SPOT-Trip: Dual-Preference Driven Out-of-Town Trip Recommendation

Yinghui Liu[1]*, **Hao Miao**[2]*, **Guojiang Shen**[1], **Yan Zhao**[3], **Xiangjie Kong**[1]†, **Ivan Lee**[4]
[1]Zhejiang Key Laboratory of Visual Information Intelligent Processing,
College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, China
[2]Department of Computing, Hong Kong Polytechnic University, Hong Kong, China
[3]Shenzhen Institute for Advanced Study, University of Electronic Science and Technology of China, Shenzhen, China
[4]STEM, University of South Australia, Adelaide, Australia
{2112112249, gjshen1975}@zjut.edu.cn, hao.miao@polyu.edu.hk,
zhaoyan@uestc.edu.cn, xjkong@ieee.org, Ivan.Lee@unisa.edu.au

Abstract

Out-of-town trip recommendation aims to generate a sequence of Points of Interest (POIs) for users traveling from their hometowns to previously unvisited regions based on personalized itineraries, e.g., origin, destination, and trip duration. Modeling the complex user preferences–which often exhibit a two-fold nature of static and dynamic interests–is critical for effective recommendations. However, the sparsity of out-of-town check-in data presents significant challenges in capturing such user preferences. Meanwhile, existing methods often conflate the static and dynamic preferences, resulting in suboptimal performance. In this paper, we for the first time systematically study the problem of out-of-town trip recommendation. A novel framework SPOT-Trip is proposed to explicitly learns the dual static-dynamic user preferences. Specifically, to handle scarce data, we construct a POI attribute knowledge graph to enrich the semantic modeling of users' hometown and out-of-town check-ins, enabling the static preference modeling through attribute relation-aware aggregation. Then, we employ neural ordinary differential equations (ODEs) to capture the continuous evolution of latent dynamic user preferences and innovatively combine a temporal point process to describe the instantaneous probability of each preference behavior. Further, a static-dynamic fusion module is proposed to merge the learned static and dynamic user preferences. Extensive experiments on real data offer insight into the effectiveness of the proposed solutions, showing that SPOT-Trip achieves performance improvement by up to 17.01%[1].

1 Introduction

With the proliferation of location-based social networks (LBSNs), location-based recommendation has become an important means to help people discover attractive and interesting points of interest (POIs) [10, 2], including POI recommendation [52, 22], trip recommendation [14, 23, 54, 39], and out-of-town recommendation [46, 47, 32]. Traditional POI and trip recommender systems are dedicated to recommending POIs within a specific region. However, they may fail when users travel out of their hometown [47]. Consequently, out-of-town recommendation [32] emerges, which generates POIs for users traveling from their hometowns to regions that they have seldom visited previously.

*Equal contribution.

†Corresponding author.

[1]The code of SPOT-Trip can be found at https://github.com/Yinghui-Liu/SPOT-Trip.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

Existing out-of-town recommendation methods often focus on addressing the problem of interest drift [43, 32] and geographical gap [25, 12]. However, these methods are mainly designed for the next POI recommendation (see the left part of Fig. 1) while lacking the capabilities to provide a comprehensive trip itinerary for travelers. In real-world scenarios, given an origin and a destination, users may prefer a model that can generate a sequence of intermediate activities (see the right part of Fig. 1) to realize a more engaging journey for convenience. To achieve this, we for the first time study a new problem, i.e., out-of-town trip recommendation, that can provide consecutive intermediate POIs given the origin, destination, and the number of stops. Such trip recommendation models are expected to enable systematic itinerary generation, facilitating efficient decision-making for the users.

However, *data sparsity* poses a great challenge for out-of-town trip recommendations since users often have few or even no historical check-in records in out-of-town regions. It is hard to obtain a well-performed out-of-town trip recommender model with such scarce data. In addition, it is challenging to learn the *complex user preferences* to alleviate interest drifts (i.e., out-of-town check-ins are not aligned with hometown check-in preferences) for effective out-of-town trip recommendation [46, 47]. Intuitively, the user preferences can be categorized into two complementary components: *static (or invariant)* and *dynamic preferences*. On the one hand, static preference captures long-term user interests and stable behavioral tendencies, which are often extracted from hometown check-in records. Although static preference shows the stable tendencies of a user, directly applying it to out-of-town trip recommendation may be suboptimal due to cold-start and data scarcity. To enable better generalization, it calls for an alignment method to effectively transfer sufficient knowledge learned from the hometown, i.e., static preferences, to the target regions. On the other hand, dynamic preference reflects the short-term behavioral patterns that are sensitive to contextual information, e.g., time, location, and intent.

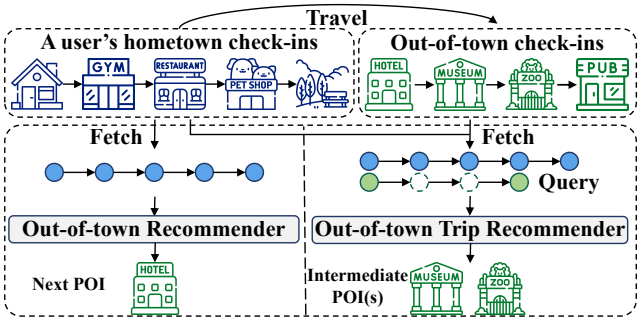


Figure 1: Comparison of two out-of-town recommendation tasks. The blue circles indicate the user's hometown historical check-ins, green circles represent the given query POIs, i.e., the origin and destination of a trip, while the hollow dashed circles denote the intermediate POI(s) to be inferred.

Nonetheless, existing methods [42, 32] often learn the entangled user preferences. We argue that effective disentanglement of the two preferences can facilitate user intent modeling across diverse scenarios and explicitly mitigate the interest drift, thereby enabling personalized recommendation and improving the robustness and effectiveness. Further, it is challenging to fuse the static and dynamic preferences, which enhances trip recommendation by taking preference consistency and personalization into account, simultaneously, enabling more accurate and context-aware user modeling. To this end, we propose a Static-dynamic Preference aware Out-of-Town Trip recommendation framework, SPOT-Trip, to explicitly learn such dual user preferences. SPOT-Trip encompasses three major modules: knowledge-enhanced static preference modeling, ODE-based dynamic preference learning, and static-dynamic preference fusion.

The core idea of SPOT-Trip lies in jointly modeling sequence-level static preferences with semantics-enhanced representations and POI-level dynamic preferences by an ODE. First, we propose a knowledge-enhanced static preference learning module. In particular, we construct a POI attribute knowledge graph based on user check-ins to incorporate rich semantic relations, e.g., a *hasRating* relation between POIs and attribute entities such as *5-star*. Next, a relation-aware attention aggregation mechanism is designed to generate enriched POI embeddings, which capture the entity semantics with diverse relational contexts, alleviating the data sparsity. Further, we propose a novel static preference alignment mechanism to transfer knowledge of the static preferences learned from hometown check-ins to the out-of-town region. The static preferences are learned by a static aggregator, which aggregates the enriched POI embeddings at the sequence level. Second, we develop an ODE-based dynamic preference learning module to model the continuous and irregular evolution based on the dynamic user behaviors during out-of-town trips. A temporal point process [21] is further employed to characterize the probability of preference behaviors over time, enabling dynamic preference

inference. Notably, we incorporate auxiliary geographical coordinate information to refine the behavior representation. Finally, a static-dynamic preference fusion module is proposed to fuse the learned static and dynamic user preferences for effective out-of-town trip recommendation.

The major contributions of the work are as follows: (1) *New Task*. To the best of our knowledge, this is the first systematic study to learn out-of-town trip recommendation. We propose a framework called SPOT-Trip to explicitly capture the static and dynamic user preferences for effective out-of-town trip recommendation. (2) *Novel Techniques*. An innovative static preference learning module is proposed, which leverages sufficient semantic knowledge to extract stable user preferences. We extract the dynamic user preference based on neural ODE in conjunction with a novel temporal point process, which can capture the continuous preference drift. (3) *Superior Performance*. We report on extensive experiments using real data, offering evidence of the effectiveness of the proposals.

2 Preliminary

Definition 1: (POI Attribute Knowledge Graph). The POI attribute knowledge graph is defined as $\mathcal{G}_k = (v, r, e)$, which encodes a semantic relation r between a POI v and an entity e. It captures external knowledge by incorporating diverse types of attribute entities and their relationships with POIs, such as (*Balboa Park*, *Located in*, *San Diego*).

Definition 2: (Check-in). A user check-in is denoted as a tuple $c = (u, t, l, v)$, indicating that user u visited POI v at time t and location l, where l includes the geographic coordinates, i.e., latitude and longitude.

Definition 3: (Out-of-town Travel Behavior). Given a user u, the out-of-town travel behavior is denoted as $\xi = \left(u, \vec{c}_h, \vec{c}_o, a_h, a_o\right)$, indicating that u departs from his/her hometown a_h to visit an out-of-town region a_o with check-in records in both regions, i.e., $\vec{c}_h$ (hometown) and $\vec{c}_o$ (out-of-town), respectively. We use $M = |\vec{c}_h|$ and $N = |\vec{c}_o|$ to represent the number of check-ins in the hometown and out-of-town regions, respectively, where $M > N$.

Problem Statement: (Out-of-town Trip Recommendation). Given a set of users $\mathcal{U} = \{u_i\}_{i=1}^{|\mathcal{U}|}$, POIs $\mathcal{V} = \{v_i\}_{i=1}^{|\mathcal{V}|}$, regions $\mathcal{A} = \{a_i\}_{i=1}^{|\mathcal{A}|}$, and out-of-town trip records $\mathcal{O} = \{\xi_i\}_{i=1}^{|\mathcal{O}|}$, our objective is to learn a recommender function $\mathcal{F}$ based on historical records $\mathcal{O}$ and the POI attribute knowledge graph $\mathcal{G}_k$. For a user $u^* \notin \mathcal{U}$ at a_h^* with hometown check-ins $\vec{c}_h$ and out-of-town origin-destination trip query $Q_o^* = \{v_s^o, v_e^o, N\}$ in region a_o^*, where v_s^o, v_e^o, and N denote the start, end points and the number of stops, the learned $\mathcal{F}$ generates a sequence of POIs $\tau = \{v_1^o, v_2^o, \ldots, v_N^o\}$, where $v_1^o = v_s^o$ and $v_N^o = v_e^o$ for u^*. The recommended POIs in τ are in a_o^*.

3 Methodology

We proceed to detail the dual preference-aware out-of-town trip recommendation framework, SPOT-Trip. We first give an overview of the framework and then provide specifics. As illustrated in Fig. 2, SPOT-Trip encompasses three major modules: Knowledge-Enhanced Static Preference Learning (*KSPL*), ODE-Based Dynamic Preference Learning (*ODPL*), and Static-Dynamic Preference Fusion.

KSPL aims to enhance the static semantic awareness of user preference by means of knowledge graph construction, which consists of two components. First, semantic knowledge aggregation is proposed to improve POI embeddings with a relational attention mechanism based on the constructed POI attribute knowledge graph. Second, we design an innovative static preference alignment component that facilitates the transfer of static hometown preferences to the out-of-town region. *ODPL* is dedicated to capturing the evolving nature of user preferences with a novel ODE-based continuous dynamics modeling. In this module, we first aggregate the hometown check-in sequence with spatiotemporal information and feed the resulting sequence into a Transformer to generate initial latent representations, which are then input into a neural ODE to capture dynamics. Subsequently, the check-in behaviors are further interpreted by a temporal point process to characterize the instantaneous probability of each behavior. Finally, we propose a static-dynamic preference fusion module to merge the learned static and dynamic user preferences.

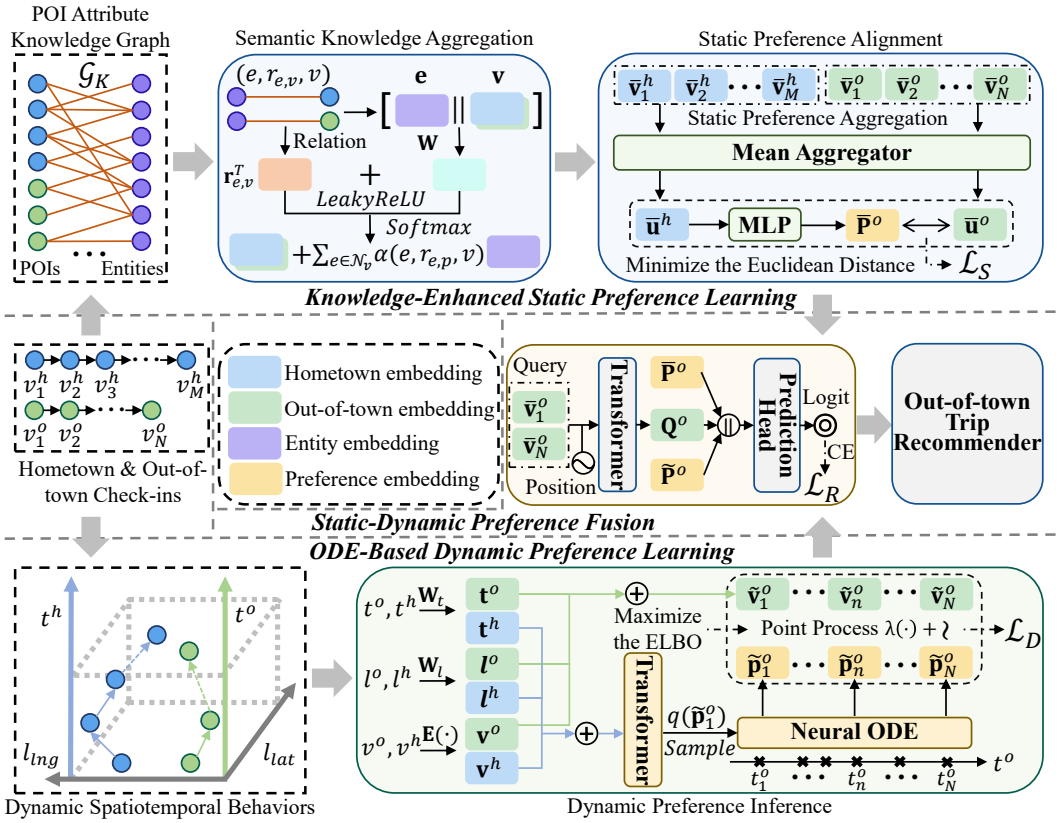


Figure 2: The framework of SPOT-Trip. CE denotes the cross-entropy loss.

3.1 Knowledge-Enhanced Static Preference Learning

Traditional out-of-town recommendation methods have attempted to model static preference with transition relationships [46] and geographic relationships [47]. However, they remain hampered by data sparsity and fail to align user preferences across different regions. Knowledge graphs (KGs) [5, 55] enrich POI embeddings using additional semantic information, thereby strengthening user preference representations. Yet, directly integrating semantic and spatial information may lead to conflicting interactions [32]. Accordingly, we leverage a KG approach solely to model the static user preferences from a semantic perspective. Spatial information will be incorporated in the dynamic learning described in Sec. 3.2.

Semantic Knowledge Aggregation. In the POI attribute knowledge graph, various relationships (e.g., POI's category and corresponding region) encode distinct semantic information. Inspired by the graph attention mechanisms in [50, 32], we first introduce a relation-aware knowledge embedding layer to reflect the heterogeneity of relations over knowledge graph connections and then gain entity- and relation-specific representations through a parameterized attention mechanism. Towards that, we construct our relation-aware message aggregation mechanism between the POI and its connected entities in $\mathcal{G}_K$, for generating knowledge-aware POI embeddings via a heterogeneous attentive aggregator illustrated as follows:

$$\bar{\mathbf{v}} = \mathbf{v} + \sum_{e \in \mathcal{N}_v} \alpha(e, r_{e,v}, v)\mathbf{e}, \alpha(e, r_{e,v}, v) = \frac{\exp(\phi(\mathbf{r}_{e,v}^T \mathbf{W}[\mathbf{e} \parallel \mathbf{v}]))}{\sum_{e \in \mathcal{N}_v} \exp(\phi(\mathbf{r}_{e,v}^T \mathbf{W}[\mathbf{e} \parallel \mathbf{v}]))}, \tag{1}$$

where $\mathcal{N}_v$ is the neighboring entities of POI v on various relations $r_{e,v}$ in $\mathcal{G}_K$, $\mathbf{v} \in \mathbb{R}^d$ and $\mathbf{e} \in \mathbb{R}^d$ represent the embedding of POI and entity, respectively. The estimated entity- and relation-dependent attentive relevance during the knowledge aggregation process is denoted as $\alpha(e, r_{e,v}, v)$, which encodes the distinct semantics of relationships between POI v and entity e. $\parallel$ means the concatenation of two embeddings. $\mathbf{W} \in \mathbb{R}^{d \times 2d}$ denotes the weight matrix that is tailored to the input POI and entity representations. ϕ is the activation function *LeakyReLU* for non-linear transformation. Additionally,

to further improve the multi-relational semantic representation space for entity–item dependencies, an alternative training is employed between the relation-aware knowledge aggregator and TransE [4] (More details about the alternative training and its loss can be found in the Appendix A.1).

Static Preference Alignment. Inspired by prior out-of-town recommendation approaches that derive user preferences through behavioral aggregation, we utilize an aggregator $\mathrm{AGG}_S(\cdot)$ (i.e., average pooling) to consolidate the enriched POI embeddings, producing enhanced user hometown and out-of-town static preference representations: $\bar{\mathbf{u}}^h = \mathrm{AGG}_S([\bar{\mathbf{v}}_m^h]_{m=1}^M), \bar{\mathbf{u}}^o = \mathrm{AGG}_S([\bar{\mathbf{v}}_n^o]_{n=1}^N)$. Subsequently, we infer user out-of-town preference by a MLP based on their hometown preferences:

$$\bar{\mathbf{P}}^o = \phi(\mathbf{W}_S \bar{\mathbf{u}}^h + \mathbf{b}_S), \tag{2}$$

where $\mathbf{W}_S \in \mathbb{R}^{d\times d}$ and $\mathbf{b}_S \in \mathbb{R}^d$ are trainable parameters. ϕ denotes the activation function *SiLU*. An Euclidean distance loss $\mathcal{L}_S$ is adopted to bridge the inferred preference and the actual out-of-town preference for static learning as follows:

$$\mathcal{L}_S = \sum_{u\in\mathcal{U}} \left\|\bar{\mathbf{P}}^o - \bar{\mathbf{u}}^o\right\|_2^2 \tag{3}$$

3.2 ODE-Based Dynamic Preference Learning

Previous trip recommenders [23, 39] incorporate position embeddings and periodic hour encodings to model the dynamic nature of user preferences. However, due to the irregular sampling [37] of check-in data, these approaches fail to capture the dynamic evolution of user preferences over actual time. Given the success of neural ODE [8] in other research fields [34, 30], we develop a neural ODE to model the continuous dynamic drift of user out-of-town preferences in latent space. To quantify the behavior event probability at each moment, we also formulate the preference inference as a temporal point process [33], where the instantaneous probability is described by the intensity function $\lambda(\cdot)$ of a non-homogeneous Poisson process (NHPP) on the latent state. Overall, the *Dynamic Preference Inference* process can be defined as:

$$\tilde{\mathbf{p}}_{t_1}^o \sim p\big(\tilde{\mathbf{p}}_{t_1}^o\big), \frac{d\tilde{\mathbf{p}}_t^o}{dt} = f\big(\tilde{\mathbf{p}}_{t_1}^o\big), \tag{4}$$

$$t_n \sim \mathrm{NHPP}(\lambda(\tilde{\mathbf{p}}_t^o)), n = 1, ..., N, \tag{5}$$

$$\tilde{\mathbf{v}}_{t_n}^o \sim p\big(\tilde{\mathbf{v}}_{t_n}^o | \tilde{\mathbf{p}}_{t_n}^o\big), n = 1, ..., N, \tag{6}$$

where $\tilde{\mathbf{p}}_t^o$ represents the latent preference state at time t. $f(\cdot)$ and $\lambda(\cdot)$ are MLPs for parameterizing ODE dynamics and the intensity function. For brevity, we denote $\tilde{\mathbf{p}}_{t_n}^o$ by $\tilde{\mathbf{p}}_n^o$ over the remainder of this paper, and the corresponding joint distribution is:

$$\begin{aligned} p(\tilde{\mathbf{p}}_1^o, \{t_n, \tilde{\mathbf{v}}_n^o\}_{n=1}^N) &= p(\tilde{\mathbf{p}}_1^o) p(\{t_n\}_{n=1}^N | \tilde{\mathbf{p}}_1^o) \prod_{n=1}^N p(\tilde{\mathbf{v}}_n^o | \tilde{\mathbf{p}}_n^o, t_n), p(\tilde{\mathbf{p}}_1^o) = \mathcal{N}(\mathbf{0}, I), \\ p(\{t_n\}_{n=1}^N | \tilde{\mathbf{p}}_1^o) &= \prod_{n=1}^N \lambda(\tilde{\mathbf{p}}_n^o) \exp(-\int \lambda(\tilde{\mathbf{p}}^o) dt), p(\tilde{\mathbf{v}}_n^o | \tilde{\mathbf{p}}_n^o, t_n) = \mathcal{N}(\tilde{\mathbf{p}}_n^o, \sigma_{\tilde{v}^o}^2 I), \end{aligned} \tag{7}$$

where $\mathcal{N}$ is the normal distribution, $\mathbf{0}$ is a zero vector, I is the identity matrix and $\sigma_{\tilde{v}^o}$ denotes a hyper-parameter that sets the model's tolerance for discrepancies between observations and predictions. In the following, the detailed descriptions and inferences of each component are provided.

Latent Dynamics of Preference Drift (Eq. 4). Before modeling the latent ODE dynamics, the user's dynamic spatiotemporal behaviors including timestamps and spatial coordinates are first mapped into the latent space via linear transformations and a new embedding layer $\mathbf{E}(\cdot)$: $\tilde{\mathbf{v}}^{h,o} = \mathbf{W}_t t^{h,o} + \mathbf{W}_l l^{h,o} + \mathbf{E}(v^{h,o})$, where $\mathbf{W}_t \in \mathbb{R}^{1\times d}$ and $\mathbf{W}_l \in \mathbb{R}^{2\times d}$ are trainable parameters. Note that the POIs' spatial features are treated solely as quantitative values because geographic information is unavailable during the recommendation. To better infer the posterior of the latent initial state of preference $\tilde{\mathbf{p}}_1^o$ and model parameters, the variational inference [1] is used for latent modeling. We define an approximate posterior $q(\tilde{\mathbf{p}}_1^o; \psi)$ with variational parameters ψ to approximate the true posterior $p(\tilde{\mathbf{p}}_1^o | \{t_n^o, \tilde{\mathbf{v}}_n^o\}_{n=1}^N)$, with the objective of minimizing the Kullback-Leibler divergence

$$\mathrm{KL}[q(\tilde{\mathbf{p}}_1^o; \psi) || p(\tilde{\mathbf{p}}_1^o | \{t_n^o, \tilde{\mathbf{v}}_n^o\}_{n=1}^N)] \tag{8}$$

over the variational parameters to derive an estimate of the approximation of the posterior and model parameters. To avoid optimizing the local variational parameters ψ for each behavior trajectory in the dataset, we use amortization [1] and define ψ as a reparameterization sampling from the output of a stacked Transformer encoder layer [40] Trans_D. To be specific, we first introduce an aggregation token, $\mathbf{AGG}$, with a learnable representation to indicate Trans_D to aggregate user hometown behaviors. This gives us the encoder input sequence: $\{\{\tilde{\mathbf{v}}_m^h\}_{m=1}^M, \mathbf{AGG}\}$, and then we acquire the output aggregation token $\tilde{\mathbf{AGG}}$ at the last layer. Finally, we obtain ψ from $\tilde{\mathbf{AGG}}$ with the reparameterization sampling dependent on an approximate posterior (see Appendix A.2 for specification of the sampling).

Temporal Point Process (Eq. 5). After receiving the inferred preference $\tilde{\mathbf{p}}_n^o$ from the ODE solver in Eq. 4, we employ the parameterized intensity function $\lambda(\cdot)$ to map it to the intensity of NHPP. In order to ensure that the intensity function value is nonnegative and enhance its numerical stability, we further exponentiate the output of $\lambda(\tilde{\mathbf{p}}_n^o)$ and then add a small constant.

Behavior Distribution Reconstruction (Eq. 6). In practice, rather than minimizing the KL divergence in Eq. 8 directly, we instead maximize the corresponding evidence lower bound (ELBO) for the dynamic behavior distribution reconstruction, and the total dynamic learning loss is defined as:

$$\mathcal{L}_D = -\sum_{u\in\mathcal{U}}\Big(\underbrace{\sum_{n=1}^{N_u}\mathbb{E}_{q(\tilde{\mathbf{p}}_1^o;\psi)}[\ln p(\tilde{\mathbf{v}}_n^o|t_n^o,\tilde{\mathbf{p}}_1^o)]}_{(i)\text{ Expected restructured behavior log-lik}} + \underbrace{\mathbb{E}_{q(\tilde{\mathbf{p}}_1^o;\psi)}\sum_{n=1}^{N_u}\ln\lambda(\tilde{\mathbf{p}}_n^o) - \int\lambda(\tilde{\mathbf{p}}^o)dt}_{(ii)\text{ Expected temporal point process log-lik}} - \underbrace{\mathrm{KL}[q(\tilde{\mathbf{p}}_1^o;\psi)||p(\tilde{\mathbf{p}}_1^o)]}_{(iii)\text{ KL between prior and posterior}}\Big), \tag{9}$$

where the term (iii) can be computed analytically, computation of terms (i) and (ii) involves approximations: term (i) is estimated via Monte Carlo integration using samples from the variational posterior, whereas term (ii) involves solving an ODE with a numerical solver and applying the trapezoidal rule to approximate the expected intensity integral. The ELBO is maximized w.r.t. the model parameters and its detailed derivation is contained in Appendix A.3.

3.3 Static-Dynamic Preference Fusion

To generate the logit $z_{u,n,i}$ for user u at position n over each POI $i \in a_o$, we first construct a unified representation by combining the user's knowledge-enhanced query representation $\mathbf{Q}^o \in \mathbb{R}^{2d\times d}$, the static preference $\bar{\mathbf{P}}^o$ inferred in Sec. 3.1, and the dynamic preference sequence $\tilde{\mathbf{P}}^o = \{\tilde{\mathbf{p}}_n^o\}_{n=1}^N$ inferred in Sec. 3.2. This joint representation is then fed into a nonlinear prediction head to compute the logits for all POIs at each position n in the out-of-town region a_o.

$$z_{u,n,i} = \phi(\mathbf{W}_R[\mathbf{Q}^o||\bar{\mathbf{P}}^o||\tilde{\mathbf{P}}^o] + \mathbf{b}_R), \tag{10}$$

where $\mathbf{W}_R \in \mathbb{R}^{4d\times d}$, $\mathbf{b}_R \in \mathbb{R}^d$ are trainable parameters and σ is the activation function *LeakyReLU*. Appendix A.3 contains more details about the acquisition of query representation $\mathbf{Q}^o$. Consistent with previous works [14, 39], the main recommendation loss is formulated as a cross-entropy loss:

$$\mathcal{L}_R = -\frac{1}{\sum_{u\in\mathcal{U}} N_u}\sum_{u=1}^{\mathcal{U}}\sum_{n=1}^{N_u}\log\frac{\exp\big(z_{u,n,v_{u,n}^o}\big)}{\sum_{i\in a_o}\exp\big(z_{u,n,i}\big)}. \tag{11}$$

3.4 Optimization and Recommendation

Optimization. With Eqs. 3, 9 and 11, we can jointly optimize the composite training optimization loss function in an end-to-end fashion as below:

$$\mathcal{L} = \beta_1\mathcal{L}_S + \beta_2\mathcal{L}_D + \beta_3\mathcal{L}_R, \tag{12}$$

where β_1, β_2 and β_3 are the hyper-parameters to balance the effects of the three losses in SPOT-Trip.

Recommendation. When carrying out the out-of-town trip recommendation, our goal is to recommend a sequence of out-of-town POIs to a new user u^* traveling to region a with a clear query. In particular, given u^*'s hometown check-in records and query $Q_o^* = \{v_s^o, v_e^o, N\}$, we first derive her/his

Table 1: The overall comparison between SPOT-Trip and baselines, where the best performance is marked in bold while the second-best results are underlined. * denotes improvements that are statistically significant, where we use two-sided t-test with p-value < 0.05 [31].

Method	Foursquare				Yelp			
	$F_1(\uparrow)$	$PairsF_1(\uparrow)$	$Full\text{-}F_1(\uparrow)$	$Full\text{-}PairsF_1(\uparrow)$	$F_1(\uparrow)$	$PairsF_1(\uparrow)$	$Full\text{-}F_1(\uparrow)$	$Full\text{-}PairsF_1(\uparrow)$
PersTour [26]	0.0258	0.0016	0.4421	0.1572	0.0251	0.0066	0.5059	0.2074
Popularity [6]	0.0261	0.0013	0.4423	0.1565	0.0257	0.0056	0.5065	0.2058
POIRank [6]	0.0253	0.0019	0.4416	0.1582	0.0264	0.0079	0.5068	0.2093
GraphTrip [14]	0.0295	0.0048	0.4498	0.1620	0.0289	0.0126	0.5107	0.2184
MatTrip [54]	0.0311	0.0037	0.4530	0.1656	0.0301	0.0119	0.5117	0.2191
AR-Trip [39]	0.0304	0.0045	0.4512	0.1673	0.0307	0.0153	0.5115	0.2204
Base	0.0339	0.0069	0.4571	0.1698	0.0315	0.0149	0.5097	0.2215
Base + KDDC [32]	<u>0.0375</u>	0.0079	<u>0.4606</u>	0.1822	<u>0.0341</u>	0.0156	<u>0.5126</u>	<u>0.2256</u>
Base + CNN-ODE [21]	0.0367	<u>0.0094</u>	0.4578	<u>0.1843</u>	0.0326	<u>0.0168</u>	0.5124	0.2237
Base + PPROC [21]	0.0330	0.0071	0.4550	0.1687	0.0334	0.0159	0.5110	0.2218
SPOT-Trip	**0.0400***	**0.0109***	**0.4723***	**0.1960***	**0.0399***	**0.0190***	**0.5261***	**0.2347***
Improvement	+6.67%	+15.96%	+2.54%	+6.34%	+17.01%	+13.90%	+2.63%	+4.03%

inferred static preference $\bar{\mathbf{P}}^o$ and dynamic preference $\tilde{\mathbf{P}}^o$ following Eqs. 2 and 4. Then we utilize Eq. 10 to generate each POI logit $z_{u^*,n,i}$ at the trip position n in region a and employ a stochastic sampling method (i.e., Top-p [20]) to recommend the intermediate POI(s) for trip completion. Note that actual timestamps are employed in Sec. 3.2 during training to fully capture the temporal dynamics; however, such precise time information is not available during the recommendation phase. Consequently, we adopt the normalization of each query position n as a surrogate time grid, thereby approximating temporal progression in a computationally efficient manner.

4 Experiment

4.1 Experimental Setups

Dateset. The experiments are carried out on two widely-used travel behavior datasets: Foursquare and Yelp. We follow existing studies [47, 32] to identify users with check-in activities in their hometown and other regions. The check-ins are reorganized to form the corresponding out-of-town travel records $\left(u, \vec{c}_h, \vec{c}_o, a_h, a_o\right)$. To improve the rationality, we filter out users who have fewer than three check-ins in out-of-town regions, or whose travel durations are shorter than 1 hour or longer than 30 days. The knowledge graphs are extracted from the supplementary descriptive information in two datasets. More details regarding datasets can be seen in Appendix B.1.

Baseline. We compare SPOT-Trip with 10 baselines, including 7 established trip recommendation methods without hometown information: PersTour [26], Popularity [6], POIRank [6], GraphTrip [14], MatTrip [54], AR-Trip [39], and Base; and 3 methods with hometown information: Base + KDDC [32], Base + CNN-ODE [21], and Base + PPROC [21], where Base denotes SPOT-Trip without *KSPL* and *ODPL*. More baseline details are provided in Appendix B.2.

Evaluation Metrics. Following existing studies [14, 23, 39], F_1 and $PairsF_1$, are adopted as the evaluation metrics, where higher values indicate better performance. As the out-of-town trip recommendation focuses on intermediate POIs, we calculate F_1 and $PairsF_1$ without the origin and destination. In addition, to enable more comprehensive comparison, we also report the experiment results with origin and destination, denoted as $Full\text{-}F_1$ and $Full\text{-}PairsF_1$. Please see the corresponding formulas in Appendix B.3.

Implementation Details. We implement our model using the Pytorch framework on NVIDIA GeForce RTX 4090 GPU. We perform statistical testing, i.e., three times, with different parameters to enable fairer comparison, where averaged results are reported in our study. The learning rate is set to 0.001 with Adam optimizer. Additionally, the batch size and training epochs are set to 32 and 1000, respectively. We provide more details about implementation in Appendix B.4. Further, the parameters of the baseline methods are set based on their original papers and associated code.

4.2 Overall Performance

We compare SPOT-Trip with 9 baselines, where Table 1 reports the F_1, $PairsF_1$, $Full\text{-}F_1$, and $Full\text{-}PairsF_1$ values. Generally, SPOT-Trip achieves the best results on both datasets across all evaluation metrics, demonstrating its effectiveness. SPOT-Trip performs better than the best among the baselines by up to 17.01% and 15.96% in terms of F_1 and $PairsF_1$, respectively. We also observe that the performance improvements obtained by SPOT-Trip on the Yelp dataset exceed those on Foursquare in terms of intermediate POIs recommendation. This is because the sufficient semantic information of Yelp introduces more external knowledge for model training, enhancing model performance. Additionally, we observe that out-of-town trip recommendation-based methods that use hometown information generally yield higher F_1 and $PairsF_1$ scores than those trip recommendation-based methods without hometown information, suggesting the significance of the static and dynamic user preference modeling with hometown context.

Further, compared to suboptimal baselines equipped with only a single learning module, i.e., Base + KDDC, Base + CNN-ODE and Base + PPROC, SPOT-Trip achieves notable improvements. Specifically, on Foursquare, it improves the F_1 and $PairsF_1$ by 6.67% and 15.96%, respectively, while the corresponding gains are 17.01% and 13.90% on Yelp. Moreover, under full-trip evaluations $Full\text{-}F_1$ and $Full\text{-}PairsF_1$, which inherently include fixed origin and destination and therefore dilute relative gains, SPOT-Trip still consistently outperforms the baselines. These observations demonstrate the superiority of SPOT-Trip due to that it can learn static and dynamic user preferences comprehensively. It is worth noting that Base + PPROC exhibits inferior performance compared to Base + CNN-ODE, potentially because of the complexity involved in predicting spatiotemporal event points, which largely depends on the design of the underlying predictive model.

4.3 Ablation Study

To gain insight into the contributions of the different components of SPOT-Trip, we evaluate three variants: (1) **w/o KS**. SPOT-Trip without the *KSPL* module; (2) **w/o OD**. SPOT-Trip without the *ODPL* module; (3) **w/o SI**. SPOT-Trip without the spatial information in *ODPL*. Figure 3 shows the results on two datasets, SPOT-Trip outperforms its counterparts with the *KSPL* module, *ODPL* module, and spatial information. This shows that these three components are useful for effective out-of-town trip recommendations. In particular, the **w/o KS** variant shows a 10.02% drop in F_1 and a 21.79% drop in $PairsF_1$ on Yelp, validating the effectiveness of our framework in modeling users' static preferences. Furthermore, the **w/o SI** variant suffers significant performance degradation, especially in terms of $PairsF_1$ on Foursquare and F_1 on Yelp, demonstrating the necessity of incorporating spatial information into the static preference learning process. More results and discussions are provided in Appendix C.1.

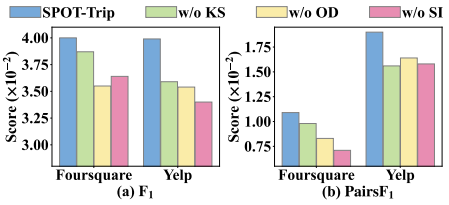


Figure 3: Performance of SPOT-Trip and its variants on two datasets.

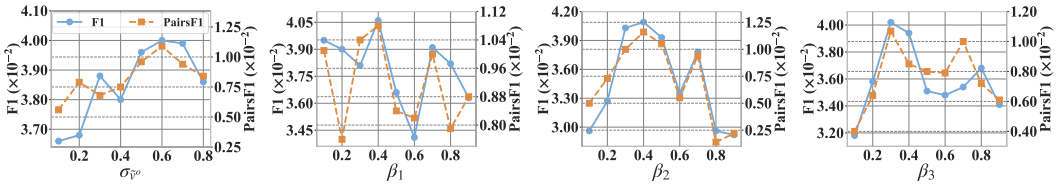


Figure 4: The effects of discrepancy tolerance parameter ($\sigma_{\tilde{v}^o}$) and various loss function weights (β_1, β_2 and β_3) on the Foursquare dataset w.r.t. the F_1 and $PairsF_1$ score.

4.4 Hyper-parameter Analysis

We study how sensitive the framework is to the tolerance parameter $\sigma_{\tilde{v}^o}$ and loss function weights β_1, β_2 and β_3. The results are reported in Fig. 4. Specifically, $\sigma_{\tilde{v}^o}$ controls the standard deviation of the behavior noise. The results show that both F_1 and $PairsF_1$ first show an increasing trend with

Table 2: The effect of data sparsity. We sample different fractions of the training data.

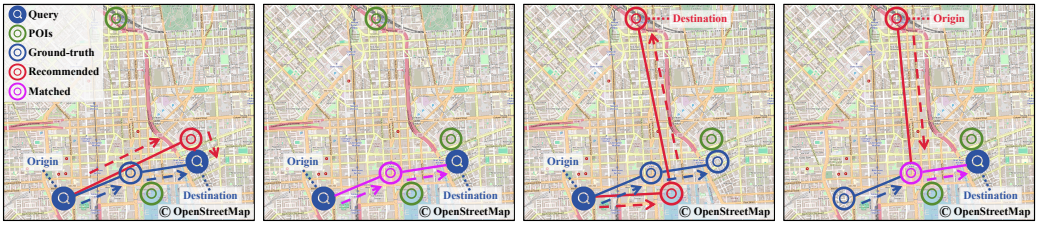
Dataset	Method	Hometown Data during Training							
		40%		60%		80%		100%	
		$F_1(\uparrow)$	$PairsF_1(\uparrow)$	$F_1(\uparrow)$	$PairsF_1(\uparrow)$	$F_1(\uparrow)$	$PairsF_1(\uparrow)$	$F_1(\uparrow)$	$PairsF_1(\uparrow)$
Foursquare	Base + KDDC	0.0325	0.0038	0.0348	0.0058	0.0364	0.0069	0.0375	0.0079
	Base + CNN-ODE	0.0314	0.0046	0.0336	0.0052	0.0352	0.0074	0.0367	0.0094
	SPOT-Trip	**0.0347**	**0.0073**	**0.0369**	**0.0085**	**0.0401**	**0.0091**	**0.0400**	**0.0109**
Yelp	Base + KDDC	0.0321	0.0140	0.0331	0.0149	0.0338	0.0151	0.0341	0.0156
	Base + CNN-ODE	0.0313	0.0154	0.0327	0.0156	0.0333	0.0161	0.0326	0.0168
	SPOT-Trip	**0.0327**	**0.0162**	**0.0343**	**0.0178**	**0.0384**	**0.0216**	**0.0399**	**0.0190**

the increase of $\sigma_{\tilde{v}^o}$ and then decline gradually, where two exceptions are observed when $\sigma_{\tilde{v}^o}$ is set to 0.3 and 0.4. This indicates a moderate $\sigma_{\tilde{v}^o}$ (e.g., 0.6 on Foursquare) would be a good option for SPOT-Trip as it achieves a balance between latent dynamics and true behavior.

Besides, we assess the effect of each loss weight (β_1, β_2, β_3) by varying one within the range $[0.1, 0.9]$ and setting the other two to $\frac{1-\beta}{2}$, e.g., varying β_1 with $\beta_2 = \beta_3 = \frac{1-\beta_1}{2}$. It can be observed that for the three parameters, the recommendation performance attains its peak when they are set to 0.3 or 0.4. This suggests that moderate loss weights help balance the contributions of different loss components during tuning. Based on this observation, we set $\beta_1 = \beta_2 = \beta_3 = 1$ in the final framework, assuming that each loss term is properly normalized to ensure balanced influence. More results regarding parameter sensitivity can be found in Appendix C.2.

4.5 The Effect of Data Sparsity

Tab. 2 summarizes the performance of various methods trained with limited hometown data. We randomly sample 40%, 60%, and 80% of the hometown trajectories, while preserving the original sequence order and avoiding adjacent duplicate POIs to ensure data quality. It is observed that SPOT-Trip consistently outperforms the strongest baselines, highlighting its robustness in data-scarce scenarios. An interesting observation is that using only 80% of the hometown data leads to better F_1 on Foursquare and $PairsF_1$ on Yelp than using the complete data. We speculate that this improvement may result from the random sampling process, which potentially removes noisy information from the hometown check-ins.



(a) AR-Trip (b) SPOT-Trip (c) SPOT-Trip (O) (d) SPOT-Trip (D)

Figure 5: Visualizations of recommendation results for the user 2964 on Foursquare. (O) denotes a query with a single origin, while (D) denotes a query with a single destination.

4.6 Case Study

To intuitively show the effectiveness of SPOT-Trip, we provide a case study on Foursquare as shown in Fig. 5. We choose AR-Trip for comparison due to its superior performance. The results show that the recommended intermediate POIs of SPOT-Trip exhibit a remarkable level of alignment with the ground truth. It is clear that SPOT-Trip can accurately trace the right out-of-town trip recommendation. Figs. 5(c) and 5(d) show that even with only the origin (SPOT-Trip (O)) or destination (SPOT-Trip (D)) provided, SPOT-Trip generates plausible recommendations (e.g., both the predicted destination and origin correspond to train stations), demonstrating its ability to capture realistic user travel patterns. We have provided more case studies in Appendix C.3.

5 Relate Work

Out-of-town Recommendation. Out-of-town recommendation aims to suggest the next likely new POI for users visiting unfamiliar regions. This task is particularly challenging due to issues such as cold start and interest drift. Early work, such as [13] addressed the problem by exploring social influence. Subsequent studies [53, 43] primarily adopted latent Dirichlet allocation (LDA) to capture interest drift, incorporating user preferences and POI content. In addition, deep learning-based approaches have been proposed to tackle the problem more effectively. For instance, [46] leveraged neural topic modeling to conduct a fine-grained analysis of users' travel intentions, while [48] exploited cross-city mobility matching to mitigate data sparsity. [47] and [32] settled the more intractable pre-travel recommendation with the crowd behavior memory and causal relationship. Despite these efforts, limited attention has been paid to the out-of-town trip recommendation scenario.

Trip Recommendation. Earlier research on trip recommendation predominantly relied on planning-based approaches rooted in the orienteering problem [17]. For example, Popularity [6] focuses solely on POI popularity and Markov [6] models transitions between POIs. Later methods began to move beyond pure optimization formulations. For instance, C-ILP [18] introduces context-aware POI embeddings via linear programming. However, these planning-based methods struggle to capture the complexity and uncertainty of human mobility, motivating the rise of recent learning-based approaches. Transformer-based models like Bert-Trip [23] treat trip recommendation as a sentence completion task to generate personalized itineraries that align with tourists' preferences and real-world constraints, while AR-Trip [39] leverages a prior position-based matrix to alleviate the issue of repetitive recommendations. While effective to some extent, these methods use only position embeddings to model preference changes and are limited in capturing irregular temporal dynamics, which our proposed SPOT-Trip is designed to overcome.

Knowledge Graph-enhanced Recommendation. Existing Knowledge Graph (KG)-enhanced approaches for general recommendation tasks can be broadly classified into three categories: embedding-based, path-based, and graph neural network (GNN)-based methods. Embedding-based methods, such as [35], utilize transition-based models (e.g., TransR [27]) to generate item representations via entity embeddings. Path-based methods [16, 45] enhance user-item connectivity by constructing meta-paths, yet they heavily rely on domain-specific prior knowledge and manual path design. More recent efforts have shifted towards GNN-based techniques [3, 44], which leverage message passing across multi-hop neighbors to capture complex relational structures. KGCL [51] and KGRec [50] further introduce joint self-supervised learning schemes to mitigate data noise but will increase computational overhead. Therefore, our framework alternately leverages embedding-based and GNN-based methods to enhance the semantic representation of users' static preferences.

Ordinary Differential Equation. Neural ODEs [9] represent a novel framework that extends discrete deep neural networks to continuous-time domains by modeling transformations as ordinary differential equations, effectively generalizing architectures such as ResNet [19]. Owing to their strong performance and modeling flexibility, neural ODEs have found wide application across diverse areas, including traffic flow prediction [34], time series forecasting [41], and continuous dynamical systems [15, 21]. Recently, researchers have begun integrating neural ODEs with GNNs by parameterizing the derivative of hidden node states for sequential recommendation [11, 37]. In contrast to existing ODE approaches, we introduce neural ODEs to effectively capture the temporal dynamics of user preferences in out-of-town regions.

6 Conclusion

We present SPOT-Trip in this study, a static-dynamic preference aware out-of-town trip recommendation framework, which features three modules. First, the knowledge-enhanced static preference learning module constructs a POI attribute knowledge graph and employs relation-aware attention to generate enriched POI embeddings, capturing the static preferences. Second, the ODE-based dynamic preference learning module is proposed to leverage neural ODEs with a temporal point process learning the continuous drift of dynamic user preferences. Together with the static-dynamic preference fusion module, SPOT-Trip delivers superior personalized trips for users traveling from their hometown to unfamiliar regions. Comprehensive experiments on two real datasets offer evidence that SPOT-Trip achieves state-of-the-art accuracy. In the future, an interesting research direction is to apply SPOT-Trip to other spatio-temporal tasks, such as trajectory prediction.

Acknowledgments and Disclosure of Funding

This work was supported in part by the National Natural Science Foundation of China under Grant 62476247, 62073295 and 62072409, "Pioneer" and "Leading Goose" R&D Program of Zhejiang under Grant 2025C01030, in part by the Zhejiang Provincial Natural Science Foundation under Grant LR21F020003, in part by the Supcon Research Fund under Grant KYY-HX-20230833, and Lantai Research Fund under Grant KYY-HX-20240573, KYY-HX-20230365, KYY-HX-20250588.

References

[1] A. Agrawal and J. Domke. Amortized variational inference for simple hierarchical models. *NeurIPS*, 34:21388–21399, 2021.

[2] I. Al-Hazwani, T. Luo, O. Inel, F. Ricci, M. El-Assady, and J. Bernard. Scrollypoi: A narrative-driven interactive recommender system for points-of-interest exploration and explainability. In *UMAP*, pages 292–304, 2024.

[3] G. Balloccu, L. Boratto, G. Fenu, and M. Marras. Post processing recommender systems with knowledge graphs for recency, popularity, and diversity of explanations. In *SIGIR*, pages 646–656, 2022.

[4] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. *NeurIPS*, 26, 2013.

[5] J. Cao, J. Fang, Z. Meng, and S. Liang. Knowledge graph embedding: A survey from the perspective of representation spaces. *CSUR*, 56(6):1–42, 2024.

[6] D. Chen, C. S. Ong, and L. Xie. Learning points and routes to recommend trajectories. In *CIKM*, pages 2227–2232, 2016.

[7] P. Y. Chen, J. Xiang, D. H. Cho, Y. Chang, G. A. Pershing, H. T. Maia, M. M. Chiaramonte, K. T. Carlberg, and E. Grinspun. Crom: Continuous reduced-order modeling of pdes using implicit neural representations. In *ICLR*, 2023.

[8] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud. Neural ordinary differential equations. *NeurIPS*, 31, 2018.

[9] R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud. Neural ordinary differential equations. *NeurIPS*, 2018.

[10] Y. Chen, X. Li, G. Cong, C. Long, Z. Bao, S. Liu, W. Gu, and F. Zhang. Points-of-interest relationship inference with spatial-enriched graph neural networks. *VLDB*, 15(3):504–512, 2021.

[11] J. Choi, J. Jeon, and N. Park. Lt-ocf: Learnable-time ode-based collaborative filtering. In *CIKM*, pages 251–260, 2021.

[12] J. Ding, G. Yu, Y. Li, D. Jin, and H. Gao. Learning from hometown and current city: Cross-city poi recommendation via interest drift and transfer learning. *IMWUT*, 3(4):1–28, 2019.

[13] G. Ference, M. Ye, and W.-C. Lee. Location recommendation for out-of-town users in location-based social networks. In *CIKM*, pages 721–726, 2013.

[14] Q. Gao, W. Wang, L. Huang, X. Yang, T. Li, and H. Fujita. Dual-grained human mobility learning for location-aware trip recommendation with spatial–temporal graph knowledge fusion. *Information Fusion*, 92:46–63, 2023.

[15] P. Ghanem, A. Demirkaya, T. Imbiriba, A. Ramezani, Z. Danziger, and D. Erdogmus. Learning physics informed neural odes with partial measurements. In *AAAI*, volume 39, pages 16799–16807, 2025.

[16] A. Gogleva, D. Polychronopoulos, M. Pfeifer, V. Poroshin, M. Ughetto, M. J. Martin, H. Thorpe, A. Bornot, P. D. Smith, B. Sidders, et al. Knowledge graph-based recommendation framework identifies drivers of resistance in egfr mutant non-small cell lung cancer. *Nature Communications*, 13(1):1667, 2022.

[17] A. Gunawan, H. C. Lau, and P. Vansteenwegen. Orienteering problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research*, 255(2):315–332, 2016.

[18] J. He, J. Qi, and K. Ramamohanarao. A joint context-aware embedding for trip recommendations. In *ICDE*, pages 292–303, 2019.

[19] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *ICCV*, pages 770–778, 2016.

[20] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. In *ICLR*, 2020.

[21] V. Iakovlev and H. Lähdesmäki. Learning spatiotemporal dynamical systems from point process observations. In *ICLR*, 2025.

[22] X. Kong, Z. Chen, J. Li, J. Bi, and G. Shen. Kgnext: Knowledge-graph-enhanced transformer for next poi recommendation with uncertain check-ins. *TCSS*, 2024.

[23] A.-T. Kuo, H. Chen, and W.-S. Ku. Bert-trip: effective and scalable trip representation using attentive contrast learning. In *ICDE*, pages 612–623. IEEE Computer Society, 2023.

[24] C.-P. Lee and C.-J. Lin. Large-scale linear ranksvm. *Neural Computation*, 26(4):781–817, 2014.

[25] D. Li and Z. Gong. A deep neural network for crossing-city poi recommendations. *TKDE*, 34(8):3536–3548, 2020.

[26] K. H. Lim, J. Chan, C. Leckie, and S. Karunasekera. Personalized tour recommendation based on user interests and points of interest visit durations. In *IJCAI*, volume 15, pages 1778–1784, 2015.

[27] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 29, 2015.

[28] C. Liu, H. Miao, Q. Xu, S. Zhou, C. Long, Y. Zhao, Z. Li, and R. Zhao. Efficient multivariate time series forecasting via calibrated language models with privileged knowledge distillation. In *ICDE*, pages 3165–3178, 2025.

[29] C. Liu, Q. Xu, H. Miao, S. Yang, L. Zhang, C. Long, Z. Li, and R. Zhao. Timecma: Towards llm-empowered multivariate time series forecasting via cross-modality alignment. In *AAAI*, volume 39, pages 18780–18788, 2025.

[30] C. Liu, S. Zhou, Q. Xu, H. Miao, C. Long, Z. Li, and R. Zhao. Towards cross-modality modeling for time series analytics: A survey in the llm era. In *IJCAI*, 2025.

[31] Q. Liu, X. Wu, Y. Wang, Z. Zhang, F. Tian, Y. Zheng, and X. Zhao. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. *NeurIPS*, 37:26701–26727, 2024.

[32] Y. Liu, G. Shen, C. Cui, Z. Zhao, X. Han, J. Du, X. Zhao, and X. Kong. Kddc: Knowledge-driven disentangled causal metric learning for pre-travel out-of-town recommendation. In *IJCAI*, pages 4–9, 2024.

[33] D. Lüdke, M. Biloš, O. Shchur, M. Lienen, and S. Günnemann. Add and thin: Diffusion for temporal point processes. *NeurIPS*, 36:56784–56801, 2023.

[34] G. Mercatali, A. Freitas, and J. Chen. Graph neural flows for unveiling systemic interactions among irregularly sampled time series. In *NeurIPS*, 2024.

[35] H. Mezni, D. Benslimane, and L. Bellatreche. Context-aware service recommendation based on knowledge graph embedding. *TKDE*, 34(11):5225–5238, 2021.

[36] H. Miao, Z. Liu, Y. Zhao, C. Guo, B. Yang, K. Zheng, and C. S. Jensen. Less is more: Efficient time series dataset condensation via two-fold modal matching. *PVLDB*, 18(2):226–238, 2024.

[37] Y. Qin, W. Ju, H. Wu, X. Luo, and M. Zhang. Learning graph ode for continuous-time sequential recommendation. *TKDE*, 36(7):3224–3236, 2024.

[38] A. Sharma, R. Johnson, F. Engert, and S. Linderman. Point process latent variable models of larval zebrafish behavior. *NeurIPS*, 31, 2018.

[39] W. Shu, K. Xu, W. Tai, T. Zhong, Y. Wang, and F. Zhou. Analyzing and mitigating repetitions in trip recommendation. In *SIGIR*, pages 2276–2280, 2024.

[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017.

[41] Y. Verma, M. Heinonen, and V. Garg. Climode: Climate and weather forecasting with physics-informed neural odes. In *ICLR*, 2024.

[42] F. Wang, C. Chen, W. Liu, M. Lei, J. Chen, Y. Liu, X. Zheng, and J. Yin. Dr-vae: Debiased and representation-enhanced variational autoencoder for collaborative recommendation. In *AAAI*, volume 39, pages 12703–12711, 2025.

[43] H. Wang, Y. Fu, Q. Wang, H. Yin, C. Du, and H. Xiong. A location-sentiment-aware recommender system for both home-town and out-of-town users. In *SIGKDD*, pages 1135–1143, 2017.

[44] S. Wang, Y. Sui, C. Wang, and H. Xiong. Unleashing the power of knowledge graph for recommendation via invariant learning. In *WWW*, pages 3745–3755, 2024.

[45] Y. Wei, W. Liu, F. Liu, X. Wang, L. Nie, and T.-S. Chua. Lightgt: A light graph transformer for multimedia recommendation. In *SIGIR*, pages 1508–1517, 2023.

[46] H. Xin, X. Lu, T. Xu, H. Liu, J. Gu, D. Dou, and H. Xiong. Out-of-town recommendation with travel intention modeling. In *AAAI*, volume 35, pages 4529–4536, 2021.

[47] H. Xin, X. Lu, N. Zhu, T. Xu, D. Dou, and H. Xiong. Captor: A crowd-aware pre-travel recommender system for out-of-town users. In *SIGIR*, pages 1174–1184, 2022.

[48] S. Xu and D. Guan. Crosspred: A cross-city mobility prediction framework for long-distance travelers via poi feature matching. In *CIKM*, pages 4148–4152, 2024.

[49] W. Xu, S. Zheng, L. He, B. Shao, J. Yin, and T.-Y. Liu. Seek: Segmented embedding of knowledge graphs. In *ACL*, pages 3888–3897, 2020.

[50] Y. Yang, C. Huang, L. Xia, and C. Huang. Knowledge graph self-supervised rationalization for recommendation. In *SIGKDD*, pages 3046–3056, 2023.

[51] Y. Yang, C. Huang, L. Xia, and C. Li. Knowledge graph contrastive learning for recommendation. In *SIGIR*, pages 1434–1443, 2022.

[52] F. Yin, Y. Liu, Z. Shen, L. Chen, S. Shang, and P. Han. Next poi recommendation with dynamic graph and explicit dependency. In *AAAI*, volume 37, pages 4827–4834, 2023.

[53] H. Yin, B. Cui, X. Zhou, W. Wang, Z. Huang, and S. Sadiq. Joint modeling of user check-in behaviors for real-time point-of-interest recommendation. *TOIS*, 35(2):1–44, 2016.

[54] J. Zhang, M. Ma, X. Gao, and G. Chen. Encoder-decoder based route generation model for flexible travel recommendation. *TSC*, 2024.

[55] Y. Zhang, X. Kong, Z. Shen, J. Li, Q. Yi, G. Shen, and B. Dong. A survey on temporal knowledge graph embedding: Models and applications. *KBS*, page 112454, 2024.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The scope and contributions of the paper are included in the abstract and Sec. 1. Please refer to the first and second paragraphs of Sec. 1 for the scope, and the last paragraph for the contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: A limitation section is included in the Appendix D.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

   Justification: There is no theoretical result in this paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The implementation details, including hardware and software specifications, are provided in Sec. 4.1 and Appendix B.4. Additionally, we release the code to facilitate reproducibility.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The code can be found at `https://anonymous.4open.science/r/SPOT-Trip-0607`. We conduct experiments on two public real-world datasets: Foursquare and Yelp.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: We provide the details of the experimental settings, such as the data split, optimizer, etc., in the experimental setting section (Sec. 4.1) and the Appendix B.4.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We present the results of two-sided t-tests with $p < 0.05$ results in the main experiments, i.e., Table 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the details of compute resources in the implementation detail section (Sec. 4.1 and Appendix B.4).

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have made sure that our paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive impacts that our work will bring in Sec.1.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: The proposed method poses no risk of misuse, and the datasets employed in this study are publicly available.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have referenced the original publications or provided links to the existing resources utilized in this paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: We provide detailed instructions for running the code as well as the license in the code repository.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: LLMs is not an important, original, or non-standard component of the core methods in this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Technical Appendix: Methodology Details

In this section, we provide more technical details of SPOT-Trip.

A.1 Alternative Training

As discussed in Sec. 3.1, the relation-aware knowledge aggregator will be alternately trained with TransE [4] to improve the multi-relational semantic representation space. The core principle of this translation-based knowledge graph embedding is to ensure that the sum of the head and relation embeddings (i.e., $\mathbf{v}$ and $\mathbf{r}$) approximates the tail embedding $\mathbf{e}$. To quantify similarity, we define $f_d(\cdot)$ as an L1 norm-based measurement function, i.e., $f_d = \|\mathbf{v} + \mathbf{r} - \mathbf{e}\|$. Formally, the translation-based loss $\mathcal{L}_{TransE}$ is expressed as

$$\mathcal{L}_{TransE} = \sum_{(v,r,e,e') \in \mathcal{G}_K} - \ln \sigma\Big(f_d(\mathbf{v}, \mathbf{r}, \mathbf{e}') - f_d(\mathbf{v}, \mathbf{r}, \mathbf{e})\Big), \tag{13}$$

where the negative sample e' is generated by randomly substituting the tail e in the observed triplet (v, r, e) from the knowledge graph $\mathcal{G}_K$.

A.2 Approximate Posterior

We define the approximate posterior with $\psi = [\psi_\mu, \psi_{\sigma^2}]$ as

$$q(\tilde{\mathbf{p}}_1^o; \psi) \;=\; \mathcal{N}\big(\tilde{\mathbf{p}}_1^o \,\big|\, \psi_\mu,\, \mathrm{diag}\big(\psi_{\sigma^2}\big)\big), \tag{14}$$

where $\mathcal{N}$ is the normal distribution, and $\mathrm{diag}\big(\psi_{\sigma^2}\big)$ is a diagonal matrix with vector ψ_{σ^2} on its diagonal. As discussed in Sec. 3.2, ψ is obtained from $\mathbf{A}\tilde{\mathbf{G}}\mathbf{G}$ with the reparameterization sampling, where we further break up ψ into $[\psi_\mu, \psi_{\sigma^2}]$ as follows:

$$\psi_\mu = \mathrm{Linear}\big(\mathbf{A}\tilde{\mathbf{G}}\mathbf{G}\big), \psi_{\sigma^2} = \exp\Big(\mathrm{Linear}\big(\mathbf{A}\tilde{\mathbf{G}}\mathbf{G}\big)\Big), \tag{15}$$

where $\mathrm{Linear}(\cdot)$ are separate linear mappings.

A.3 ELBO

Based on the definitions presented in the previous sections, the ELBO can be formulated as:

$$\begin{aligned}
\mathrm{ELBO} &= \int q(\tilde{\mathbf{p}}_1^o)\, \ln \frac{p\big(\tilde{\mathbf{p}}_1^o, \{t_n, \tilde{\mathbf{v}}_n^o\}_{n=1}^N\big)}{q(\tilde{\mathbf{p}}_1^o)}\, d\tilde{\mathbf{p}}_1^o \\
&= \int q(\tilde{\mathbf{p}}_1^o)\, \ln \frac{\prod_{n=1}^N p\big(\tilde{\mathbf{v}}_n^o | \tilde{\mathbf{p}}_1^o, t_n\big) p\big(\{t_n\}_{n-1}^N | \tilde{\mathbf{p}}_1^o\big) p(\tilde{\mathbf{p}}_1^o)}{q(\tilde{\mathbf{p}}_1^o)}\, d\tilde{\mathbf{p}}_1^o \\
&= \int q(\tilde{\mathbf{p}}_1^o)\, \ln p\big(\tilde{\mathbf{p}}_1^o, \{t_n, \tilde{\mathbf{v}}_n^o\}_{n=1}^N\big) d\tilde{\mathbf{p}}_1^o \\
&\quad + \int q(\tilde{\mathbf{p}}_1^o)\, \ln p\big(\{t_n\}_{n-1}^N | \tilde{\mathbf{p}}_1^o\big) d\tilde{\mathbf{p}}_1^o \\
&\quad + \int q(\tilde{\mathbf{p}}_1^o)\, \ln \frac{p(\tilde{\mathbf{p}}_1^o)}{q(\tilde{\mathbf{p}}_1^o)} d\tilde{\mathbf{p}}_1^o \\
&= \sum_{n=1}^N \mathbb{E}_{q(\tilde{\mathbf{p}}_1^o)}[\ln p(\tilde{\mathbf{v}}_n^o | t_n^o, \tilde{\mathbf{p}}_1^o)] \\
&\quad + \mathbb{E}_{q(\tilde{\mathbf{p}}_1^o)} \sum_{n=1}^N \ln \lambda(\tilde{\mathbf{p}}_n^o) - \int \lambda(\tilde{\mathbf{p}}^o) dt \\
&\quad - \mathrm{KL}[q(\tilde{\mathbf{p}}_1^o) || p(\tilde{\mathbf{p}}_1^o)]
\end{aligned} \tag{16}$$

Algorithm 1 Optimization phase of SPOT-Trip

Input: POI knowledge graph $\mathcal{G}_k$, user out-of-town trip records $\mathcal{O}$, POI set $\mathcal{V}$, training epochs E
Output: Out-of-town trip recommender $\mathcal{F}_\theta$

1: Initialize POI/entity/relation embedding layers and other framework parameters;
2: **for** epoch $= 1$ to E **do**
3: Update POI/entity/relation embedding layers with TransE (Appendix A.1) from $\mathcal{G}_k$;
4: Get knowledge-aware POI embeddings via Eq. 1 and latent POI embeddings by $\mathbf{E}(\cdot)$;
5: **for** each user u with trip $\xi = (u, \vec{c}_h, \vec{c}_o, a_h, a_o)$ **do**
6: Get the static preference representations, i.e., $\bar{\mathbf{u}}^h$ and $\bar{\mathbf{u}}^o$;
7: Obtain inferred out-of-town static preference $\bar{\mathbf{P}}^o$ via Eq. 2;
8: Calculate the static learning loss $\mathcal{L}_S$ via Eq. 3;
9: Obtain inferred out-of-town dynamic preference $\tilde{\mathbf{P}}^o = \{\tilde{\mathbf{p}}_n^o\}_1^N$ via Eq. 4;
10: Calculate the dynamic learning loss $\mathcal{L}_D$ via Eq. 9;
11: Fusion preferences and compute the logits $z_{u,n,i}$ for all POIs located in a_o via Eq. 10;
12: Calculate the main recommendation loss $\mathcal{L}_R$ via Eq. 11;
13: Sum $\mathcal{L}_S$, $\mathcal{L}_D$ and $\mathcal{L}_R$ to obtain $\mathcal{L}$ (Eq. 12). Then, update $\mathcal{F}_\theta$ by minimizing $\mathcal{L}$;
14: **end for**
15: **end for**
16: **return** $\mathcal{F}_\theta$.

Algorithm 2 Recommendation phase of SPOT-Trip

Input: User u^*, hometown check-ins $\vec{c}_h$, query $Q_o^* = \{v_s^o, v_e^o, N\}$, region a_o^*
Output: Out-of-town trip τ

1: Obtain the inferred static preference $\bar{\mathbf{P}}^o$ and dynamic preference $\tilde{\mathbf{P}}^o$ via Eqs. 2 and 4;
2: Fuse preferences and compute the logits $z_{u,n,i}$ for all POIs located in a_o^* via Eq. 10;
3: Initialize recommended trip $\tau = [v_s^o]$;
4: **for** $n = 2$ to $N - 1$ **do**
5: Select $v_n^o \sim$ Top-P$(z_{u,n,i})$;
6: Append v_n^o to τ;
7: **end for**
8: Append $v_N^o = v_e^o$ to τ;
9: **return** $\tau = \{v_1^o, \ldots, v_N^o\}$.

A.4 Query Representation

A user's query typically reflects her/his inherent semantic preference. Therefore, given the query $Q^o = \{v_0^*, v_N^*, N\}$, we use semantic knowledge aggregation explained in Sec. 3.1 to obtain the knowledge-enhanced query embeddings $\bar{\mathbf{Q}}^o$. Then we combine $\bar{\mathbf{Q}}^o$ with positional embeddings $\mathbf{N} = \{\mathbf{n}\}_{n=1}^N$ to initialize a Transformer encoder Trans_Q, thereby gaining the position-aware query representation $\mathbf{Q}^o = \mathrm{Trans}_Q(\bar{\mathbf{Q}}^o || \mathbf{N})$.

A.5 Optimization and Recommendation Phases

The detailed algorithm for the optimization and recommendation phases of SPOT-Trip are summarized in Algorithm 1 and 2, to facilitate understanding and implementation. At the beginning of the optimization (Algorithm 1), the POI, entity, and relation embedding layers and other parameters in the framework are initialized (line 1). Next, knowledge-aware POI embeddings are obtained via alternating training, while latent POI embeddings are learned through a separate embedding layer (lines 3-4). Next, calculate the static learning loss (lines 6-8), the dynamic learning loss (lines 9-10), and the main recommendation loss (lines 11-12). Through the sum of these three losses (line 13), we can optimize the whole SPOT-Trip. For the recommendation phase (Algorithm 2), we first obtain the inferred static preference and dynamic preference for a new user (line 1). Based on these preferences, logits for all POIs within the target region are computed for Top-P sampling (line 2). Finally, the recommended trip is generated according to the user's query (lines 3-9).

Table 3: Statistics of the two datasets.

Dataset	Raw Records						Knowledge Graph		
	#Users	#Regions	#POIs	#Check-ins	#Hometown Check-ins	#Out-of-town Check-ins	#Relations	#Entities	#Triples
Foursquare	3,007	21	23,884	126,219	109,225	16,994	2	411	47,768
Yelp	4,417	214	29,930	78,882	58,403	20,479	8	53,549	353,918

B Additional Experimental Details

We proceed to provide more details about the experimental settings.

B.1 Dataset

The experiments in this study are conducted on two widely used check-in datasets: Foursquare[2] and Yelp[3]. Table 3 presents the statistical summary of these datasets, including details on check-in records and knowledge graph characteristics. For check-in data processing, we first identified users who had check-ins in both their hometown and out-of-town regions. Each user's check-in sequence was then reformulated into an out-of-town travel record $\xi = (u, \vec{c}_h, \vec{c}_o, a_h, a_o)$ (see **Definition 3**). To ensure dataset quality, we filtered out Points of Interest (POIs) visited fewer than two times and removed users whose travel records did not meet the following criteria: (1) $\vec{c}_h \geq 4$; (2) $\vec{c}_o \geq 3$; and (3) the frequency of $(a_h, a_o) \geq 10$. Furthermore, we normalized the timestamps and geographic coordinates (latitude and longitude) of all check-ins to the $[0, 1]$ range to facilitate model training. Subsequently, the datasets were randomly partitioned by user into training, validation, and testing sets with a ratio of 80%, 10%, and 10%, respectively. To ensure fairness in evaluation, all users were anonymized. For the construction of knowledge graphs, we generated entity-specific relations using various types of auxiliary information, such as categories, star ratings, user reviews, and associated regions.

B.2 Baseline

There are two groups of state-of-the-art baselines that are compared within this paper, i.e., 7 trip recommendation-based baselines and 3 out-of-town trip recommendation-based baselines.

Trip Recommendation-based Baselines. These methods primarily focus on modeling POIs within the target out-of-town regions, while paying limited attention to the users' historical check-in behaviors in their hometowns.

- PersTour [26]. It treats trip recommendation as an orienteering problem and generates a trip by leveraging the features of POIs while respecting a given time budget.
- Popularity [6]. It recommends the most popular or frequently visited POIs to a user at each query candidate position.
- POIRank [6]. It generates a travel trajectory by first ranking POIs using the RankSVM [24] method, and then sequentially connecting them based on their ranking scores.
- GraphTrip [14]. This method proposes a two-stage graph learning framework to model multiple heterogeneous POI graphs, and designs a dual-grained mobility module to capture both coarse-grained category information and fine-grained transitional patterns among POIs.
- MatTrip [54]. This work employs dual long-short-term-memory (LSTM) based encoders to learn users' category preferences and the geographical proximity of POIs, followed by an attention-based LSTM decoder that generates user-preferred trips with the aid of an optimized search strategy.
- AR-Trip [39]. This approach is based on a Transformer encoder-only architecture and incorporates additional prior position information to recommend trips with low repetition rates.
- Base. It is a simplified version of our framework that removes all hometown-related information learning and serves as a baseline to examine the contribution of hometown-aware modeling.

[2]https://sites.google.com/site/yangdingqi/home/foursquare-dataset

[3]https://www.yelp.com.tw/dataset

Out-of-town Trip Recommendation-based Baselines. These baselines distinctively incorporate supplementary historical hometown information for out-of-town recommendations.

- Base + KDDC [32]. KDDC introduces a knowledge graph approach that strengthens semantic interaction and leverages disentangled causal metric learning to align recommended POIs more closely with the target preferences. In the Base + KDDC variant, the knowledge graph approach is integrated to strengthen the Base's static semantic preference alignment ability.
- Base + CNN-ODE [21]. We isolate the module *ODPL* from SPOT-Trip and replace the Transformer encoder before the Neural ODE with a CNN encoder to support dynamic learning in the Base framework.
- Base + PPROC [21]. PPROC combines neural spatiotemporal point processes [38] and neural partial differential equations [7] to improve the prediction of observations at probabilistic locations and timings. It serves as the dynamic preference learning module in the Base + PPROC variant.

B.3 Evaluation Metrics

As described in Sec. 4.1, our evaluation employs four metrics: F_1, $PairsF_1$, $Full\text{-}F_1$, and $Full\text{-}PairsF_1$. Formally, let $\tau = \{v_1^o, v_2^o, ..., v_N^o\}$ be the generated trips by a method and τ^* be the ground truth, the calculation procedures of these metrics are detailed as follows.

$$\begin{aligned} Precision = \frac{|\tau_{[2:N-1]} \cap \tau^*_{[2:N-1]}|}{|\tau_{[2:N-1]}|}, Recall = &\frac{|\tau^*_{[2:N-1]} \cap \tau_{[2:N-1]}|}{|\tau^*_{[2:N-1]}|}, \\ F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}&. \end{aligned} \tag{17}$$

$$\begin{aligned} Precision_{pairs} = \frac{N_c}{\binom{|\tau_{[2:N-1]}|}{2}} \quad , \quad Recall_{pairs} = &\frac{N_c}{\binom{|\tau^*_{[2:N-1]}|}{2}}, \\ PairsF_1 = \begin{cases} \frac{2 \times Precision_{pairs} \times Recall_{pairs}}{Precision_{pairs} + Recall_{pairs}}, & N_c > 0, \\ 0, & N_c = 0, \end{cases} \end{aligned} \tag{18}$$

where N_c is the number of correctly ordered POI-pairs in the recommendation. $\binom{|\tau_{[2:N-1]}|}{2}$ and $\binom{|\tau^*_{[2:N-1]}|}{2}$ are the total number of ordered pairs, respectively. Correspondingly,

$$\begin{aligned} Full\text{-}Precision = \frac{|\tau \cap \tau^*|}{|\tau|}, Full\text{-}Recall = &\frac{|\tau^* \cap \tau|}{|\tau^*|}, \\ Full\text{-}F_1 = \frac{2 \times Full\text{-}Precision \times Full\text{-}Recall}{Full\text{-}Precision + Full\text{-}Recall}&. \end{aligned} \tag{19}$$

$$\begin{aligned} Full\text{-}Precision_{pairs} = \frac{N_c}{\binom{|\tau|}{2}} \quad , \quad Full\text{-}Recall_{pairs} = &\frac{N_c}{\binom{|\tau^*|}{2}}, \\ Full\text{-}PairsF_1 = \begin{cases} \frac{2 \times Full\text{-}Precision_{pairs} \times Full\text{-}Recall_{pairs}}{Full\text{-}Precision_{pairs} + Full\text{-}Recall_{pairs}}, & N_c > 0. \\ 0, & N_c = 0. \end{cases} \end{aligned} \tag{20}$$

Due to the less precise evaluative nature of $Full\text{-}F_1$ and $Full\text{-}PairsF_1$, we report them only in the overall comparison experiments.

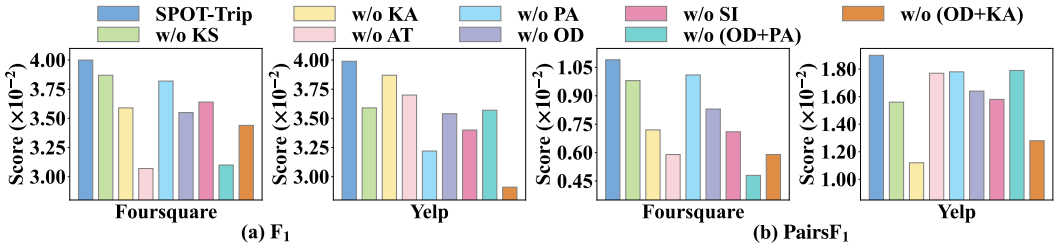


Figure 6: Performance of SPOT-Trip and its variants on two datasets.

B.4 Implementation Details

We re-implement the baselines and their hyper-parameters based on the details provided in their original papers and publicly available source codes[4][5][6][7][8]. For the two baselines that require periodic information, Graph-Trip [14] and AR-Trip [39], we provide additional hour-of-day features as input. In contrast, PPROC [21] relies on both temporal and spatial points to infer preferences at specific spatiotemporal locations. To support this, we additionally design two separate gated recurrent unit (GRU) layers to predict the potential time and location points for future trips. Following [39], we fix the hidden size of all embeddings to 32 in all our experiments. The optimizer is uniformly chosen as Adam with an initial learning rate of 0.001 and L2 regularization with a weight of 10^{-5}. To avoid overfitting, we adopt the early stop strategy with an 8-epoch patience. In addition, the parameters of our framework and all its variants are consistently set as follows: Consistent with [21], the number of Transformer layers in module *ODPL* is set to 4, while both $f(\cdot)$ and $\lambda(\cdot)$ are implemented as 3-layer MLPs. The latent dimensions of these MLPs are searched from $\{16, 32, 64, 128, 256\}$, with 128 selected as the optimal value based on validation performance. Similarly, we use a differentiable $dopri5$ ODE solver with $rtol = atol = 10^{-5}$ from $torchdiffeq$ package [9]. For the static-dynamic preference fusion (Sec. 3.3), the number of Transformer layers is set to 1, following the configuration in [39]. All Transformer layers employ 4 attention heads. The hyper-parameter $\sigma_{\tilde{v}^o}$ is tuned separately for each dataset, with the optimal value set to 0.6 for Foursquare and 0.4 for Yelp. In the optimization stage, the weights of the loss terms β_1, β_2 and β_3 for two datasets are set as 1.

C Additional Experimental Results

In this section, we will show more experimental results to further analyze the effectiveness of our SPOT-Trip.

C.1 Ablation Study

Beyond the three variants introduced in Sec. 4.3 (**w/o KS**, **w/o OD**, and **w/o SI**), Fig. 6 presents a more comprehensive ablation study involving the removal of various modules and submodules. These include: (1) **w/o KA**. SPOT-Trip removes the semantic knowledge aggregation in *KSPL*; (2) **w/o AT**. SPOT-Trip removes the alternative training in KA; (3) **w/o PA**. SPOT-Trip removes static preference alignment in *KSPL*; and two combination removals: (4) **w/o (OD+PA)**. SPOT-Trip makes recommendations solely based on the knowledge-enhanced query embedding; and (5) **w/o (OD+KA)**. SPOT-Trip utilizes the raw hometown representation, without semantic enhancement, to support the recommendation. Overall, the performance degradation observed in each variant demonstrates the effectiveness of every component within the SPOT-Trip architecture. In terms of F_1 score, the most pronounced performance drop was observed for the **w/o AT** variant on Foursquare and the **w/o (OD+KA)** variant on Yelp. This discrepancy may stem from dataset scale: Foursquare's sparse triples require additional training, whereas Yelp's larger dataset risks noise without sufficient semantic augmentation. Regarding the PairsF1 score, the **w/o (OD+PA)** ablation on Foursquare and the **w/o**

[4]https://github.com/gcooq/GraphTrip
[5]https://github.com/Mamingqian/MatTrip
[6]https://github.com/Joysmith99/AR-Trip
[7]https://github.com/Yinghui-Liu/KDDC
[8]https://github.com/yakovlev31/pproc-dyn

Table 4: Several Component Replacement Ablation Experiments.

Method	Foursquare		Yelp	
	F_1	$PairsF_1$	F_1	$PairsF_1$
SPOT-Trip	**0.0400**	**0.0109**	**0.0399**	**0.0190**
The replacements of the two main modules				
Base + KDDC [32] + *ODPL*	0.0385	0.0089	0.0361	0.0174
Base + *KSPL* + CNN-ODE [21]	0.0390	0.0107	0.0351	0.0186
Base + *KSPL* + Transformer	0.0362	0.0063	0.0370	0.0172
Base + *KSPL* + GRU	0.0341	0.0070	0.0358	0.0169
The replacements of the alternative training method				
TransR [27]	0.0378	0.0091	0.0366	0.0182
SEEK (2) [49]	0.0348	0.0082	0.0342	0.0146
SEEK (4) [49]	0.0376	0.0087	0.0346	0.0154
SEEK (8) [49]	0.0394	0.0108	0.0361	0.0183
SEEK (16) [49]	0.0346	0.0073	0.0335	0.0134
The replacement of the parallel design				
SPOT-Trip (*ODPL -> KSPL*)	0.0351	0.0078	0.0344	0.0153

Table 5: Activation Function Replacement Ablation Experiments. Triple $\phi_{(1)}$–$\phi_{(2)}$–$\phi_{(10)}$ denotes the activations in Eqs 1, 2 and 10. L/S/G is *LeakyReLU*/*SiLU*/*GELU*.

$\phi_{(1)}$–$\phi_{(2)}$–$\phi_{(10)}$	Foursquare-F_1	Foursquare-$PairsF_1$	Yelp-F_1	Yelp-$PairsF_1$
L-L-L	0.0386	0.0094	0.0378	0.0175
L-L-S	0.0391	0.0101	0.0384	0.0181
L-L-G	**0.0407**	0.0097	0.0391	0.0178
L-S-L	0.0400	**0.0109**	**0.0399**	0.0190
L-S-S	0.0397	0.0105	0.0394	0.0182
L-S-G	0.0391	0.0103	0.0389	0.0179
L-G-L	0.0381	0.0085	0.0387	0.0193
L-G-S	0.0387	0.0100	0.0391	0.0181
L-G-G	0.0381	0.0097	0.0392	0.0185
S-L-L	0.0378	0.0095	0.0381	0.0187
S-L-S	0.0389	0.0101	0.0394	0.0185
S-L-G	0.0389	0.0104	0.0398	0.0181
S-S-L	0.0402	0.0107	0.0380	0.0184
S-S-S	0.0401	0.0103	0.0389	0.0181
S-S-G	0.0392	0.0108	0.0379	0.0178
S-G-L	0.0384	0.0102	0.0385	0.0184
S-G-S	0.0397	0.0103	0.0381	0.0175
S-G-G	0.0391	0.0103	0.0394	0.0191
G-L-L	0.0395	0.0108	0.0391	0.0179
G-L-S	0.0381	0.0103	0.0394	0.0187
G-L-G	0.0394	0.0101	0.0397	**0.0195**
G-S-L	0.0384	0.0107	0.0378	0.0178
G-S-S	0.0389	0.0102	0.0394	0.0184
G-S-G	0.0394	0.0095	0.0388	0.0190
G-G-L	0.0381	0.0084	0.0398	0.0185
G-G-S	0.0375	0.0098	0.0387	0.0183
G-G-G	0.0391	0.0105	0.0391	0.0188

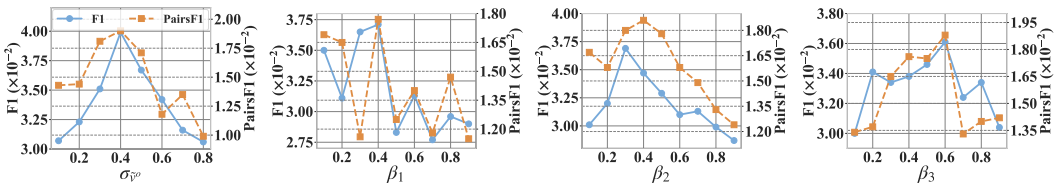
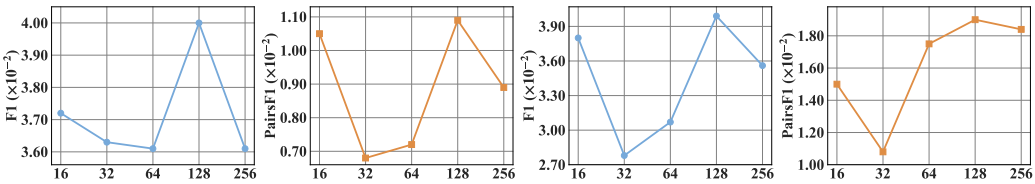


Figure 7: The effects of discrepancy tolerance parameter ($\sigma_{\tilde{v}^o}$) and various loss function weights (β_1, β_2 and β_3) on the Yelp dataset w.r.t. the F_1 and $PairsF_1$ score.



(a) Foursquare (b) Yelp

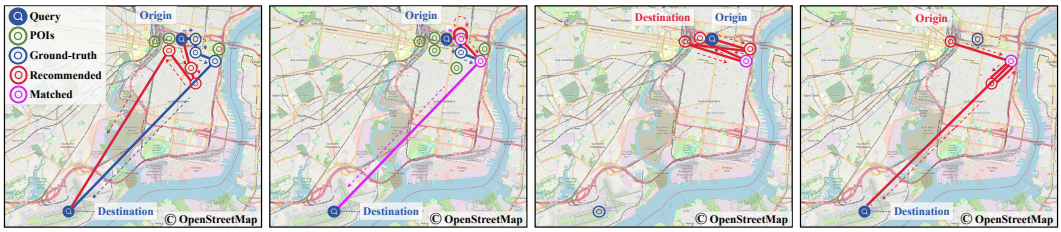
Figure 8: The effects of the number of hidden states of MLPs in both $f(\cdot)$ and $\lambda(\cdot)$ on two datasets.

SM ablation on Yelp exhibit the largest performance drops. The findings demonstrate the crucial importance of learning static and dynamic preferences.

In addition to the removal-based ablation studies, we further conducted a series of component replacement ablation experiments in Tab. 4. Firstly, we replace the two main modules of our framework with top-performing baselines: KDDC (Base + KDDC + *ODPL*) and CNN-ODE (Base + *KSPL* + CNN-ODE), to further assess the relative contribution of each component. Compared to our full framework, the variant Base + KDDC + *ODPL* results in a 22.47% drop in $PairsF_1$ on the Foursquare dataset, while Base + *KSPL* + CNN-ODE leads to a 13.68% decrease in F_1 on Yelp. To further examine the advantage of employing ODE-based temporal modeling, we also replace the ODE module with alternative sequence modelers, including Transformer and GRU, forming Base + *KSPL* + Transformer and Base + *KSPL* + GRU. Both variants exhibit noticeable performance degradation across datasets, suggesting that while Transformer and GRU can capture temporal dependencies to some extent, they struggle to represent the continuous and irregular temporal dynamics effectively modeled by ODEs. These results highlight the unique advantages and effectiveness of our framework design. Secondly, to assess the effectiveness of our alternating training strategy, we replace TransE in *KSPL* with TransR [27] and SEEK [49]. TransR models entities and relations in separate vector spaces to capture complex relational patterns, while SEEK introduces segmented embeddings and interaction-aware scoring functions to enhance expressiveness. For SEEK (k), we set the number of embedding segments k to $\{2, 4, 8, 16\}$. The observed performance drops indicate that TransE is better suited for our framework, likely due to its low model complexity and stable optimization behavior under alternating training. In contrast, more expressive models such as TransR and SEEK may require more careful tuning to fully realize their potential. Next, we replace the original parallel architecture with a sequential variant, where the knowledge-enhanced POI embeddings are directly used for modeling users' dynamic preferences, denoted as SPOT-Trip (*ODPL -> KSPL*). This cascaded design significantly underperforms our parallel architecture, likely due to interference between static and dynamic preference signals when modeled in a non-decoupled manner. This result further validates the advantage of our parallel modeling design. Finally, we additionally performed an exhaustive activation-function replacement ablation covering all $3 \times 3 \times 3 = 27$ combinations of $\phi_{(1)}, \phi_{(2)}, \phi_{(10)} \in \{$*LeakyReLU, SiLU, GELU*$\}$. The full results are reported in Tab. 5. Across the 27 configurations, although L–L–G achieves the highest F_1 on Foursquare (0.0407) and G–L–G attains the highest $PairsF_1$ on Yelp (0.0195), our original activation setup consistently performs best overall across both datasets and other metrics. Moreover, the performance of alternative configurations varies only slightly, reflecting the model's strong stability with respect to the choice of activation functions.

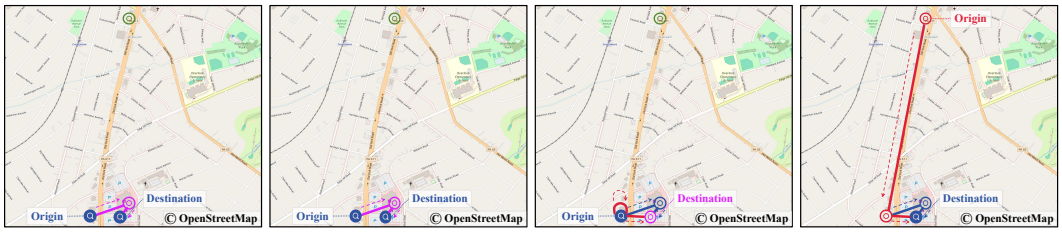
C.2 Hyper-parameter Analysis

Fig. 7 illustrates how varying the tolerance parameter $\sigma_{\tilde{v}^o}$ and the loss function weights β_1, β_2 and β_3 affects the performance of SPOT-Trip on the Yelp dataset. This line chart result of $\sigma_{\tilde{v}^o}$ shows a similar



(a) AR-Trip (b) SPOT-Trip (c) SPOT-Trip (O) (d) SPOT-Trip (D)

Recommendation results for user 1090 on Foursquare



(e) AR-Trip (f) SPOT-Trip (g) SPOT-Trip (O) (h) SPOT-Trip (D)

Recommendation results for user 1544 on Yelp

Figure 9: Visualizations of recommendation results for users on Foursquare and Yelp.

trend to the result in Fig. 4: as $\sigma_{\tilde{v}^o}$ increases, both F_1 and $PairsF_1$ scores rise to a maximum at 0.4 and then gradually fall, with only a minor rebound at 0.7. Therefore, setting $\sigma_{\tilde{v}^o} = 0.4$ balances model expressiveness with fidelity to real user behavior. For β_1, β_2 and β_3, we observe that F_1 and $PairsF_1$ generally peak when $\beta = 0.3$ or 0.4. The only notable exceptions occur at $\beta_1 = 0.3$ and $\beta_3 = 0.6$, where performance slightly deviates. Nevertheless, these results confirm that moderate weighting yields the best balance across loss terms. Accordingly, we retain $\beta_1 = \beta_2 = \beta_3 = 1$ in our final framework, assuming proper normalization of each component.

Moreover, we simultaneously vary the latent dimensions of the ODE functions $f(\cdot)$ and $\lambda(\cdot)$ from 16 to 256 to study their joint impact on user dynamic preference modeling in SPOT-Trip. As shown in Fig. 8, we vary the latent dimensions from 16 to 256 and evaluate the performance on both Foursquare and Yelp datasets using F_1 and $PairsF_1$ as metrics. On Foursquare, both F_1 and $PairsF_1$ initially decrease at smaller dimensions (32 and 64), but sharply increase at 128, which yields the best performance. A slight degradation is observed at 256, likely due to overfitting or increased optimization difficulty. A similar trend is observed on Yelp, where performance significantly improves with larger dimensions and peaks at 128 before plateauing or dropping slightly at 256. These findings suggest that the choice of latent dimension also plays a critical role in Neural ODE-based preference modeling. Extremely small dimensions may limit expressiveness, while overly large dimensions can introduce instability. Thus, a latent dimension of 128 is adopted in our framework.

C.3 Case Study

We present more case studies for user 1090 on the Foursquare dataset and user 1544 on the Yelp dataset in Fig. 9. To avoid visual overlap and improve readability, some marks and lines have been reduced in size or omitted. The results display that as the number of intermediate query points increases (i.e., the user 1090's), AR-Trip's recommendations become completely off-target, whereas SPOT-Trip tracks the ground truth closely, demonstrating our framework's robustness to varying user query lengths. Notably, SPOT-Trip's performance drops when handling queries that specify only an origin or only a destination. In future work, we will strive to enhance the method's capability to meet such specific user requirements.

D Limitation

Despite the superior overall performance of SPOT-Trip, Fig. 9 (b) and (g) exhibit repeated recommendations at proximate positions, which may undermine trip diversity. While AR-Trip [39] incorporates a prior position matrix to partially mitigate this repetition, it is prone to error propagation: an incorrect POI recommendation at a position can mislead predictions at other positions, resulting in compounded inaccuracies. This cascading effect may ultimately degrade the quality of the recommended trip. In future work, we plan to explore more robust and context-aware multimodal frameworks [36, 29, 28] that balance recommendation accuracy and diversity, paving the way toward more adaptive and user-centric recommendation systems.