making them less efficient.

**Table 3: Performance comparison on TKY, NYC, and Gowalla in terms of Recall@5/10 and NDCG@5/10. Results marked with the superscript * indicate that the method is statistically more prominent than the other baselines with a t-test p-value less than 0.02. The bold value indicates the best performance.**

| Data | | TKY | | | | NYC | | | | Gowalla | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods \ Metrics | | R@5 | R@10 | N@5 | N@10 | R@5 | R@10 | N@5 | N@10 | R@5 | R@10 | N@5 | N@10 |
| CF-based | DirectAU | 0.0334 | 0.0473 | 0.0301 | 0.0422 | 0.0272 | 0.0290 | 0.0155 | 0.0211 | 0.1284 | 0.1368 | 0.1291 | 0.1409 |
| | LightGCN | 0.0345 | 0.0498 | 0.0319 | 0.0438 | 0.0272 | 0.0281 | 0.0156 | 0.0201 | 0.1272 | 0.1338 | 0.1273 | 0.1389 |
| | SGL | 0.0372 | 0.0597 | 0.0360 | 0.0684 | 0.0421 | 0.0519 | 0.0174 | 0.0285 | 0.1341 | 0.1457 | 0.1322 | 0.1472 |
| | SGRec | 0.0702 | 0.0797 | 0.0674 | 0.0762 | 0.0465 | 0.0548 | 0.0247 | 0.0321 | 0.1584 | 0.1892 | 0.1587 | 0.1856 |
| CF + Text | CAPE | 0.0313 | 0.0471 | 0.0295 | 0.0426 | 0.0214 | 0.0303 | 0.0203 | 0.0256 | 0.1397 | 0.1684 | 0.1432 | 0.1688 |
| | STaTRL | 0.0495 | 0.0673 | 0.0461 | 0.0769 | 0.0474 | 0.0583 | 0.0261 | 0.0410 | 0.1495 | 0.1893 | 0.1438 | 0.1796 |
| LLM-based | ICL | 0.0214 | 0.0248 | 0.0192 | 0.0218 | 0.0165 | 0.0210 | 0.0152 | 0.0208 | 0.0339 | 0.0368 | 0.0237 | 0.0382 |
| | Soft-Prompting | 0.0761 | 0.0822 | 0.0739 | 0.0828 | 0.0497 | 0.0584 | 0.0340 | 0.0479 | 0.1708 | 0.1929 | 0.1492 | 0.1892 |
| | POD | 0.0694 | 0.0728 | 0.0694 | 0.0782 | 0.0473 | 0.0524 | 0.0278 | 0.0310 | 0.1693 | 0.1782 | 0.1468 | 0.1783 |
| Ours | SeCor(DirectAU) | 0.0821* | 0.0891 | 0.0803* | 0.0885 | 0.0541* | 0.0629* | **0.0376*** | 0.0509* | **0.1772*** | **0.2029*** | **0.1561*** | 0.1985* |
| | SeCor(LightGCN) | **0.0827*** | **0.0907*** | **0.0811*** | **0.0896*** | **0.0550*** | **0.0631*** | 0.0371* | **0.0514*** | 0.1768* | 0.2027* | 0.1549* | **0.1987*** |

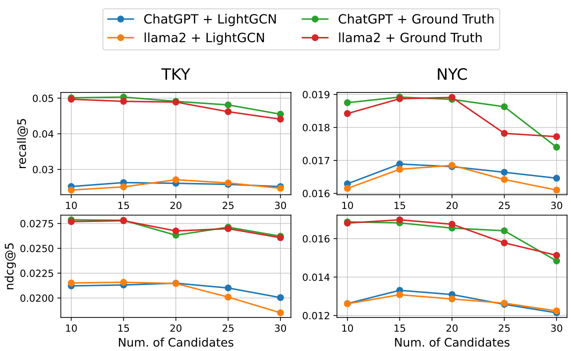


**Figure 4: Performance tendency of ICL methods w.r.t. the number of provided candidates, ranging from 10 to 30.**

- Among all LLM methods, ICL performs the worst, with significant performance improvements observed after optimization. ICL directly utilizes the language and logical reasoning abilities of LLM without involving vector operations. This indicates that vector calculations still play a necessary role in recommenders. Relying solely on text output exacerbates the shortcomings of LLM, hard to provide accurate and in-range recommendations.
- To fully explore LLM's ability, we selected ChatGPT and Llama2-13b for ICL, and the candidate POIs were selected from the top k by LightGCN or ground-truth. The results were shown in Figure 4, indicating that ICL heavily depends on the quality and quantity of the provided candidates. However, even given the ground truth as candidates, LLMs cannot provide highly reliable answers.
- Compared to methods that solely use semantic or collaborative information, approaches that combine both, such as STaTRL and POD, show more prominent results. STaTRL relies on user review data, but many real-world users have not provided specific reviews for locations. Furthermore, STaTRL directly concatenates user reviews and POI encodings to obtain the final encoding, which is a rather crude approach. For POD, it only solves the ID representation problem in LLMRec but does not convey and align the global collaborative signals to LLM, which causes inferior performance.

**Table 4: Performance comparison on TKY and NYC datasets in terms of Recall@5 and NDCG@5.**

| Dataset | Metrics | | w/o prefix | w/ random prefix | SeCor |
|---|---|---|---|---|---|
| TKY | R@5 | Avg | 0.05624 | 0.08404 | 0.08268 |
| | | Std | 0.00238 | 0.01261 | 0.00394 |
| | N@5 | Avg | 0.05839 | 0.08123 | 0.08113 |
| | | Std | 0.00223 | 0.01150 | 0.00276 |
| NYC | R@5 | Avg | 0.03816 | 0.05432 | 0.05498 |
| | | Std | 0.00202 | 0.01319 | 0.00422 |
| | N@5 | Avg | 0.01357 | 0.03328 | 0.03756 |
| | | Std | 0.00142 | 0.00598 | 0.00163 |

- SeCor gets relatively less improvement in the Gowalla. After a deeper analysis, we found that about 10.3% locations in Gowalla have been updated, resulting in the actual visitation differing from the current description text. However, most of the updated locations are cold locations that are rarely visited, bringing less influence. Despite this impact, SeCor still produced effective improvements. We believe that a new dataset in the field of POI recommendation would further highlight SeCor's outstanding performance. In addition, the sparsity problem in Gowalla is relatively minor, bringing lots of convenience for the traditional CF-based model to process.

## 4.3 Design Analysis of SeCor (RQ2-5)

This subsection focuses on model design, examining the impact of each module and exploring how much efficiency our design affords. In this subsection, the CF model in SeCor has been implemented using LightGCN.

*4.3.1 The Effect of Collaborative Information (RQ2).* The central component of our designs is the collaborative semantics extractor to enlighten LLM on the global collaborative signals. To measure its effectiveness, we compare SeCor with two variants: 1) the variant that directly omits the CF-Module, solely relying on the textual semantics (referred to as "w/o prefix"), and 2) the variant that excludes the collaborative semantics extractor but instead uses trainable random prefix embeddings (referred to as "w/ random prefix"). We conducted the comparison on 25 experiments with

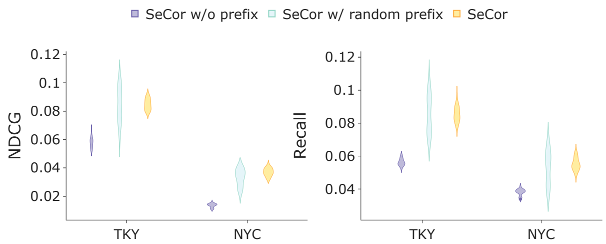


**Figure 5: Distribution of 25 randomized experiments results of the various tuning methods.**

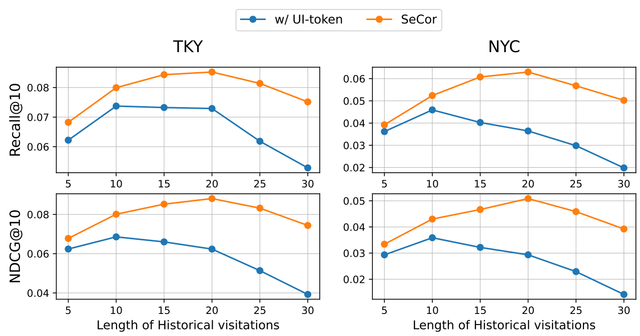


**Figure 6: Impact of the length of POI visiting sequence introduced in user prompts.**

different random seeds. The comparison results are summarized in Table 4 and Figure 5.

As the charts illustrate, the "w/o prefix" variant performs the poorest, while the "w/ random prefix" variant exhibits slightly better than default SeCor with high variability. We attribute this to the quality of carried information in the random prefixes, which is largely dependent on the initialization and challenging to optimize to meet the needs of LLM. In contrast, the prefix embeddings in SeCor directly provide the information that LLM lacks, relying less on subsequent optimization. Consequently, the "w/ random prefix" requires more training resources to converge, and although it shows a slight superiority in mean performance, it is accompanied by a large variance, making it less reliable and more difficult to reproduce.

*4.3.2 The Influence of User Prompts (RQ3).* The performance of LLMs is greatly influenced by the prompts used [37, 40]. This subsection is to explore the efficiency of prompts in SeCor.

Since the original POI description templates agree with human habits, altering them would deviate from normal presentation logic; therefore, we no longer explore variants for POI descriptions. We introduced a variant called "w/ UI-token", which uses user and POI IDs in instructions instead of the entire POI description text. To investigate the impact of POI sequence length, we explored the POI window size for both this variant and our original SeCor in the range of 5, 10, 15, 20, 25, 30. The results are shown in Figure 6.

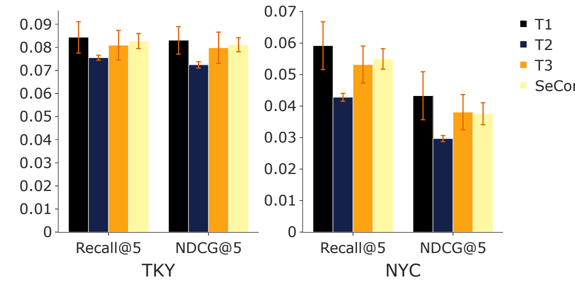


**Figure 7: Performance of different tuning strategies for SeCor on TKY and NYC.**

The results show that the "w/ UI-token" variant, which directly utilizes user and POI ID tags, does not promote the LLM to understand the underlying interaction relationships. Using only IDs, the LLM struggles to maintain long-term memory to acquire collaborative information from the entire dataset. As discussed in [8, 49], this may be because using ID tags increases tokenization redundancy in the LLM, introducing extra noise, and reducing its compression efficiency. In contrast, SeCor employs the CF-based model to effectively maintain the collaborative information in a low-rank state to reduce redundancy.

Furthermore, the length of POI sequence in the instructions can also impact their performance. Too few POIs hinder the LLM's ability to capture user preferences and habits, while too much historical information causes the text length to increase, complexity to grow, and reasoning ability to decrease[24, 27]. Particularly for the "w/ UI-token" variant, whose prompts contain ID tags, the longer prompts render it more challenging to understand the relationships between locations, leading to performance degradation. Choosing 20 or 30 POIs does not significantly affect the results, and we select 20 as the final model setting to balance performance and computational resources.

*4.3.3 The Influence of Tuning Choices (RQ4).* It has always been crucial to fine-tune LLM efficiently to achieve task-adaptive results [12]. In this section, we explore other tuning strategies to assess our tuning design as follows.

(1) T1: Two-step fine-tuning, where the first step remains consistent with training the CF-based model, but in the second step, both LoRA and the whole CSE module are fine-tuned simultaneously.
(2) T2: Two-step fine-tuning, reversing the order, fine-tuning the CF-based model after adjusting the LoRA module and the mapping.
(3) T3: One-step fine-tuning, simultaneously adjusting both the CSE and LoRA modules.

The results are shown in Figure 7. The charts show that the T1 method performed slightly better in most aspects. It adjusts and learns more from the CSE, but it also significantly increases training complexity, which makes it causes more computing resources. As for T2, its performance was inferior to the default method. We believe that during the fine-tuning process, LoRA relies more on the soft-prompt embeddings, which were heavily modified by the subsequent tuning step, leading to a decrease in LLM's alignment

**Table 5: Impact of $\lambda_1$ and $\tau$ on TKY dataset.**

| Metric | Recall@5 | | | | |
|---|---|---|---|---|---|
| $\lambda_1, \tau$ | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 |
| 0.5 | 0.0765 | 0.0774 | 0.0818 | 0.0795 | 0.0795 |
| 0.1 | 0.0787 | 0.0785 | **0.0827** | 0.0787 | 0.0769 |
| 0.05 | 0.804 | 0.0785 | 0.0813 | 0.0758 | 0.0749 |
| 0.01 | 0.0798 | 0.0765 | 0.0752 | 0.0731 | 0.0719 |

**Table 6: The average runtime performance of 5 randomized experiments on TKY dataset.**

| Method | Train(min/epoch) | Inference (s/sample) |
|---|---|---|
| ICL | - | 0.852 |
| Soft-Prompting | 198.9 | 0.752 |
| POD | 87.7 | 0.176 |
| SeCor | 220.4 | 0.189 |

capability. Regarding T3, the average effect is not significantly different from our fine-tuning scheme, but its results exhibit high variance, indicating unstable learning. We attribute this to the simultaneous fine-tuning of both components causing significant changes in the determined collaborative representation space, making it challenging for LLM to achieve alignment.

*4.3.4 Hyper-parameter Sensitivity Analysis (RQ5).* We further explore the influence of $lambda_1$ and $\tau$ for controlling alignment strength. In particular, $\lambda_1$ and $\tau$ are searched from the range of $[0.5, 0.1, 0.05, 0.01]$ and $[0.1, 0.2, 0.3, 0.4, 0.5]$, respectively. We can observe that the best performance can be achieved by $\lambda_1 = 0.1$ and $\tau = 0.3$. The chart suggests that a large $\tau$ value may reduce discrimination among negative instances, causing inferior performance, and a small $\lambda_1$ is too weak to promote alignment optimization.

## 4.4 Runtime Analysis (RQ6)

To further validate the temporal feasibility of the model, we conducted a detailed runtime comparison on an NVIDIA RTX 6000 Ada Generation GPU. The results are shown in Table 6.

For ICL, its inference time is relatively high due to the self-regressive nature of LLMs, which requires repeated computations to generate a complete text. For Soft-prompting, its architecture necessitates paired data, leading to duplicated calculations on user representations and increased time cost. POD uses IDs to construct prompts and outputs a single token to map to an ID. While POD is computationally efficient, it may result in suboptimal performance. For SeCor, its optimization of LoRA matrices and mapping procedures contributes to its relatively higher training time but superior performance. Crucially, SeCor's inference is efficient, requiring only a single computation to generate an embedding via LLM.

## 4.5 Scalability Exploration (RQ7)

In order to further assess the scalability of SeCor, we replaced some classical LLM models. The results are shown in Table 7.

The differences are insignificant, and we believe the main reason is that their model sizes are essentially the same. The improvements

**Table 7: Performance comparison on different LLMs with LightGCN as CF model.**

| Data | TKY | | NYC | |
|---|---|---|---|---|
| Used LLMs | Recall@5 | NDCG@5 | Recall@5 | NDCG@5 |
| Mixtral-7b | 0.0845 | 0.0813 | 0.0588 | 0.0411 |
| ChatGLM-6b | 0.0816 | 0.0798 | 0.0543 | 0.0362 |
| Llama-2-7b (original) | 0.0827 | 0.0811 | 0.055 | 0.0371 |

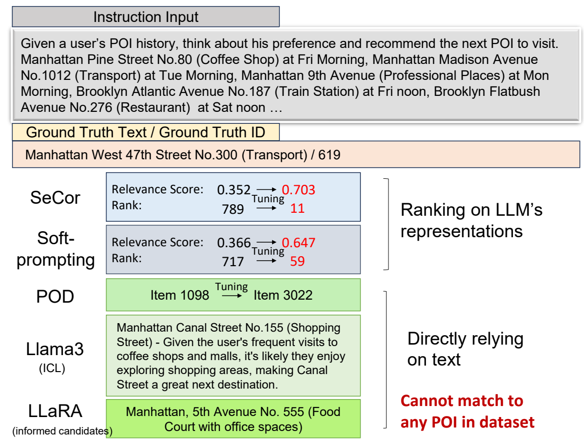


**Figure 8: Case study on relying on text output.**

on the two datasets for the new LLMs are slightly different, which we attribute to the differences in their pre-training datasets.

We attempted to replace the backbone of the CF model to further validate the scalability of the model. However, we found that substituting it with more advanced models, such as SGRec[22], did not yield significant improvements in model performance but in time cost. We believe that SGRec and many advanced models emphasize sequential relationships. However, the prompts also contain this type of information, allowing vanilla SeCor to capture these sequential patterns, resulting in minimal differences with SGRec.

## 4.6 Case Study (RQ8)

There are many advanced sequential recommendation studies using LLMs, such as LLaRA[23]. However, most of these studies rely on the text output of LLM, which brings a large performance decline in POI recommendation tasks. We select a typical example to illustrate this dilemma and test the effectiveness of SeCor, Soft-prompting, POD, Llama3(7b), and LLaRA.

Fig 8 presents a case study where text output cannot process well either using total description or id. We provide all of the POI information to LLama for overall comprehension, and the candidate sets (containing the correct one) to LLaRA, following its original setting. However, both Llama-3 and LLaRA still answer an out-of-range POI. We attribute this to the long POI description in the user history and candidates, where many similar POIs are combined, confusing for LLM to distinguish and answer correctly. Therefore, these non-existent answers cause difficulties in matching and hence cannot be assessed in our comparison. For POD, the ID tag brings some noise for LLM and cannot convey the actual contents and meaning of the POI. The inefficient prompts limit LLM's ability and didn't improve much after tuning. Both SeCor and Soft-Prompting remain vector similarity ranking in the POI recommendation, which contributes to the better utilization of LLM, getting more significant improvement after tuning. This indicates that the vector calculation still makes sense in POI recommendations where the text output cannot be used directly.

# 5 RELATED WORK

## 5.1 CF for Next POI Recommendation

Collaborative filtering has long been the foundation of recommendation algorithms, and has undergone significant expansion and evolution over time[10, 29]. Given the complexity of the next POI recommendation task, CF algorithms are often combined with graph neural networks and sequential neural networks[33, 46]. Graph neural networks explore spatial graph-structured data, while sequential neural networks analyze temporal sequence data. By flexibly utilizing, adjusting, and integrating these two approaches, many current methods have achieved impressive performance.

For example, Q. Yuan et al.[45] proposed a time-aware POI recommender system, which pay attention on added time in check-in data when calculating the users' similarity in collaborative filtering. H. Huang et al.[15] employed CF to mine GPS trajectories and considered the popularity and spatio-temporal motion behaviour. PR-RCUC[46] model integrated the region-based collaborative filtering method with a user-based mobile context. EEDN[34] focuses on both graphic and sequential relationships, successful in alleviating implicit feedback and integrating both local and global interaction patterns.

However, the data utilized in collaborative filtering is still incomplete and unable to capture the complex characteristics of users and POIs, such as the category of the POI, the geographic characteristics where the POI is located, and the meaning of the user's visit time. SeCor doesn't solely rely on the interaction data, but utilizes the collaborative information from these data to enhance the global sensitivity of LLM.

## 5.2 LMs for Next POI Recommendation

Language encompasses a wealth of information that CF-based methods fail to capture. Prominent LMs can uncover the underlying visit intentions directly by providing visit information, which traditional methods cannot. Numerous studies have incorporated language as a crucial feature in POI recommendation models. CAPE[4] and CAPRE[3] utilized user-generated textual content, obtaining word embedding vectors to express users' preferences. CPC[41] exploited user reviews to extract POI properties, user interests, and sentiment indications, enhancing recommendation performance. STaTRL[35] employed Transformer to learn long-term dependencies among visits and explored users' perspectives and POIs' reputations from textual reviews to improve the performance.

However, these models have limited language representation capabilities, and rely on user reviews which are not easily accessible. LLM has shown its powerful abilities in many research area, but there is few research on the utilization of it in next POI recommendation. SeCor utilizes the LLM's compression ability to align the language and collaborative information to get better performance.

## 5.3 LLMs for Recommendation Systems

LLMs have emerged in recent years, demonstrating their superior capabilities for many problems such as knowledge Q&A, and multi-modal tasks[16]. Recent studies[11, 51] have increasingly focused on prompt tuning and instruction tuning approaches. For instance, TALLRec[1] employs an instruction tuning strategy, designing instruction templates and leveraging the LoRA to fine-tune the llama LLM, achieving impressive results. Wu et al.[40] proposed a novel soft-prompting method, applying contrastive learning to capture user representations and encode them into prompt tokens. These above studies have overlooked collaborative signals, leading to their limited usage of LLM. CoLLM[49] utilizes a CF model and LLM to get hybrid encoding and assess the CTR by LLM's output text. LLaRA[23] splices collaborative representations with token embeddings, inserts them into prompts, and ultimately recommends based on the output text. These models utilize collaborative information, however, they still rely on the output text. This is impractical in the next POI recommendation. The hallucination of LLM makes its output incorrect or ambiguous to describe a location correctly and completely.

SeCor utilizes collaborative information, which complements LLM's lack of perception of global interaction. More important, SeCor takes advantage of the information compression capability of LLM rather than simply utilizing its text output, achieving better results by viewing next POI task as a multi-modal task.

# 6 CONCLUSION

It is less efficient to use semantic information or collaborative information solely. We proposed SeCor, aligning collaborative representations with semantic representations and utilizing LoRA to enhance the alignment capability of LLMs, thereby achieving superior integration. We conducted multiple comparative experiments and investigated the rationality of each component of SeCor. The experiments demonstrate that SeCor ensures the effectiveness and feasibility of the next POI recommendation by successfully integrating semantic information with collaborative information.

SeCor can be further extended, which in this study was conducted using Llama2-7b and LightGCN/DirectAU. We have investigated its scalability, which could be explored with more advanced LLM and more effective tuning methods in the future. The mapping process can also be further extended to more complicated and effective architecture for better alignment.

# ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China (No. 72271145).

# REFERENCES

[1] 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*. 1007–1014.

[2] Zhuo Cai, Guan Yuan, Shaojie Qiao, Song Qu, Yanmei Zhang, and Rui Bing. 2022. FG-CF: Friends-aware graph collaborative filtering for POI recommendation. *Neurocomputing* 488 (2022), 107–119.

[3] Buru Chang, Yookyung Koh, Donghyeon Park, and Jaewoo Kang. 2020. Content-aware successive point-of-interest recommendation. In *Proceedings of the 2020 SIAM International Conference on Data Mining*. SIAM, 100–108.

[4] Buru Chang, Yonggyu Park, Donghyeon Park, Seongsoon Kim, and Jaewoo Kang. 2018. Content-aware hierarchical point-of-interest embedding model for successive poi recommendation.. In *IJCAI*, Vol. 20. 3301–3307.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[6] Eunjoon Cho, Seth A Myers, and Jure Leskovec. 2011. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1082–1090.

[7] Sunhao Dai, Ninglu Shao, Haiyuan Zhao, Weijie Yu, Zihua Si, Chen Xu, Zhongxiang Sun, Xiao Zhang, and Jun Xu. 2023. Uncovering chatgpt's capabilities in recommender systems. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1126–1132.

[8] Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, Marcus Hutter, and Joel Veness. 2024. Language Modeling Is Compression. In *ICLR*.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

[10] Michael D Ekstrand, John T Riedl, Joseph A Konstan, et al. 2011. Collaborative filtering recommender systems. *Foundations and Trends® in Human–Computer Interaction* 4, 2 (2011), 81–173.

[11] Wenqi Fan, Zihuai Zhao, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Jiliang Tang, and Qing Li. 2023. Recommender systems in the era of large language models (llms). *arXiv preprint arXiv:2307.02046* (2023).

[12] Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. 2024. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608* (2024).

[13] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[14] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

[15] Haosheng Huang and Georg Gartner. 2014. Using trajectories for collaborative filtering-based POI recommendation. *International Journal of Data Mining, Modelling and Management* 6, 4 (2014), 333–346.

[16] Jie Huang and Kevin Chen-Chuan Chang. 2023. Towards Reasoning in Large Language Models: A Survey. In *Findings of the Association for Computational Linguistics: ACL 2023*. 1049–1065.

[17] Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. 2024. Language is not all you need: Aligning perception with language models. *Advances in Neural Information Processing Systems* 36 (2024).

[18] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*. PMLR, 19730–19742.

[19] Juncheng B Li, Jackson Sam Michaels, Laura Yao, Lijun Yu, Zach Wood-Doughty, and Florian Metze. 2023. Audio-Journey: Efficient Visual+LLM-aided Audio Encodec Diffusion. In *Workshop on Efficient Systems for Foundation Models @ ICML2023*. https://openreview.net/forum?id=vzMXsTCdFB

[20] Lei Li, Yongfeng Zhang, and Li Chen. 2023. Prompt Distillation for Efficient LLM-based Recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management* (<conf-loc>, <city>Birmingham</city>, <country>United Kingdom</country>, </conf-loc>) *(CIKM '23)*. Association for Computing Machinery, New York, NY, USA, 1348–1357. https://doi.org/10.1145/3583780.3615017

[21] Wen Li, Zheng Xu, Dong Xu, Dengxin Dai, and Luc Van Gool. 2018. Domain Generalization and Adaptation Using Low Rank Exemplar SVMs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 5 (2018), 1114–1127. https://doi.org/10.1109/TPAMI.2017.2704624

[22] Yang Li, Tong Chen, Yadan Luo, Hongzhi Yin, and Zi Huang. 2021. Discovering collaborative signals for next POI recommendation with iterative Seq2Graph augmentation. In *IJCAI International Joint Conference on Artificial Intelligence*. AAAI Press, 1491–1497.

[23] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. LLaRA: Large Language-Recommendation Assistant. arXiv:2312.02445 [cs.IR]

[24] Jianghao Lin, Rong Shan, Chenxu Zhu, Kounianhua Du, Bo Chen, Shigang Quan, Ruiming Tang, Yong Yu, and Weinan Zhang. 2023. Rella: Retrieval-enhanced large language models for lifelong sequential behavior comprehension in recommendation. *arXiv preprint arXiv:2308.11131* (2023).

[25] Junling Liu, Chao Liu, Peilin Zhou, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? a preliminary study. *arXiv preprint arXiv:2304.10149* (2023).

[26] Yifang Qin, Yifan Wang, Fang Sun, Wei Ju, Xuyang Hou, Zhe Wang, Jia Cheng, Jun Lei, and Ming Zhang. 2023. DisenPOI: Disentangling sequential and geographical influence for point-of-interest recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. 508–516.

[27] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Representation learning with large language models for recommendation. *arXiv preprint arXiv:2310.15950* (2023).

[28] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[29] Sadaf Safavi, Mehrdad Jalali, and Mahboobeh Houshmand. 2022. Toward point-of-interest recommendation systems: A critical review on deep-learning Approaches. *Electronics* 11, 13 (2022), 1998.

[30] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[31] Chenyang Wang, Yuanqing Yu, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2022. Towards representation alignment and uniformity in collaborative filtering. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 1816–1825.

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

[33] Wei Wang, Junyang Chen, Jinzhong Wang, Junxin Chen, Jinquan Liu, and Zhiguo Gong. 2019. Trust-enhanced collaborative filtering for personalized point of interests recommendation. *IEEE Transactions on Industrial Informatics* 16, 9 (2019), 6124–6132.

[34] Xinfeng Wang, Fumiyo Fukumoto, Jin Cui, Yoshimi Suzuki, Jiyi Li, and Dongjin Yu. 2023. Eedn: Enhanced encoder-decoder network with local and global context learning for poi recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 383–392.

[35] Xinfeng Wang, Fumiyo Fukumoto, Jiyi Li, Dongjin Yu, and Xiaoxiao Sun. 2023. STaTRL: Spatial-temporal and text representation learning for POI recommendation. *Applied Intelligence* 53, 7 (2023), 8286–8301.

[36] Zhaobo Wang, Yanmin Zhu, Chunyang Wang, Wenze Ma, Bo Li, and Jiadi Yu. 2023. Adaptive Graph Representation Learning for Next POI Recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 393–402.

[37] Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382* (2023).

[38] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 726–735.

[39] Likang Wu, Zhi Zheng, Zhaopeng Qiu, Hao Wang, Hongchao Gu, Tingjia Shen, Chuan Qin, Chen Zhu, Hengshu Zhu, Qi Liu, et al. 2023. A survey on large language models for recommendation. *arXiv preprint arXiv:2305.19860* (2023).

[40] Yiqing Wu, Ruobing Xie, Yongchun Zhu, Fuzhen Zhuang, Xu Zhang, Leyu Lin, and Qing He. 2024. Personalized Prompt for Sequential Recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2024), 1–14. https://doi.org/10.1109/TKDE.2024.3357498

[41] Shuning Xing, Fang'ai Liu, Qianqian Wang, Xiaohui Zhao, and Tianlai Li. 2019. Content-aware point-of-interest recommendation based on convolutional neural network. *Applied Intelligence* 49 (2019), 858–871.

[42] Lanling Xu, Zhen Tian, Gaowei Zhang, Junjie Zhang, Lei Wang, Bowen Zheng, Yifan Li, Jiakai Tang, Zeyu Zhang, Yupeng Hou, Xingyu Pan, Wayne Xin Zhao, Xu Chen, and Ji-Rong Wen. 2023. Towards a More User-Friendly and Easy-to-Use Benchmark Library for Recommender Systems. In *SIGIR*. ACM, 2837–2847.

[43] Dingqi Yang, Daqing Zhang, Vincent W Zheng, and Zhiyong Yu. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45, 1 (2014), 129–142.

[44] Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178* (2023).

[45] Quan Yuan, Gao Cong, Zongyang Ma, Aixin Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proceedings of the 36th*

*international ACM SIGIR conference on Research and development in information retrieval*. 363–372.

[46] Jun Zeng, Haoran Tang, Yizhu Zhao, Min Gao, and Junhao Wen. 2021. PR-RCUC: A POI recommendation model using region-based collaborative filtering and user-based mobile context. *Mobile Networks and Applications* 26, 6 (2021), 2434–2444.

[47] Mengmei Zhang, Mingwei Sun, Peng Wang, Shen Fan, Yanhu Mo, Xiaoxiao Xu, Hong Liu, Cheng Yang, and Chuan Shi. 2024. GraphTranslator: Aligning Graph Model to Large Language Model for Open-ended Tasks. *arXiv preprint arXiv:2402.07197* (2024).

[48] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2024. Instruction Tuning for Large Language Models: A Survey. arXiv:2308.10792 [cs.CL]

[49] Yang Zhang, Fuli Feng, Jizhi Zhang, Keqin Bao, Qifan Wang, and Xiangnan He. 2023. Collm: Integrating collaborative embeddings into large language models for recommendation. *arXiv preprint arXiv:2310.19488* (2023).

[50] Zizhuo Zhang and Bang Wang. 2023. Prompt learning for news recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 227–237.

[51] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).