OneLoc: Geo-Aware Generative Recommender Systems for Local Life Service

Zhipeng Wei*
Kuaishou Inc., Beijing, China
weizhipeng@kuaishou.com

Kuo Cai*
Kuaishou Inc., Beijing, China
caikuo@kuaishou.com

Junda She
Kuaishou Inc., Beijing, China
shejunda@kuaishou.com

Jie Chen
Kuaishou Inc., Beijing, China
chenjie20@kuaishou.com

Minghao Chen
Kuaishou Inc., Beijing, China
chenminghao@kuaishou.com

Yang Zeng
Kuaishou Inc., Beijing, China
zhengchengyi@kuaishou.com

Qiang Luo†
Kuaishou Inc., Beijing, China
luoqiang@kuaishou.com

Wencong Zeng
Kuaishou Inc., Beijing, China
zengwencong@kuaishou.com

Ruiming Tang†
Kuaishou Inc., Beijing, China
tangruiming@kuaishou.com

Kun Gai
Unaffiliated, Beijing, China
gai.kun@qq.com

Guorui Zhou†
Kuaishou Inc., Beijing, China
zhouguorui@kuaishou.com

Abstract

Local life service is a vital scenario in Kuaishou App, where video recommendation is intrinsically linked with store's location information. Thus, recommendation in our scenario is challenging because we should take into account user's interest and real-time location at the same time. In the face of such complex scenarios, end-to-end generative recommendation has emerged as a new paradigm, such as OneRec in the short video scenario, OneSug in the search scenario, and EGA in the advertising scenario. However, in local life service, an end-to-end generative recommendation model has not yet been developed as there are some key challenges to be solved. The first challenge is how to make full use of geographic information. The second challenge is how to balance multiple objectives, including user interests, the distance between user and stores, and some other business objectives. To address the challenges, we propose OneLoc. Specifically, we leverage geographic information from different perspectives: (1) geo-aware semantic ID incorporates both video and geographic information for tokenization, (2) geo-aware self-attention in the encoder leverages both video location similarity and user's real-time location, and (3) neighbor-aware prompt captures rich context information surrounding users for generation. To balance multiple objectives, we use reinforcement learning and propose two reward functions, i.e., geographic reward and GMV reward. With the above design, OneLoc achieves outstanding offline and online performance. In fact, OneLoc has been deployed in local life service of Kuaishou App. It serves 400 million active users daily, achieving 21.016% and 17.891% improvements in terms of gross merchandise value (GMV) and orders numbers.

CCS Concepts

• **Information Systems → Recommendation Systems.**.

Keywords

Location-Based Recommendation, Generative Recommendation, Large Language Model

ACM Reference Format:
Zhipeng Wei, Kuo Cai, Junda She, Jie Chen, Minghao Chen, Yang Zeng, Qiang Luo†, Wencong Zeng, Ruiming Tang†, Kun Gai, and Guorui Zhou. 2018. OneLoc: Geo-Aware Generative Recommender Systems for Local Life Service. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Local life service recommendation (LLSR) [8, 18] has become a critical scenario for some major internet platforms, such as Kuaishou, Meituan and Douyin. In the local life service of Kuaishou App, videos with stores' location information displayed in the left-bottom corner are uploaded. LLSR aims to recommend such videos to nearby users to attract their consumption. For the ease of understanding, an illustrated example is presented in Figure 1. When a user enters the Kuaishou App, her real-time location is recorded[1] and LLSR recommends videos with the consideration of both the user's interests and location information. For instance, at the same location, two users want to buy a drink and open Kuaishou App. Meanwhile in local life service scenario, there are three videos nearby, which

*Equal contribution.
†Corresponding author.

Conference acronym 'XX, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

[1]Local life service is disabled if the user forbids Kuaishou App to visit her real-time location.

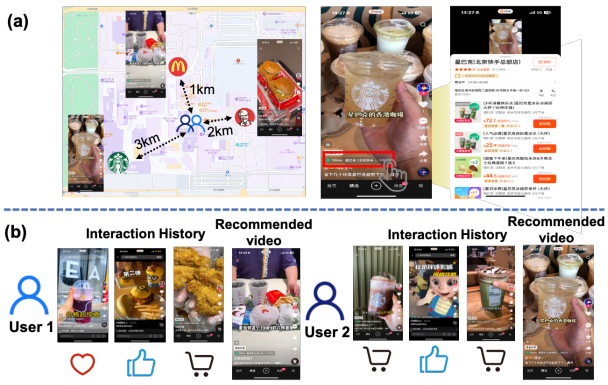

Figure 1: The background of local life service in Kuaishou. (a) In local life service, we recommend videos with stores' location information displayed in the left-bottom (red box). After clicking, it will display detail store and products information. (b) Our system makes recommendation according to users' preference and the distance of the stores.

are about McDonald's, KFC and Starbucks. The first user recently took positive actions (such as sharing, liking, etc) on videos about KFC, and LLSR would recommend the video of McDonald's instead of KFC to the user, because McDonald's is closer. The second user recently made multiple purchases of Starbucks and LLSR would recommend the video of Starbucks even though it is farther compared with KFC and McDonald's. The above example indicates that user's interests and real-time location are both vital in our scenario.

Recently, Large Language Models (LLMs) [14] based on autoregressive generation have demonstrated powerful capabilities in zero-shot learning and multi-domain generalization. This breakthrough success drives the emergence of a new paradigm of recommendation [1, 3, 5, 6, 10, 15, 17, 20, 22–24, 26–28] - the shift from matching-based to generation-based approaches, known as generative recommendation. As an innovative paradigm, it demonstrates superior performance compared to traditional cascaded recommendation architectures across various industrial applications. In short video recommendation scenario, OneRec [28] proposes a generative recommendation paradigm with reward models to align with user preference and industry requirements. In search scenario, Onesug [3] proposes an end-to-end generative framework for e-commerce query suggestion. In advertising scenario, EGA [27] designs an end-to-end generative advertising system for critical advertising requirements, such as explicit bidding, creative selection, ad allocation, and payment computation. In POI recommendation scenario, GNPR-SID [17] mitigates the generative recommendation paradigm and achieves promising result. However, in the local life service scenario, generative recommendation models have not yet been developed as there are some key challenges that need to be solved. The *first* challenge is to make full use of geographic information. Existing works propose to utilize geographic from different aspects. GNPR-SID [17] uses geographic information to construct semantic IDs, thus allocating each item a geography-aware representation. Rotan [2] leverages temporal information as positional encoding, as part of input to attention mechanism. TPG [12] proposes using temporal information as prompts to guide location recommendation. As can be observed above, existing works utilize geographic information from either *representation* perspective or *decoder prompt* perspective. Such strategies are far from making full use of geographic information. The *second* challenge is to balance multiple objectives, such as user interests, the distance between user and stores, and some other business objectives. In the field of LLMs, ChatGPT [14] uses reinforcement learning of human feedback (RLHF) to balance model's general capabilities and user experience. In short video recommendation, Onerec [28] balances multiple recommendation objectives through carefully designed rewards. However, how to balance multiple objectives in local life service has not yet been researched.

To address these two challenges, we propose OneLoc (One Model for Local Life Service), an end-to-end generative recommendation model tailored for the local life service scenario. OneLoc is with encoder-decoder structure and trained in a two-stage paradigm, following [28], i.e., pre-training and post-traninig (reinforcement learning). To make full use of geographic information (w.r.t. the first challenge), we design three key components to enhance such information: (1) *geo-aware semantic IDs* from *representation* perspective, (2) *geo-aware self-attention* from *encoder attention* perspective and (3) *neighbor-aware prompt* from *decoder prompt* perspective. *Firstly*, traditional methods (such as Rotan [2] and TPG [12]), encode geographic information as an independent feature (represented as numeric IDs), which fails to model semantic relationships between similar locations. Similar to GNPR-SID [17], we represent videos by Semantic IDs (SIDs) which are tokens derived from textual video descriptions and other features. Geographic information is integrated into the raw video features such that SIDs of videos contain geographic semantics, for which we refer to *geo-aware SIDs*. *Secondly*, we propose *geo-aware self-attention* structure to extract user behavior sequential patterns with geographic information encoded. To the best of our knowledge [25], this is the first architecture to model these two kinds of information simultaneously. Geo-aware self-attention calculates attention scores with the user's interacted video sequence and the corresponding location sequence. More specifically, the attention score between two videos is defined as the combination of two parts: (1) content similarity using comprehensive video embedding (with video information and its geographic information) as Queries and Keys and (2) location context similarity between the two videos to further enhance the location semantics. *Thirdly*, existing methods [2, 12, 17] utilize user's coordinates or timestamps as prompts to guide generating recommendation. Besides user's own location, we additionally utilize surrounding location information to model richer user's geographic context. More specifically, *neighbor-aware prompt* is constructed in decoder to guide generation by modeling user's real-time location as Query and surrounding location as Key and Value such that neighborhood information can be effectively integrated into the user's location.

To balance multiple objectives (w.r.t. the second challenge), we design a reward function, including both geographic reward signal and GMV reward signal for reinforcement learning. In LLSR scenario, the distance between a user and stores in the videos is a critical factor in determining the user's consumption in these stores. Therefore, we design a geographic reward function, where

a closer location would get a higher reward. Besides distance, GMV is another vital objective, therefore we further design the GMV reward function. Finally, the geographic reward function and GMV reward function are used to guide reinforcement learning.

To summarize, our contributions are as follows:

- We propose OneLoc, an end-to-end generative recommender system for short-video local life service. The framework integrates a generative architecture with a business-value-optimized reinforcement learning module, significantly outperforming traditional cascading recommendation models.
- We propose three core components to make full use of geographic information: (1) a geo-aware tokenizer (generating geo-aware SIDs) that combines geographic semantics with multi-modal video information, (2) a geo-aware self-attention structure which captures user behavior sequential patterns with geographic information encoded and (3) a neighbor-aware prompt in decoder to guide recommendation generation which considers user's location as well as neighborhood information.
- In reinforcement learning phase, we propose two reward functions: (1) Geographic Reward to reinforce the distance factor between a user and stores in videos and (2) GMV Reward for business objectives.
- Extensive offline experiments on Kuaishou large-scale industry dataset demonstrate the effectiveness of OneLoc, compared to traditional models (including generative models). Ablation studies are also conducted which validates the functionalities of our proposed components.
- OneLoc has been deployed in local life service of Kuaishou App and achieves 21.016% and 17.891% improvements in terms of GMV and order numbers. Now OneLoc serves the full traffic in our system, supporting 400 million users in local life service daily.

2 Method

In this section, we first elaborate on the problem formulation in our scenario. After that, we introduce how we leverage geographic information in an encoder-decoder architecture, including core designs geo-aware tokenizer, geo-aware self-attention in the encoder and neighbor-aware prompt in the decoder. In addition, we elaborate on how we design the geographic reward and the GMV reward in reinforcement learning. In the end, we introduce the final loss of our framework. The overall framework is depicted in Figure 2.

2.1 Problem Formulation

Let $\mathcal{U}$, $\mathcal{V}$ and $\mathcal{L}$ represent the set of users, videos, and location, respectively. In our task, each video $v \in \mathcal{V}$ is assigned a location $l \in \mathcal{L}$. The location is actually a GeoHash block and contains rich context information, including geographical coordinates, brand, and category. Thus, each video can be represented by a video embedding e^v, a location id embedding e^{lid} and a location context embedding e^{lc}. In our scenario, the location l_u where a user interacts with videos is also associated with a context embedding e^{lc}_u. Furthermore, each video is mapped to a semantic ID, denoted by $Q_v = (q^1_v, q^2_v, ..., q^T_v)$, where T is the number of codebooks. Given a user's interacted video sequence $S = \{v_1, v_2, ..., v_{t-1}\}$ and real-time location l_u, the objective is to predict the next video v_t that would attract consumption, which is formulated as maximize $p(v_t|l_u, S)$.

2.2 Geo-aware Semantic IDs

Following OneRec [28], we use res-kmeans to generate semantic IDs, which employs residual quantization to map video embedding into multi-level discrete codes. To inject geographic information, each residual r^0_i in the initial residual set $\mathcal{R}^0$ is represented as an embedding of video content and location context information, extracted by a multimodal large language model. At each layer i, we construct the codebook C^i by applying K-means clustering on the residuals $\mathcal{R}^i$:

$$C^i = K\text{-}means(\mathcal{R}^i, N_c), \tag{1}$$

where the codebook $C^i = \{c^i_k | k = 1, 2, ..., N_c\}$ is the centroids set derived from *K-means*, N_c is the codebook size.

With the codebook C^i, we can compute the nearest centroid index q^i_j for each residual r^i_j in the residual set $\mathcal{R}^i$. The residual r^{i+1}_j of the next layer is the difference between $\mathcal{R}^i_j$ and its nearest centroid:

$$q^i_j = \arg\min_k ||c^i_k - r^i_j||, \tag{2}$$

$$r^{i+1}_j = r^i_j - c^i_{q^i_j}, \tag{3}$$

where the index q^i_j is the i-th code of the semantic ID.

The above process iterates T times and we can get T codebooks $(C^1, C^2, ..., C^T)$. In our scenario, we set T to 3. Thus, we can transform the target video v_t into its corresponding semantic ID $Q_t = Tokenize(v_t) = (q^1_t, q^2_t, q^3_t)$.

2.3 Encoder

The encoder is used to encode the user behavior sequence for useful information extraction. In this section, we first introduce the input and architecture of the encoder. After that, we elaborate on the geo-aware self-attention, which is the core module to capture user's behavior patterns.

2.3.1 Multi-behavior Sequence. In our scenario, watching sequences are crucial for capturing user preference. To capture different scales of user behavior patterns, we additionally incorporate user clicking and purchasing sequence:

$$S = \{S^{watch}, S^{click}, S^{pay}\}, \tag{4}$$

where S^{watch}, S^{click} and S^{pay} are the watching, clicking and purchasing sequence, respectively.

The multi-behavior sequence S is then transformed into a comprehensive embedding sequence Z and a location context embedding sequence E^{lc}, which are sent to the encoder:

$$Z = \{z_1, z_2, ..., z_i, ..., z_{|S|}\}, \tag{5}$$

$$z_i = MLP(Concat(e^v_i, e^{lid}_i, e^{lc}_i)), \tag{6}$$

$$E^{lc} = \{e^{lc}_1, e^{lc}_2, ..., e^{lc}_i, ..., e^{lc}_{|S|}\}, \tag{7}$$

where $|S|$ is the length of sequence S, *Concat* is the concatenate operation along the feature dimension, z_i is the video embedding, e^v_i, e^{lid}_i and e^{lc}_i are the video id embedding, location id embedding

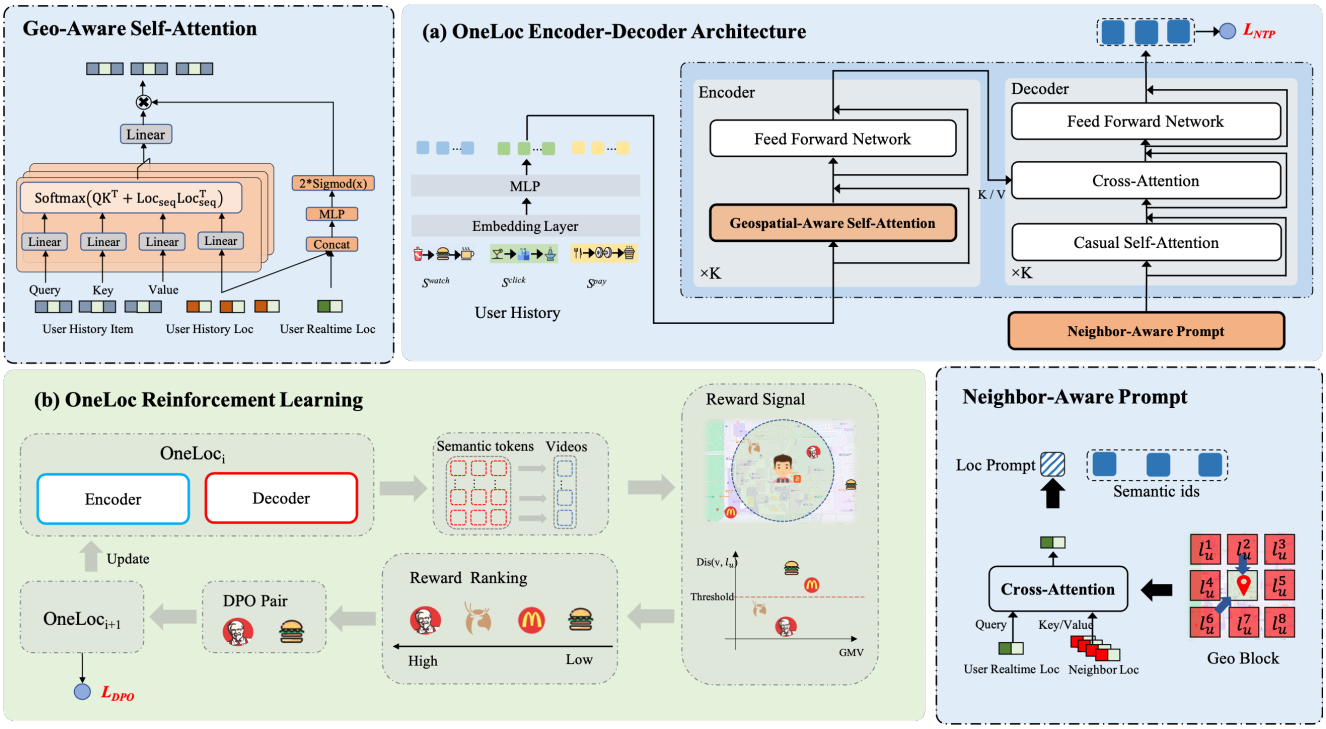

Figure 2: The overall framework of OneLoc includes encoder-decoder architecture and reinforcement learning. (a) The core module in encoder is geo-aware self-attention, which introduces location similarity into attention scores and leverages user's real-time location as a gate to control the outputs. The outputs of the encoder act as keys and values of the cross-attention in the decoder. Meanwhile, a neighbor-aware prompt is use to control the generation, which aggregates the surrounding context using cross-attention. Finally, the output of the decoder is used to calculate the next token prediction loss. (b) With the model parameters initialized , we sample multiple videos from the probability distribution. After that, we scores the videos using the geographic reward and GMV reward. Finally, we choose videos with the highest score and lowest score as preference pair for direct preference optimization.

and location context embedding of the i-th video in the sequence, respectively, E^{lc} is the location context embedding sequence.

2.3.2 Encoder architecture. The embedding sequences mentioned above, together with the user location, are then sent to the encoder, which stacks K transformer blocks. Each block contains a *GA-Attn* (Geo-aware Self-attention) module and a *FFN* module with *RMSNorm*. Formally:

$$\widetilde{Z}^{i+1} = Z^i + \textit{GA-Attn}(\textit{RMSNorm}(Z^i), E^{lc}, e_u^{lc}), \tag{8}$$

$$Z^{i+1} = \widetilde{Z}^{i+1} + \textit{FFN}(\textit{RMSNorm}(\widetilde{Z}^{i+1})), \tag{9}$$

$$Z^0 = Z, \tag{10}$$

where Z is the input embedding sequence of user u, Z^i is the output of the i-th encoder layer, e_u^{lc} is the context embedding of user's location l_u, *GA-Attn* is the core module to capture user behavior patterns.

2.3.3 Geo-aware Self-attention. *GA-Attn* module is proposed to capture relevant behaviors from the user's interaction history according to the user's real-time location. Specifically, the attention score is defined as the combination of two parts: (1) comprehensive similarity using the comprehensive embedding sequence Z as Queries and Keys, and (2) location context similarity using location context embedding sequence E^{lc} to further enhance location semantics.

$$A = \textit{Softmax}((ZW_q)(ZW_k)^T/\sqrt{d} + E^{lc}(E^{lc})^T), \tag{11}$$

$$\widetilde{O} = A(ZW_v)W_o, \tag{12}$$

where W_q, W_k, W_v, W_o are the weight matrix of query, key, value and output, E^{lc} is the location context sequence.

To inject user's real-time location information, we further leverage the location context embedding e_u^{lc} as a gate:

$$g_i = 2 * \textit{Sigmoid}(\textit{MLP}(\textit{Concat}(e_u^{lc}, E_{i*}^{lc}))), \tag{13}$$

$$O_{i*} = g_i \widetilde{O}_{i*}, \tag{14}$$

where E_{i*}^{lc} is the i-th row of E^{lc}, $g_i \in (0, 2)$ is the scaling parameter, O_{i*} is the i-th row of the output of *GA-Attn* module.

2.4 Decoder

The decoder is used to generate recommendation results according to the output of the encoder and the prompt about the user's location. To model user's richer geographic context, we propose

neighbor-aware attention, which additionally utilizes user's surrounding location information. The neighbor-aware attention together with the semantic ID is then sent to the decoder to compute the next token prediction loss.

2.4.1 Neighbor-aware Prompt. In local life service, it is vital to model location contexts surrounding the user, such as surrounding brands, bestselling products, etc. Thus, we use cross-attention to capture the surrounding information. Specifically, given the user's real-time geographic location l_u, we calculate the surrounding geographic location $\{l_u^1, l_u^2, ..., l_u^8\}$ and obtain their context information. Thus, we can obtain their context embedding e_u^{lc} and $E^s = \{e_{l_u^1}^{lc}, ..., e_{l_u^8}^{lc}\}$. After that, we calculate cross attention using e_u^{lc} as query and E^s as keys and values:

$$e^s = CrossAttn(e_u^{lc}, E^s), \tag{15}$$

where E^s is the context embedding set of surrounding locations, e_u^{lc} is the context embedding of the user's real-time location, e^s is the embedding of neighbor-aware prompt, $CrossAttn(e_u^{lc}, E^s)$ calculates cross-attention using e_u^{lc} as query and E^s as key and value.

2.4.2 Decoder architecture. The decoder stacks K blocks, each of which contains a casual self-attention module, a cross-attention module and a feed forward network (*FFN*) module with *RMSNorm*. Formally, to generate the semantic ID of the target video, the calculation can be represented as follows:

$$\widetilde{H}^{i+1} = H^i + SelfAttn(RMSNorm(H^i)), \tag{16}$$

$$\widetilde{H}^{i+1} = \widetilde{H}^{i+1} + CrossAttn(RMSNorm(\widetilde{H}^{i+1}), S^K), \tag{17}$$

$$H^{i+1} = \widetilde{H}^{i+1} + FFN(RMSNorm(\widetilde{H}^{i+1})), \tag{18}$$

$$H^0 = \{e^s, e_{q_t^1}, e_{q_t^2}, e_{q_t^3}\}, \tag{19}$$

where H^i is the output of the i-th decoder layer, $\widetilde{H}^{i+1}$ denote the intermediate results of the $i+1$-th decoder layer, $CrossAttn(a, b)$ calculates cross-attention using a as query and b as key and value, $SelfAttn$ and FFN is the self-attention module and feed forward network module, e^s is the embedding of geo-aware prompt, $e_{q_t^1}$, $e_{q_t^2}$ and $e_{q_t^3}$ are the embedding of three-digit semantic ID of the target video.

The output H^K of the decoder is used to predict the next token. Specifically, the output embedding H_0^K of geo-aware prompt is used to predict the first digit of the target semantic ID q_t^1. The output embedding H_1^K of q_t^1 is used to predict the second digit of the target semantic ID q_t^2. Formally, the training loss is the cross-entropy loss:

$$\hat{y}^j = Softmax(MLP(H_{j-1}^K)) \in \mathbb{R}^{N_c}, j \in \{1, 2, 3\} \tag{20}$$

$$L_{ntp} = \sum_{j=1}^{3} -log\hat{y}_{q_t^j}^j, \tag{21}$$

where $\hat{y}_t^i \in \mathbb{R}^{N_c}$ is the predicted probability distribution of the i-th digit of the target semantic ID, $\hat{y}_{q_t^i}^i$ is the predicted probability of q_t^i.

After training a certain number of samples, we obtain a pre-trained model with parameters θ_0. Using the interacted sequence S as input of the encoder and the real-time location l_u as input of the decoder, the model outputs the probability of semantic ID. Formally, $p_{\theta_0}(Q_t|S, l_u) = p_{\theta_0}(q_t^1|S, l_u)p_{\theta_0}(q_t^2|S, l_u, q_t^1)p_{\theta_0}(q_t^3|S, l_u, q_t^1, q_t^2)$.

2.5 Reinforcement Learning

In the pre-training phase above, the model only fits the exposed videos through next token prediction, which are obtained from the traditional recommendation system. Although the exposed videos satisfy multiple objectives to some degree, it is hard to balance multiple objectives in a fine-grained manner if only aligning with exposed videos. To solve this challenge, we introduce two rewards tailored for our scenario, i.e., geographic reward and GMV reward. After that, we use direct preference optimization (DPO) for alignment.

2.5.1 Reward Signals. The geographic reward signal is proposed to encourage generating videos nearby and thus a closer video would get a higher reward:

$$R^{geo}(v, l_u) = \begin{cases} 0 & \text{if } Dis(v, l_u) > D, \\ \frac{1}{Dis(v, l_u)} & \text{else .} \end{cases} \tag{22}$$

where $Dis(v, l_u)$ operation would calculate the distance between video v and the user's location l_u, D is the distance threshold.

The GMV reward signal is proposed to encourage the generation of videos that attract consumption. To achieve this goal, we use a traditional GMV scoring model as the reward model:

$$R^{gmv}(v, S, l_u) = GMV(v, S, l_u), \tag{23}$$

where GMV is a GMV scoring model.

2.5.2 Direct Preference Optimization. To construct preference pairs for direct preference optimization, we first sample several videos from the distribution of the pre-trained model p_{θ_0}. Specifically, we generate N different videos for each sample (S, l_u) by beam search:

$$\mathcal{B}_u^N = TopN(p_{\theta_0}(S, l_u)), \tag{24}$$

where $TopN$ operation would select N videos with the highest probability, $\mathcal{B}_u^N$ is the generated results.

Then we calculate the reward for each of the results through $R^{geo}(v, l_u)$ and $R^{gmv}(v, S, l_u)$. After that, we construct the preference pairs $D_{pairs} = (v_p, v_n, S, l_u)$ by choosing the video v_p with highest reward as the positive sample and the video v_n with lowest reward as the negative sample. Given the preference pairs, we can now train a new model with parameters θ_{i+1}, which is initialized from θ_i. The loss corresponding to each preference pair is as follows:

$$L_{dpo} = -log(\beta log\frac{p_{\theta_{i+1}}(Q_p|S, l_u)}{p_{\theta_i}(Q_p|S, l_u)} - \beta log\frac{p_{\theta_{i+1}}(Q_n|S, l_u)}{p_{\theta_i}(Q_n|S, l_u)}), \tag{25}$$

where $Q_p = Tokenize(v_p), Q_n = Tokenize(v_n)$ is the semantic ID of the positive video and the negative video, respectively.

The training loss in reinforcement learning is:

$$L = L_{ntp} + \lambda L_{dpo}. \tag{26}$$

Table 1: Recommendation performance improvement on different datasets in terms of Recall and NDCG. The best and second-best results are highlighted in bold font and underlined. The superscript * indicates the improvement is statistically significant where the p-value is less than 0.05.

Dataset	Metric	Traditional					POI Related		Generative			Improvement
		SASRec	BERT4Rec	GRU4Rec	Caser	S^3-Rec	TPG	Rotan	TIGER	GNPR-SID	Ours	
KuaiLLSR	Recall@5	0.0927	0.0682	0.0350	0.0438	0.1218	0.1750	0.2185	0.2832	0.3142	**0.3565***	13.46%
	Recall@10	0.1336	0.1071	0.0602	0.0712	0.1889	0.2559	0.2843	0.3637	0.4207	**0.4563***	8.46%
	Recall@20	0.2048	0.1623	0.1090	0.1147	0.2808	0.3241	0.3563	0.4413	0.5056	**0.5584***	10.44%
	NDCG@5	0.0408	0.0311	0.0178	0.0221	0.0793	0.0897	0.1029	0.1500	0.1775	**0.2032***	14.47%
	NDCG@10	0.0523	0.0338	0.0255	0.0266	0.0971	0.1018	0.1147	0.1584	0.1874	**0.2114***	12.81%
	NDCG@20	0.0705	0.0565	0.0360	0.0369	0.1143	0.1117	0.1266	0.1615	0.1904	**0.2151***	12.97%
NYC	Recall@5	0.3151	0.2857	0.1977	0.2883	0.3071	0.3551	0.4448	0.4965	0.5311	**0.6107***	14.98%
	Recall@10	0.3896	0.3564	0.2460	0.3570	0.3854	0.4441	0.5223	0.5514	0.5942	**0.6563***	10.45%
	Recall@20	0.4506	0.4130	0.2889	0.4135	0.4503	0.5121	0.5834	0.6001	0.6455	**0.6977***	8.09%
	NDCG@5	0.2224	0.2074	0.1442	0.2044	0.2235	0.2464	0.3471	0.4131	0.4430	**0.5355***	20.88%
	NDCG@10	0.2467	0.2304	0.1599	0.2267	0.2489	0.2755	0.3723	0.4276	0.4634	**0.5504***	18.77%
	NDCG@20	0.2622	0.2448	0.1708	0.2410	0.2654	0.2927	0.3878	0.4443	0.4766	**0.5608***	17.66%
TKY	Recall@5	0.3450	0.2649	0.2514	0.3257	0.3365	0.3725	0.4333	0.5031	0.5354	**0.5964***	11.39%
	Recall@10	0.4284	0.3326	0.3106	0.4067	0.4115	0.4601	0.5113	0.5808	0.6130	**0.6620***	7.99%
	Recall@20	0.4976	0.3943	0.3651	0.4758	0.4739	0.5291	0.5894	0.6431	0.6675	**0.7152***	7.15%
	NDCG@5	0.2384	0.1907	0.1833	0.2273	0.2423	0.2591	0.3293	0.4003	0.4437	**0.4961***	11.81%
	NDCG@10	0.2655	0.2127	0.2025	0.2535	0.2666	0.2881	0.3568	0.4251	0.4623	**0.5174***	11.92%
	NDCG@20	0.2831	0.2284	0.2163	0.2711	0.2825	0.3051	0.3739	0.4401	0.4788	**0.5306***	10.82%

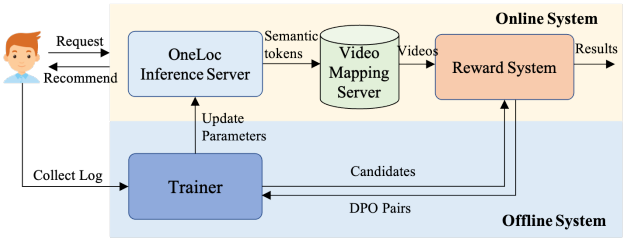

Figure 3: Framework of System Deployment

3 System Deployment

OneLoc has been successfully implemented in real-world local life service scenario of Kuaishou. As illustrated in Fig.3, our deployment architecture consists of several core components: (1) Trainer, (2) OneLoc Inference Server, (3) Video Mapping Server, (4) Reward System. In the offline training phase, the Trainer collects user logs for streaming training, requests the Reward System to score and construct positive/negative sample pairs for RL training, and periodically updates parameters to the inference server during the process. The OneLoc inference server processes user requests by converting user features and real-time geographic locations into user tokens and geo prompts, which are then fed into the model for semantic token generation via beam search. The Video Mapping Server serves as a storage service that maps semantic tokens to video IDs. The candidate videos are subsequently processed by the Reward System through GMV score estimation and rule-based filtering, with the TopK results ultimately being recommended to users. For inference performance optimization, we implement mixed-precision computation, KV cache, dynamic batching, and TensorRT acceleration on NVIDIA A10 GPUs, achieving 25% Model FLOPs Utilization (MFU).

4 Experiments

In this section, we conduct extensive experiments on Kuaishou online platform to answer the following research questions:

- **RQ1**: How does our proposed OneLoc perform compared to the state-of-the-art methods?
- **RQ2**: How do the components of OneLoc (e.g., neighbor-aware prompt and geo-aware self-attention) affect the performance?
- **RQ3**: How does OneLoc perform under different hyperparameters (i.e., model size, sequence length, loss weight λ)?

4.1 Experimental Setting

Table 2: Dataset statistics.

	KuaiLLSR	NYC	TKY
#users	60 M	1,042	2,187
#items	900 K	36,359	59,472
#interactions	440 M	212,347	545,301

4.1.1 Datasets. The Kuaishou platform facilitates daily engagement of hundreds of millions of users with short videos about local life service, generating over hundreds of millions interactions. We construct the **KuaiLLSR** Dataset from this behavioral data for model training and evaluation. Specifically, the KuaiLLSR dataset contains eight days of uniformly sampled local life service short video interaction records, comprising 60 million users, 900K items, and 440 million interactions, with an average user sequence length of around 200. We train OneLoc in a streaming setup where the first 7 days' data are used for model training and the final day's data is reserved for evaluation. In addition, we conduct experiments on the publicly available **Foursquare** dataset, which contains user

check-in records across various points of interest (POIs) in multiple cities. Each record includes the user ID, POI ID, geographical coordinates, and timestamp. We adopt LibCity's [19] standardized pre-processing pipeline to filter users and POIs, and then sort each user's check-ins chronologically to construct interaction sequences. Following common practice, we adopt the NYC and Tokyo subsets, which differ in user density and POI distribution, allowing us to evaluate model performance under diverse spatial–temporal patterns. For these public datasets, we split the interaction sequences into 80% for training, 10% for validation, and 10% for testing. Table 2 gives a rough sketch of the statistics of the five datasets.

4.1.2 Baselines. We compare OneLoc with competitive baselines within two groups of work, traditional recommender models and generative recommender models: 1) **SASRec** [7] employs a unidirectional Transformer encoder that effectively models user preference through self-attention mechanisms. 2) **BERT4Rec** [16] leverages BERT's pre-trained language representations to encoder user behavior sequences. 3) **GRU4Rec** [4] employs gated recurrent units to capture temporal dependencies within user interaction sequences for session-based recommendation tasks. 4) **S^3-Rec** [29] mines self-supervised learning with mutual information maximization for extracting inherent correlations within user behavior sequences. 5) **TPG** [12] is a transformer-based approach that leverages target timestamps as prompts to enhance geography-aware location recommendations. 6) **Rotan** [2] encodes time intervals through rotational position vector representations in transformer architectures, effectively capturing temporal dynamics in user behavior sequences. 7) **TIGER** [15] pioneers a codebook-based semantic quantization framework through RQ-VAE, generating discrete code sequences as identifiers. 8) **GNPR-SID** [17] transforms POI information into discrete semantic identifiers and employs a generative approach for next-POI prediction.

4.1.3 Evaluation Metrics. Following most prior works, we use two metrics for offline experiments: top-K Recall (Recall@K) and NDCG (NDCG@K). For the online A/B experiments, we use GMV and order quantity as core evaluation metrics.

4.1.4 Implement Details. We train OneLoc using the AdamW optimizer (initial learning rate: 2×10^{-4}, weight decay: 0.1) on NVIDIA A800 GPUs. Each codebook layer uses K = 8192 clusters for semantic identifier clustering, with a total of L = 3 codebook layers. Both the encoder and the decoder stack 4 blocks with 1024 hidden units, 8 attention heads, and the dimension of the feedforward network (FFN) is 4096. The lengths of watch sequence, click sequence, and pay sequence are 256, 32, and 10 respectively. The DPO loss weight is set to 0.05.

4.2 Overall Performance (RQ1)

To demonstrate the effectiveness of our method, we conducted experiments on two industry datasets (Foursquare and Kuaishou). We compare our method with the state-of-the-art recommendation methods and the results are demonstrated in Table.1. From the result, we have the following observation:

- (1) We observe that OneLoc consistently surpasses existing baselines in performance. Notably, on the KuaiLLSR datasets, OneLoc achieves multi-scale improvements—including a 13.46% increase in Recall@5, a 10.44% increase in Recall@20, a 14.47% increase in NDCG@5, and a 12.97% increase in NDCG@20, while also demonstrating superiority over the second-best baseline. Similarly, on the Foursquare global dataset, it attains an average performance boost of 13.18% in Recall@5 and 16.34% in NDCG@5. These improvements underscore the efficacy of our proposed integrates geographic generative architecture in the Local Life Service recommendation task. This is attributed to our novel approach, which leverages a dedicated geographic information module. The module integrates users' historical behavioral preferences with real-time spatial context to generate optimized recommendations.
- (2) All generative methods (Ours, TIGER, GNPR-SID) consistently exceed traditional recommender models by more than 29% in Recall@5 and more than 45% in NDCG@5. This improvement demonstrates the comprehensive semantic expression and deep reasoning capabilities of large models. Specifically, it significantly outperforms traditional recall solutions based on representation learning and ANN retrieval.

4.3 Ablation Study (RQ2)

In this section, we conduct an ablation study to investigate whether each component of our framework is effective (Recall and NDCG). We focus on three key modules: (1) neighbor-aware prompt, (2) geo-aware self-attention, and (3) rewards function.

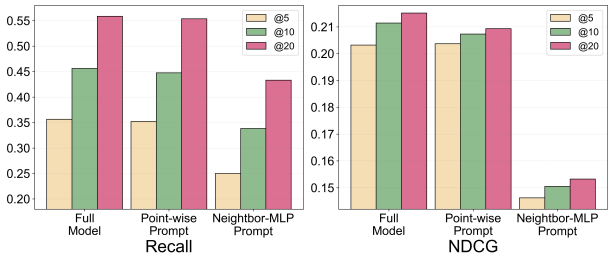

Figure 4: Ablation study of different prompt techniques. The result shows that Neighbor-aware prompt perform significantly better compared to Point-wise prompt and Neighbor-mlp prompt.

Neighbor-aware prompt. Neighbor-aware prompt uses cross attention to capture location context surrounding users. Two questions may arise: (1) whether the surrounding context is effective and (2) whether cross attention is necessary to aggregate surrounding context. To address the first question, we replace the neighbor-aware prompt with a point-wise prompt, which only uses the current location context rather than the surrounding context.

To address the second question, we replace cross attention with naive MLP (neighbor-mlp prompt), which simply concatenates surrounding context embedding and utilizes MLP to aggregate information. The results are illustrated in Fig 4. From the result, we have the following observation: (1) Replacing neighbor-aware prompt with point-wise prompt leads to performance decline, which reveals the effectiveness of the surrounding context. (2) We observe a sharp performance drop when using the neighbor-mlp prompt. The result indicates that the surrounding context may introduce

noise at the same time, and thus, an effective module is essential for capturing useful information within it.

Table 3: Ablation study of key designs in geo-aware self-attention.

Method	Recall			NDCG		
	@5	@10	@20	@5	@10	@20
Full Model	**0.3565**	**0.4563**	**0.5584**	**0.2032**	**0.2114**	**0.2151**
w/o Location Scores	0.3476	0.4439	0.5229	0.1758	0.1847	0.1884
w/o Location Gate	0.3501	0.4489	0.5295	0.1810	0.1914	0.1950
w/o Geo-aware Self-attention	0.3315	0.4261	0.4989	0.1552	0.1640	0.1673

Geo-aware self-attention. Two key designs of geo-aware self-attention are: (1) introducing location context similarity into attention scores (location scores) and (2) leverage user's real-time location as a gate (location gate). To demonstrate the effectiveness of geo-aware self-attention, we carefully design three variants:

- *w/o location scores*: We calculate attention scores without location context scores, i.e., removing the term $E^{lc}(E^{lc})^T$ in Eq 11.
- *w/o location gate*: We remove the location gate, i.e., Eq 14.
- *w/o geo-aware self-attention*: We replace geo-aware self-attention with vanilla self-attention, which can be considered as the combination of *w/o location context scores* and *w/o location gate*.

The results are summarized in Table 3. Compared to the full model, the performance decline of the three variants underscores the critical role of geo-aware self-attention and its key designs.

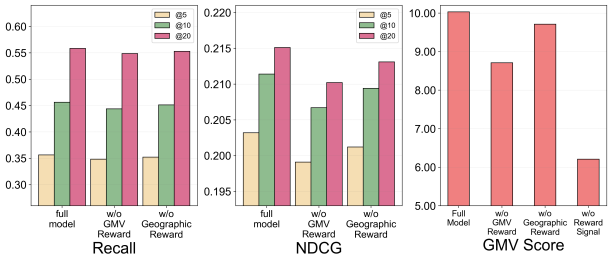

Figure 5: Conduct a comparative analysis of Recall, NDCG, and GMV metrics across varied reward signals.

Reward Signals. In the reinforcement learning, we use two rewards to align with multiple objectives. We remove the geographic reward and the GMV reward separately to investigate their impact. The results are showed in figure 5. Specifically, we remove geographic reward (*w/o geographic reward*) and observe a decrease in recall and NDCG metrics. This indicates that the geographic reward would force the model to consider the distance in spite of user preference, thus boosting the recommendation. In addition, we remove the GMV reward (*w/o GMV reward*) and evaluate the performance using the three metrics (Recall, NDCG and GMV). The result underscores that the GMV reward can not only boost the GMV objective but also benefit the recall objective.

4.4 Hyperparameter Experiments (RQ3)

In this section, we conducted experiments to investigate the impact of each hyperparameters, including model size, sequence length and the DPO loss weight λ. We change the hyperparameters one by one and the results are illustrated in Fig 6. From the results, we have the following observation:

- In terms of model size and sequence length, we observe scaling laws. When scaling the model size from 0.05B (billion) to 0.1B, and then to 0.3B, we consistently observe an increase. When scaling the sequence length, we have the similar observation.
- From 0.05B to 0.3B, we achieve an average improvement of 6.96% and 7.29% in terms of recall and NDCG. When scaling sequence length from 100 to 300, we achieve an average improvement of 13.02% and 51.24% in terms of recall and NDCG.
- In terms of the loss weight, we find that the loss weight is sensitive, i.e., setting $\lambda = 0.01$ is significantly worse than setting $\lambda = 0.05$. Setting $\lambda = 0.03$, both Recall@10 and Recall@20 exhibit significantly lower values compared to $\lambda = 0.05$, whereas the NDCG metric demonstrates superior performance under $\lambda = 0.03$. After a comprehensive evaluation of these trade-offs, we ultimately selected $\lambda = 0.05$ as the final choice for our model.

Table 4: The absolute improvement of OneLoc compared to the production-level multi-stage system in the online A/B test.

Online Metrics	OneLoc
GMV	+21.016%
Number of Orders	+17.891%
Number of Paying Users in Local Services	+18.585%
New Paying Users in Local Services	+23.027%

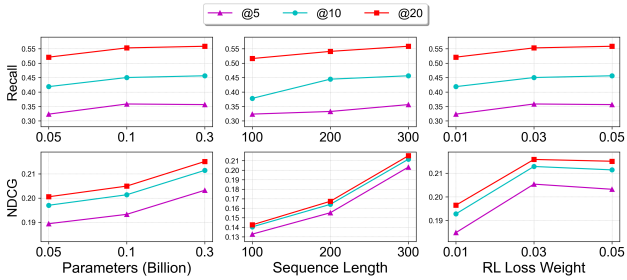

Figure 6: Hyperparameters include model parameters, sequence length and DPO loss weight λ.

5 Online A/B Test

To validate the effectiveness of our proposed model OneLoc, we conducted a one-week online A/B test on Kuaishou's primary short-video recommendation scenario, which serves over 400 million daily active users. In this experiment, 10% of the traffic was allocated to the treatment group using the end-to-end OneLoc system, while the control group retained the production-level multi-stage recommendation pipeline (including multi-channel retrieval, coarse/fine ranking, and link-specific refinements). Both groups shared identical candidate pools and system constraints to ensure fairness.

As shown in Table 4, OneLoc achieves significant improvements in key business objectives with $p < 0.01$ under two-tailed t-tests at

$\alpha = 0.05$: GMV +21.016%, order volume +17.891%, and new local service buyers +23.027%. These results demonstrate that OneLoc is able to surpass complex cascade systems in high-traffic industrial environments, especially in addressing cold-start and data sparsity challenges within local commerce.

6 Related Work

6.1 POI Recommendation

POI recommendation is similar with local life service recommendation (LLSR) since both need geographic information. TPG [12] explicitly uses target times as prompts for a geography- sensitive recommendation. STAN [13] proposes a spatial-temporal attention mechanism to capture spatial-temporal relevance within POI trajectories. Rotan [2] introduces a novel time-aware attention mechanism by representing time intervals as rotational position vectors. Thus, it is effective to capture time information within user behaviors. LLM-Mob [21] introduce in-context learning to enhance next POI recommendation using historical and contextual trajectories NextLocLLM [11] transforms ID prediction to coordinate prediction and injects spatial coordinates into LLM to enhance its understanding of spatial relationships between locations. LLM4POI [9] transforms the next POI recommendation task into a question-answering task to effectively use abundant contextual information in POI recommendation.

In spite of the effectiveness of the above methods, they are all discriminative models, while generative recommendation has emerged as a new paradigm and showcased promising results.

6.2 Generative Recommendation

Generative recommendation has garnered attention from both academia and industry. TIGER [15] is the first work to propose the generative recommendation framework using hierarchical semantic IDs encoded with RQ-VAE. OneRec [1, 28] further utilizes reinforcement learning for user preference learning and unifies the cascading framework with an end-to-end generative framework. OneSug [3] proposes an end-to-end generative framework for e-commerce query suggestion. COBRA [22] proposes a coarse-to-fine framework that first generates semantic IDs and then generates dense vectors for retrieval. LC-Rec [26] aligns with collaborative filtering signals with multiple training tasks. LETTER [20] improves the tokenizer by integrating hierarchical semantics, collaborative signals, and code assignment diversity. ActionPiece [6] argues that the same action may have different meanings depending on its surrounding context and proposes a context-aware tokenizer. EAGER [5] aligns the linguistic semantics of pre-trained LLMs and the collaborative semantics in a non-intrusive manner. Recently, GNPR-SID [17] migrates semantic-ID-baesd generative recommendation to the POI recommendation scenario.

However, prior generative recommendation work has not sufficiently explored the use of geographic information and the modeling of business goals. Thus, they are insufficient for a local life service system in the real industry scenario.

7 Conclusion

In this paper, we focus on our local life service scenario and propose a generative recommendation framework tailored for this scenario. We propose three core modules to effectively model location information, including geo-aware semantic IDs, geo-aware self-attention, and neighbor-aware prompt. To improve business objectives, we propose a reinforcement learning paradigm with a dual reward function including geographic reward and GMV reward. Offline and online experiments have demonstrated the effectiveness of our method. We also conducted rich experiments to investigate the impact of each component and hyperparameters. Now OneLoc serves 400 million users daily in local life service of Kuaishou App and achieves 21.016% and 17.891% improvements in terms of GMV and orders numbers. In the future, we will continue to explore some promising directions, including the scaling laws of model size and sequence length, and RL methods adapted to local life services.

References

[1] Jiaxin Deng, Shiyao Wang, Kuo Cai, Lejian Ren, Qigen Hu, Weifeng Ding, Qiang Luo, and Guorui Zhou. 2025. Onerec: Unifying retrieve and rank with generative recommender and iterative preference alignment. *arXiv preprint arXiv:2502.18965* (2025).

[2] Shanshan Feng, Feiyu Meng, Lisi Chen, Shuo Shang, and Yew Soon Ong. 2024. Rotan: A rotation-based temporal attention network for time-specific next poi recommendation. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*. 759–770.

[3] Xian Guo, Ben Chen, Siyuan Wang, Ying Yang, Chenyi Lei, Yuqing Ding, and Han Li. 2025. OneSug: The Unified End-to-End Generative Framework for E-commerce Query Suggestion. *arXiv preprint arXiv:2506.06913* (2025).

[4] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[5] Minjie Hong, Yan Xia, Zehan Wang, Jieming Zhu, Ye Wang, Sihang Cai, Xiaoda Yang, Quanyu Dai, Zhenhua Dong, Zhimeng Zhang, et al. 2025. EAGER-LLM: Enhancing Large Language Models as Recommenders through Exogenous Behavior-Semantic Integration. In *Proceedings of the ACM on Web Conference 2025*. 2754–2762.

[6] Yupeng Hou, Jianmo Ni, Zhankui He, Noveen Sachdeva, Wang-Cheng Kang, Ed H Chi, Julian McAuley, and Derek Zhiyuan Cheng. 2025. ActionPiece: Contextually Tokenizing Action Sequences for Generative Recommendation. *arXiv preprint arXiv:2502.13581* (2025).

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

[8] Xiaochong Lan, Jie Feng, Jiahuan Lei, Xinlei Shi, and Yong Li. 2025. Benchmarking and Advancing Large Language Models for Local Life Services. *arXiv preprint arXiv:2506.02720* (2025).

[9] Peibo Li, Maarten de Rijke, Hao Xue, Shuang Ao, Yang Song, and Flora D Salim. 2024. Large language models for next point-of-interest recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1463–1472.

[10] Enze Liu, Bowen Zheng, Cheng Ling, Lantao Hu, Han Li, and Wayne Xin Zhao. 2024. End-to-End Learnable Item Tokenization for Generative Recommendation. *arXiv preprint arXiv:2409.05546* (2024).

[11] Shuai Liu, Ning Cao, Yile Chen, Yue Jiang, and Gao Cong. 2024. nextlocllm: next location prediction using LLMs. *arXiv preprint arXiv:2410.09129* (2024).

[12] Yan Luo, Haoyi Duan, Ye Liu, and Fu-Lai Chung. 2023. Timestamps as prompts for geography-aware location recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1697–1706.

[13] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. Stan: Spatio-temporal attention network for next location recommendation. In *Proceedings of the web conference 2021*. 2177–2185.

[14] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, and etc. Sam Altman. 2024. GPT-4 Technical Report. (2024). arXiv:2303.08774 [cs.CL]

[15] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2023), 10299–10315.

[16] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[17] Dongsheng Wang, Yuxi Huang, Shen Gao, Yifan Wang, Chengrui Huang, and Shuo Shang. 2025. Generative Next POI Recommendation with Semantic ID.

arXiv preprint arXiv:2506.01375 (2025).

[18] Guoquan Wang, Qiang Luo, Weisong Hu, Pengfei Yao, Wencong Zeng, Guorui Zhou, and Kun Gai. 2025. FIM: Frequency-Aware Multi-View Interest Modeling for Local-Life Service Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1748–1757.

[19] Jingyuan Wang, Jiawei Jiang, Wenjun Jiang, Chao Li, and Wayne Xin Zhao. 2021. Libcity: An open library for traffic prediction. In *Proceedings of the 29th international conference on advances in geographic information systems*. 145–148.

[20] Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. Learnable item tokenization for generative recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 2400–2409.

[21] Xinglei Wang, Meng Fang, Zichao Zeng, and Tao Cheng. 2023. Where would i go next? large language models as human mobility predictors. *arXiv preprint arXiv:2308.15197* (2023).

[22] Yuhao Yang, Zhi Ji, Zhaopeng Li, Yi Li, Zhonglin Mo, Yue Ding, Kai Chen, Zijian Zhang, Jie Li, Shuanglong Li, et al. 2025. Sparse meets dense: Unified generative recommendations with cascaded sparse-dense representations. *arXiv preprint arXiv:2503.02453* (2025).

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

[24] Jianyang Zhai, Zi-Feng Mai, Chang-Dong Wang, Feidiao Yang, Xiawu Zheng, Hui Li, and Yonghong Tian. 2025. Multimodal Quantitative Language for Generative Recommendation. *arXiv preprint arXiv:2504.05314* (2025).

[25] Qianru Zhang, Peng Yang, Junliang Yu, Haixin Wang, Xingwei He, Siu-Ming Yiu, and Hongzhi Yin. 2025. A survey on point-of-interest recommendation: Models, architectures, and security. *IEEE Transactions on Knowledge and Data Engineering* (2025).

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

[27] Zuowu Zheng, Ze Wang, Fan Yang, Jiangke Fan, Teng Zhang, and Xingxing Wang. 2025. EGA: A Unified End-to-End Generative Framework for Industrial Advertising Systems. *arXiv preprint arXiv:2505.17549* (2025).

[28] Guorui Zhou, Jiaxin Deng, Jinghao Zhang, Kuo Cai, Lejian Ren, Qiang Luo, Qianqian Wang, Qigen Hu, Rui Huang, Shiyao Wang, et al. 2025. OneRec Technical Report. *arXiv preprint arXiv:2506.13695* (2025).

[29] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.