Explicit Intent-Enhanced Knowledge Distillation for Trip Recommendation

Shuliang Wang[1], Xiaoting Leng[1], Sijie Ruan[1*], Dingqi Yang[2], Yicheng Tang[1], Qianyu Yang[1], Qianxiong Xu[3], Jiabao Zhu[1], Hanning Yuan[1*]

[1]Beijing Institute of Technology, Beijing, China
[2]University of Macau, Macao SAR, China
[3]Nanyang Technological University, Singapore
{slwang2011, lengxt, sjruan, tyc, yangqy, jiabao, yhn6}@bit.edu.cn,
dingqiyang@um.edu.mo, qianxiong.xu@ntu.edu.sg

Abstract

Trip recommendation aims to generate a sequence of points of interest (POIs) under a user's query input. Existing data-driven methods mainly fall into two categories: supervised approaches and self-supervised approaches. The former cannot fully capture the transition patterns among POIs, while the latter fail to comprehensively model user's query intents. Fortunately, privileged knowledge distillation (PKD) provides us an unique opportunity to align user's query intents with its corresponding trip in historical data. However, such knowledge alignment is implicit, which may not directly reflect the query intents. To this end, in this paper, we propose EKD-Trip, an explicit intent-enhanced knowledge distillation framework. EKD-Trip first trains a trajectory encoder (teacher model) and a trip generator jointly in a self-supervised manner. Then, a query encoder (student model) is trained via multi-task learning to extract implicit knowledge by PKD from teacher and explicit knowledge from an auxiliary task, respectively. At inference time, we use the query encoder and the trip generator to recommend trips. Extensive experiments on four real-world datasets demonstrate that EKD-Trip outperforms all baselines over three metrics, with a particularly notable improvement of 13.70% in pairs-F_1.

Code — https://github.com/stdi-lab/EKD-Trip

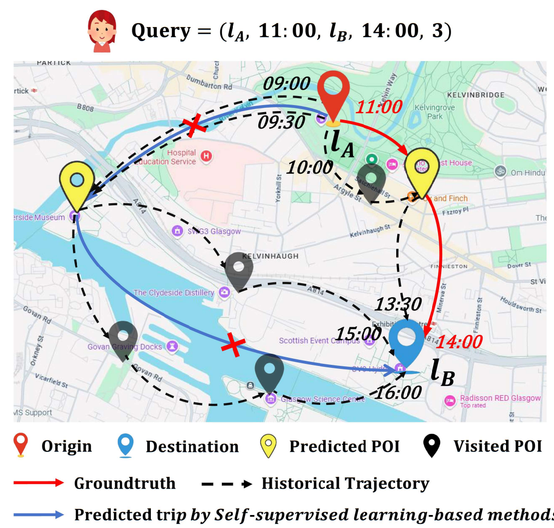


Figure 1: An example query from l_A to l_B between 11:00 and 14:00 with 3 POIs. Massive historical trajectories(the dash black routes) lead self-supervised learning-based methods to recommend a frequent route (the blue route in this case). However, due to the specific query intent, faraway POIs may lead to a violation of the constraint.

Introduction

The widespread use of GPS-enabled mobile devices and the rise of location-based social networks (LBSNs), such as Foursquare and Flickr, have led to the accumulation of massive trajectory data, i.e., check-in sequences, providing unique opportunities in gaining deep insights into human mobility and intent. This surge has driven various applications, including next POI recommendation (Sun et al. 2024; Feng et al. 2024), co-movement pattern mining (Chen et al. 2019), travel time estimation (Yang et al. 2023), and notably, trip recommendation (Kuo, Chen, and Ku 2023), which enhances user experience and fosters tourism development.

Trip recommendation typically involves generating a trip, i.e., a sequence of POIs based on a query comprising origin, destination (OD), start and end time, and desired number of visiting POIs. Existing data-driven trip recommendation methods can be mainly categorized into two types:

- **Supervised learning-based methods** (Yang et al. 2021; Shu et al. 2024; Rakesh et al. 2017; Chen, Ong, and Xie 2016), which design end-to-end models that take the query as input and give the trip as output. Those models are trained in a supervised manner. Though the error signals are propagated end-to-end, given limited samples, the transition patterns among POIs are not fully captured, which leads to limited trip recommendation capabilities.
- **Self-supervised learning-based methods** (Gao et al. 2022; Kuo, Chen, and Ku 2023; Zhou et al. 2021), which first pre-train trajectory base models via masked POI modeling or contrastive learning, and then utilize the pre-trained models to recommend trips. Though such meth-

[*]Sijie Ruan and Hanning Yuan are corresponding authors.

ods effectively exploit the transition knowledge in trajectories, the intents contained in the query are not comprehensively modeled due to the learning task discrepancies. *For example*, as shown in Figure 1, since these self-supervised learning-based methods overlook constraints like POI number and time, the recommended trip may not meet the user's actual intent.

To combine the best of two worlds, we aim to find a way to capture the query intent while preserve the transition pattern extraction capability via self-supervised learning. Recently, privileged knowledge distillation (PKD) (Lopez-Paz et al. 2016; Aslam et al. 2023; Liu et al. 2025a) has demonstrated great success in various domains. It is achieved by aligning the representation given by a student model, which takes ordinary input, with that given by a teacher model, which additionally takes some training-time available information as input, so that the student model can implicitly master those privileged information. For historical data, the query intent the query intent. However, directly employing PKD is still insufficient, since such learning strategy only makes the trajectory knowledge distillated into the student model, which cannot guarantee the distillated knowledge is directly related to query intents, which are vital for trip generation.

To this end, in this paper, we propose Explicit Intent-Enhanced Knowledge Distillation for Trip Recommendation (EKD-Trip). The main idea is that we not only employ PKD to distill the *implicit* query intents from trajectories, but also derive *explicit* intents, i.e., travel modes (e.g., visiting POIs along the way between OD), from queries to facilitate the training of the student model. More specifically, EKD-Trip comprises three key components: a trajectory encoder (teacher), a query encoder (student), and a trip generator. EKD-Trip is trained in two stages. Firstly, the trajectory encoder is jointly trained with the trip generator in a self-supervised manner to compress the trajectory knowledge as well as learn the transition knowledge in trajectories. Secondly, the query encoder (student) is trained via multi-task learning to extract implicit and explicit knowledge by PKD (from teacher) and the auxiliary task, travel mode prediction, respectively. During the inference, we combine the query encoder and the trip generator to recommend trips.

Our main contributions are summarized as follows:

- To the best of authors' knowledge, we are the first to introduce the privileged knowledge distillation (PKD) into trip recommendation.
- We propose EKD-Trip, which not only learns transition patterns via self-supervised learning, but also extracts query intents from the query both implicitly via PKD and explicitly via the auxiliary task.
- Extensive experiments on four real-world datasets over three evaluation metrics show the superiority of EKD-Trip, which outperforms all baseline methods, especially with a 13.70% average improvement in pairs-F_1.

Preliminaries

In this section, we introduce basic definitions, and formulate the problem.

Definitions

Point of Interest (POI). A POI is a geographical place, which is denoted by a triplet $l = (id, lat, lon) \in \mathcal{L}$ where id, lat and lon represent its identifier, latitude and longitude, respectively.

Check-in. A check-in is a record generated by a certain user, which is denoted by $c = (l, t)$. It indicates that POI l was visited by the user at timestamp t.

Trajectory. A trajectory is a check-in sequence of a user ordered by check-in timestamps, which is denoted by $T =< c_1, c_2, \cdots, c_n >$, where $c_1.l$ and $c_n.l$ represent the origin and destination of the trajectory, respectively, and n is the length of the trajectory.

Problem Formulation

Given historical trajectories $\mathcal{T}$, learn a trip recommender, which takes trip query $Q = (l_s, t_s, l_e, t_e, N)$ consisting of the desired origin POI l_s and start timestamp t_s, the desired destination POI l_e and end timestamp t_e, and the length of the trip N (i.e., the number of visited POIs) as input, and recommends a trip $tr =< l_1, l_2, \cdots, l_N >$ for user is of interest, where $l_1 = l_s$ and $l_N = l_e$.

Note that, in this work, we do not consider the semantic meaning of POIs, e.g., POI type, which may also benefit the trip recommendation, for the universality and consistency with existing work (Gao et al. 2019, 2021). However, we argue that the semantic meaning can be easily integrated with the proposed framework, which will be later introduced.

Methodology

In this section, we present the proposed EKD-Trip, whose framework is shown in Figure 2. We first introduce how trajectory encoder and trip generator are jointly learned via Trajectory Representation Learning. Then, we extract the implicit intents of the query encoder using Query Representation Learning via Distillation. Finally, we adjust the training objective of the query encoder to incorporate explicit intents via Mode Prediction-based Representation Enhancement. We elaborate each module in detail as follows.

Trajectory Representation Learning

In this subsection, we introduce the structure and training process of trajectory encoder and trip generator.

Trajectory Encoder. The trajectory encoder encodes a historical trajectory into a hidden representation $\mathbf{h}_T \in \mathbf{R}^H$ by capturing both the semantics of individual check-ins and their sequential dependencies.

Following existing work (Gao et al. 2019), we first use a check-in embedding layer to transform each check-in into a dense representation, which is encoded from three views: 1) spatial view, which encodes POI ID via an embedding layer; 2) temporal view, which embeds time of day via another embedding layer, and 3) contextual view, which considers its location relative to the origin and destination, i.e.,

$$\mathbf{c} = [\frac{d(c.l, l_s)}{d_{max}}\mathbf{w}_s; \frac{d(c.l, l_e)}{d_{max}}\mathbf{w}_e], \tag{1}$$

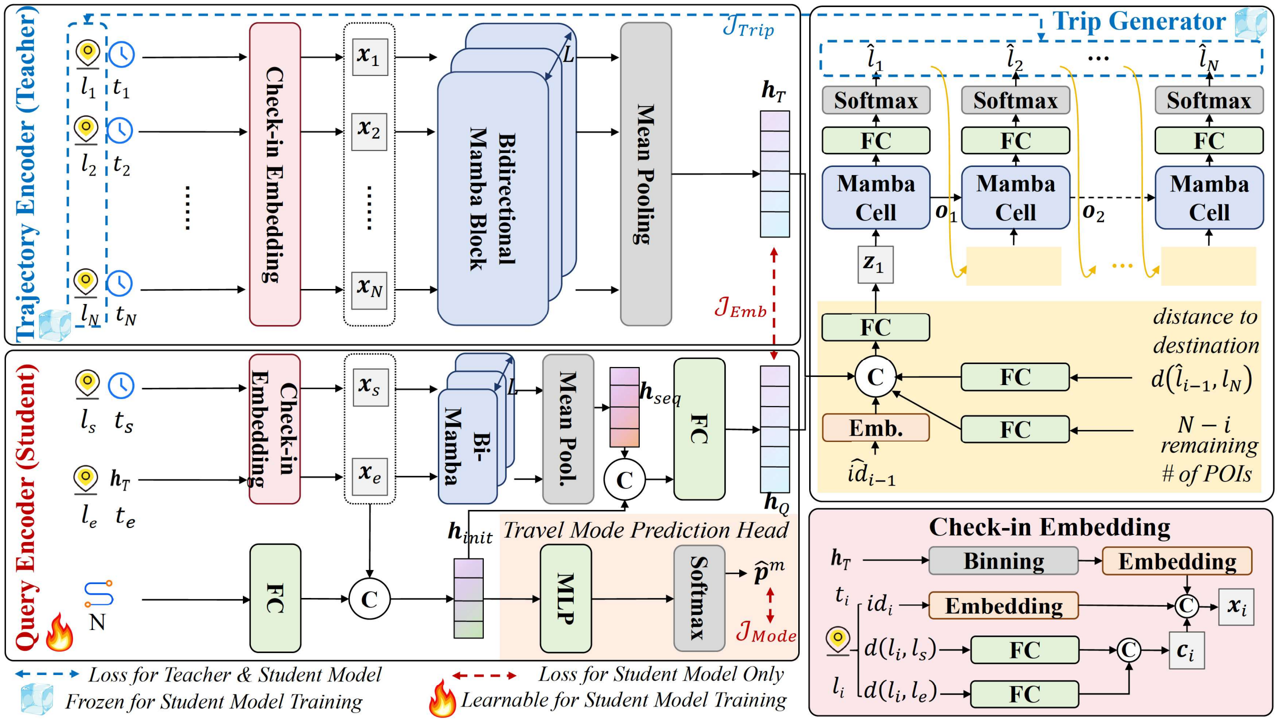


Figure 2: Framework of EKD-Trip, which is mainly composed of three modules: Trajectory Encoder (teacher), Trip Generator and Query Encoder (student). Two-stages training: (1) Firstly, the Trajectory Encoder and Trip Generator are trained in a self-supervised manner; (2) Secondly, we freeze them, and train the Query Encoder via multi-task learning (knowledge distillation and travel mode prediction). At the test time, we use Query Encoder and Trip Generator to recommend a trip.

where $\mathbf{w}_s$ and $\mathbf{w}_e$ are learnable vectors, $d(\cdot,\cdot)$ denotes geographical distance, and d_{max} is the dataset's maximum distance for normalization.

The final dense representation $\mathbf{x} \in \mathbf{R}^D$ is obtained by concatenating the three parts:

$$\mathbf{x} = [\mathtt{Emb}(id); \mathtt{Emb}(\mathtt{Bin}(t)); \mathbf{c}]. \tag{2}$$

where $\mathtt{Bin}(\cdot)$ is the binning operation. Additional semantic attributes of POIs can also be incorporated if available.

Next, we need a sequential module to model the sequential dependencies among check-in embeddings. Unlike existing work, which uses LSTM (Gao et al. 2019), GRU (Gao et al. 2022) and Transformer (Shu et al. 2024), we adopt Mamba (Gu and Dao 2023) here, which excels at refining and summarizing pertinent information rather than indiscriminately traversing all sequences (Gu and Dao 2023). It aligns well with the decision-making process of tourists who focus on key cues such as spatial proximity.

Given an embedded check-in sequence $\mathbf{T} =< \mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N >$, we apply L layers of bidirectional Mamba blocks to model both forward and reverse orderings. The final trajectory representation is obtained by averaging both directions:

$$\begin{aligned}\mathbf{h}_T = \; & \mathrm{Avg}\big(\mathrm{MambaBlock}_{\mathrm{forw}}(\mathbf{T}) \\ & + \mathrm{MambaBlock}_{\mathrm{back}}(\mathbf{T}')\big)\end{aligned} \tag{3}$$

where $\mathbf{T}'$ is the reversed sequence, $\mathtt{MambaBlock}_{\mathtt{forw}}(\cdot)$ and $\mathtt{MambaBlock}_{\mathtt{back}}(\cdot)$ denotes forward and backward directional Mamba blocks, $\mathtt{Avg}(\cdot)$ means to perform mean pooling operation on the output of the model.

Trip Generator. The trip generator takes the hidden representation $\mathbf{h}_T$ generated by the encoder and the trip length N to generate a trip tr.

A straightforward idea is to use another Mamba block to decode locations step by step, where each step takes the previously predicted location and $\mathbf{h}_T$ as input. However, this ignores the "context" that influences tourists' choices, e.g., *1) the number of remaining POIs*, and *2) the distance from the current location to the destination.*

To address this, we introduce context-aware trip generation. At each decoding step i, in addition to feeding the predicted location $\hat{id}_{i-1}$ from the previous step (at the initial step, token "GO" is fed) as well as the hidden representation $\mathbf{h}_T$, we fuse the number of available locations $N-i$ as well as the current distance to the destination l_e. Specifically, the input of the i-th decoding step $\mathbf{z}_i$ is:

$$\mathbf{z}_i = \mathtt{ReLu}(\mathtt{FC}([\mathtt{Emb}(\hat{id}_{i-1}); \mathbf{h}_T; (N-i)\mathbf{w}_n; \frac{d(l_{i-1}, l_e)}{d_{\max}}\mathbf{w}_d])). \tag{4}$$

where $\mathtt{ReLu}(\cdot)$ is the non-linear activations, and $\mathtt{FC}(\cdot)$ is the fully-connected layer. $\mathbf{w}_n$ and $\mathbf{w}_d$ are learnable vectors.

Then, $\mathbf{z}_i$ is fed into the Mamba cell to model the dependency with the previous time step. Here we only use unidirectional Mamba since we only know the preceding information.

$$\mathbf{o}_i = \texttt{MambaCell}(\mathbf{z}_i, \mathbf{o}_{i-1}), \tag{5}$$

where $\mathbf{o}_i$ is the hidden state at time step i. We also leverage $\mathbf{o}_i$ to generate the probability distribution of the recovered location $\hat{\mathbf{p}}_i$ via a classification head:

$$\hat{\mathbf{p}_i} = \texttt{Softmax}(\texttt{FC}(\mathbf{o}_i)), \tag{6}$$

where $\texttt{Softmax}(\cdot)$ is the softmax activation, $\mathbf{O} =< \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_n >$ contains all outputs from Mamba block.

Self-supervised Learning. The trajectory encoder and the trip generator are trained in a self-supervised manner like autoencoder, i.e., the trajectory encoder takes a historical trajectory $T =< c_1, c_2, \cdots, c_N >$, and generates a latent representation $\mathbf{h}_T$, and the trip generator aims to recover the trip based on $\mathbf{h}_T$ and the length of the trip N. We use the cross-entropy loss to train the model:

$$\mathcal{J}_{Trip} = -\sum_{i=1}^{N}\sum_{j=1}^{|\mathcal{L}|} y_i^j log \hat{p}_i^j. \tag{7}$$

Here, y_i^j indicates whether location j is the i-th visited location in the trip, and $\hat{p}_i^j$ is the probability that the i-th visited location is location j.

Query Representation Learning via Distillation

In this subsection, we introduce the structure and the training process of the query encoder via knowledge distillation.

Query Encoder. The query encoder aims to generate a hidden representation $\mathbf{h}_Q \in \mathbf{R}^H$ based on a trip query $Q = (l_s, t_s, l_e, t_e, N)$. Since the origin and start time as well as the destination and the end time essentially can be regarded as two check-ins. We first feed them into check-in embedding layer, and concatenate all information in Q into an initial representation $\mathbf{h}_{init}$:

$$\mathbf{h}_{init} = \texttt{FC}([N\mathbf{w}_N; \mathbf{x}_s; \mathbf{x}_e]). \tag{8}$$

Furthermore, since the origin and the destination naturally form a sequence, we use another L Bidirectional Mamba layers to capture their sequential dependency. The OD embedding $\mathbf{h}_{seq}$ is obtained in the same way as Eq. (3), except that the input is replaced with the OD pair $\langle \mathbf{x}_s, \mathbf{x}_e \rangle$.

Finally, we concatenate and fuse $\mathbf{h}_{init}$ and $\mathbf{h}_{seq}$ to obtain the query representation $\mathbf{h}_Q$:

$$\mathbf{h}_Q = \texttt{FC}([\mathbf{h}_{init}; \mathbf{h}_{seq}]). \tag{9}$$

Distillation-based Representation Learning. Unlike response-based knowledge distillation (Gou et al. 2021), which typically matches teacher and student output logits via divergence-based objectives (e.g., JSD), our framework performs feature-based distillation. Since both teacher and student produce latent vector rather than logits, we adopt Mean Squared Error (MSE) loss to directly minimize the distance between the query representation $\mathbf{h}_Q$ and the trajectory embedding $\mathbf{h}_T$, enabling the student to learn the teacher's understanding of massive trajectories:

$$\mathcal{J}_{Emb} = \frac{1}{H}\sum_{i=1}^{H}(\mathbf{h}_T^i - \mathbf{h}_Q^i)^2, \tag{10}$$

where H is the hidden dimension of $\mathbf{h}_T$ and $\mathbf{h}_Q$. Since the ultimate purpose of the query encoder is to generate an accurate trip recommendation, we also feed $\mathbf{h}_Q$ into the trip generator, whose parameters are frozen, to consider the prediction loss. We follow the privileged knowledge distillation (Aslam et al. 2023) to train the query encoder using the loss from the hidden representations $\mathcal{J}_{Emb}$ as well as the loss from the ground truth $\mathcal{J}_{Trip}$ using the following hybrid loss function:

$$\mathcal{J}_{Dis} = \alpha \mathcal{J}_{Trip} + (1-\alpha)\mathcal{J}_{Emb}, \tag{11}$$

where α is a hyperparameter.

After the query encoder is trained, we can combine it with the pre-trained trip generator to recommend trips. Note that, at the recommendation stage, we know the origin and destination, which will be set to the first and the last visited location in the prediction.

Mode Prediction-based Representation Enhancement

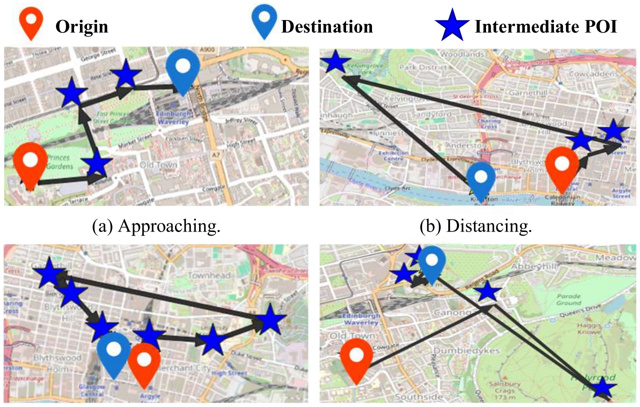
While the distillation process encourages $\mathbf{h}_Q$ to approximate $\mathbf{h}_T$, it fails to explicitly model the dependency between the query and movement dynamics. For instance, a query from hotel to airport typically reflects a trip with a progressively approaching pattern. To further enhance the capability of the query encoder, we introduce an auxiliary task, i.e., travel mode prediction, to enhance the query representation.

Firstly, we need to obtain the travel mode for each query. For the training data, the travel mode, i.e., label, can be derived from its corresponding trajectory.

Specifically, for each trajectory $T =< c_1, c_2, \cdots, c_n >$, we compute the distance vector $\mathbf{D}_T =< d_1, d_2, \cdots, d_{n-1} >$, where d_i is the distance from check-in c_i to the destination (excluding the final point). Then, T can be categorized into one of four modes as shown in Figure 3 based on the following criteria:

- **Approaching**: $\mathbf{D}_T$ shows a monotonically decreasing trend, i.e., moving consistently toward the destination.
- **Distancing**: $\mathbf{D}_T$ increases monotonically, i.e., initially moving away, then returning.
- **U-Turn**: $\mathbf{D}_T$ fits a Gaussian curve, i.e., visiting distant POIs first, then turning back to visit some POIs closer to the destination.
- **Irregular**: Patterns do not fit any above cases, i.e., no consistent spatial trend.

To introduce the auxiliary task, we use a multilayer perceptron layer to transform the initial representation $\mathbf{h}_{init}$ of the query encoder into mode classification space, and use the softmax activation to make the classification.



(a) Approaching. (b) Distancing.

(c) U-Turn. (d) Irregular.

Figure 3: Illustrations of different travel modes.

City	#Trip	#Check-in	#POI	#User
Glasgow	2,227	11,434	27	601
Osaka	1,115	7,747	27	450
Toronto	6,057	39,419	29	1,395
Tokyo	6,414	8,721	30	200

Table 1: Statistic of Datasets.

$$\hat{\mathbf{p}}^m = \texttt{Softmax}(\texttt{MLP}(\mathbf{h}_{init})), \tag{12}$$

where $\texttt{MLP}(\cdot)$ is composed of three fully-connected layers, interleaved with the ReLu activations. Again, we use the cross-entropy loss to learn the auxiliary travel mode.

$$\mathcal{J}_{Mode} = -\sum_{j=1}^{M} y_j^m log \hat{p}_j^m, \tag{13}$$

where y_j^m indicates whether mode j is the travel mode of the trip, and $\hat{p}_j^m$ is the probability that the travel mode of the current trip is mode j.

Finally, we modify the distillation loss $\mathcal{J}_{Dis}$ in Equation 11 to the multi-task loss $\mathcal{J}'_{Dis}$ to train the query encoder:

$$\mathcal{J}'_{Dis} = \lambda \mathcal{J}_{Dis} + (1-\lambda)\mathcal{J}_{Mode}, \tag{14}$$

where λ is a trade-off hyperparameter.

Experiments

In this section, we conduct extensive experiments to show the effectiveness of EKD-Trip.

Experimental Setup

Datasets. Following existing work (Gao et al. 2019, 2021; Zhou et al. 2021), we use four real-world trip datasets, including Glasgow, Osaka, Toronto and Tokyo, which are obtained from Flickr (Lim et al. 2015; Chen, Ong, and Xie 2016) and Foursquare (Yang et al. 2014). Following existing work (Kuo, Chen, and Ku 2023), we performed the five-fold cross-validation for model evaluation. The basic statistics of the datasets are shown in Table 1.

Baselines. We compare EKD-Trip with the following three types of baselines:

- **Heuristics-based methods**, which recommend POIs in a trip based on location popularity, i.e., POIPopularity (Chen, Ong, and Xie 2016).
- **Supervised learning-based methods**, which train a model that takes the query as input and the trip as output in a supervised manner, including POIRank (Chen, Ong, and Xie 2016), Markov (Chen, Ong, and Xie 2016), Rank+Markov (Chen, Ong, and Xie 2016), DeepTrip (Gao et al. 2021), AR-Trip (Shu et al. 2024).
- **Self-supervised learning-based methods**, which train a trajectory model in a self-supervised manner based on historical trips, e.g., masked language modeling or contrastive learning, and then adopts the pre-trained model to recommend trips. This type includes SelfTrip (Gao et al. 2022), CTLTR (Zhou et al. 2021), GraphTrip (Gao et al. 2023) and Bert-Trip (Kuo, Chen, and Ku 2023).

Evaluation Metrics. Following prior work (Gao et al. 2022; Kuo, Chen, and Ku 2023; Shu et al. 2024), we use F_1, pairs-F_1 and REP to evaluate the performance of different methods. F_1 and pairs-F_1 evaluate the accuracy of recommended POIs or order of the trip, respectively, and REP measures the ratio of repeating POIs in the trip. A higher F_1 and pairs-F_1, or a lower REP, indicates better performance.

Implementation & Training Details. All methods are implemented in Python with PyTorch. Experiments were conducted on a server with 56 Cores@2.6GHz, 128GB memory, and models were trained using one GeForce RTX 4090 (24GB) GPU. In EKD-Trip, we use Adam optimizer for batch gradient descent, and also set the batch size to 8. We set the learning rate of the teacher model to 0.003, and the learning rate of the student model to 0.001. The hidden dimension of the check-in embedding is $D = 64$, the hidden dimension of the trip and query representation is $H = 128$, and Mamba blocks are stacked for $L = 2$ layers. The hyperparameter of the learning objective α is set to 0.375, and λ is set to 0.6 by default.

Overall Performance

The overall performance of EKD-Trip and baselines is reported in Table 2. Based on the experimental results, we have the following observations: 1) Heuristics-based methods (POIPopularity, POIRank, Markov, Rank+Markov) perform poorly. While statistical methods yield low REP scores by reducing repetition, they rely on oversimplified assumptions and fail to model location dependencies, resulting in low F_1 and Pairs-F_1. 2) Self-supervised learning-based methods (e.g., SelfTrip, Bert-Trip) generally outperform supervised learning-based methods (e.g., DeepTrip, AR-Trip) by better capturing transition patterns. Notably, AR-Trip achieves competitive REP and F_1 on some datasets due to its cycle-aware design and training strategy. 3) Benefiting from the human transition pattern extracted from historical trajectories and explicit intent learned from trip queries, EKD-Trip outperforms all baselines over three metrics, improves the

Methods	Glasgow			Osaka			Toronto			Tokyo		
	$F_1\uparrow$	pairs-$F_1\uparrow$	$REP\downarrow$	$F_1\uparrow$	pairs-$F_1\uparrow$	$REP\downarrow$	$F_1\uparrow$	pairs-$F_1\uparrow$	$REP\downarrow$	$F_1\uparrow$	pairs-$F_1\uparrow$	$REP\downarrow$
POIPopularity	0.508	0.191	-	0.424	0.223	-	0.472	0.235	-	0.511	0.196	-
POIRank	0.576	0.241	-	0.476	0.306	-	0.559	0.334	-	0.521	0.252	-
Markov	0.522	0.185	0.068	0.429	0.262	0.039	0.454	0.287	0.072	0.456	0.174	0.083
Rank+Markov	0.564	0.229	0.073	0.469	0.298	0.042	0.531	0.318	0.067	0.48	0.299	0.081
DeepTrip	0.667	0.334	0.238	0.679	0.417	0.175	0.667	0.433	0.197	0.738	0.516	0.244
AR-Trip	0.796	0.546	0.092	0.758	0.562	0.033	0.728	0.532	0.065	0.631	0.537	0.079
SelfTrip	0.776	0.604	0.093	0.741	0.579	0.063	0.638	0.576	0.104	0.653	0.551	0.098
CTLTR	0.625	0.309	0.138	0.637	0.381	0.125	0.672	0.369	0.167	0.775	0.508	0.142
GraphTrip	0.724	0.431	0.147	0.687	0.469	0.108	0.705	0.458	0.158	0.695	0.497	0.187
Bert-Trip	0.789	0.612	0.101	0.752	0.537	0.047	0.767	0.571	0.074	0.790	0.591	0.153
EKD-Trip (Ours)	**0.813**	**0.688**	**0.063**	**0.833**	**0.708**	**0.031**	**0.864**	**0.783**	**0.045**	**0.859**	**0.734**	**0.073**

Table 2: Results comparisons among different methods in four cities.The best results are marked with bold face, and the second best results are underlined.A hyphen (-) indicates the use of statistical methods to prevent duplication.

accuracy of trip recommendation through knowledge distillation, and effectively reduces the repetitiveness of POIs through the context-aware mechanism in the trip generator. Overall, EKD-Trip consistently outperforms all baseline methods over three metrics, especially with a 13.70% average improvement in pairs-F_1.

Ablation Studies

To assess the contribution of key components in EKD-Trip, we conduct an ablation study with the following variants: 1) **w/o KD**: removes knowledge distillation, using only the student model; 2) **w/o TMPH**: removes the travel mode prediction head; 3) **w/o CA**: removes the context-aware mechanism in the trip generator; 4) **w/o Mamba**: replaces Bi-Mamba with Transformer; 5) **w/o QE**: removes the query encoder and uses only the pre-trained trajectory encoder and generator; 6) **Bert-Trip+TMPH**: adds our travel mode prediction to Bert-Trip for comparison.

Due to space limits, we show results on Toronto and Tokyo in Figure 4, the trends on other datasets are similar.

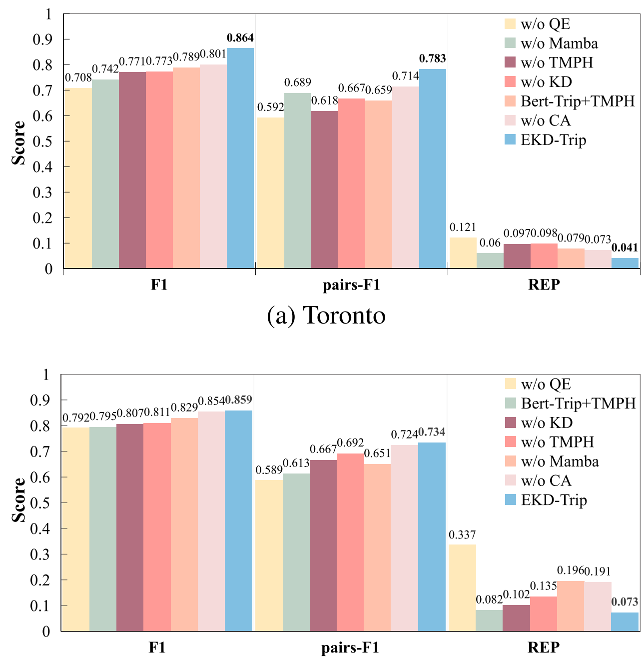
First of all, all of our design choices contribute to the overall performance improvement.

Without KD, model underperforms, confirming that distilling human transition patterns from trajectories helps the student encoder better capture implicit user intent.

The performance drops after removing TMPH, which highlights the importance of capturing the explicit travel mode from trip queries. Travel mode prediction plays an important role in enhancing the learning of the query encoder.

After removing QE, the performance drops significantly, indicating that learning transition patterns from historical trajectories alone is not sufficient for trip recommendation. Our dedicated query encoder is crucial for understanding users' implicit and explicit query intent. After replacing Mamba with Transformer, the performance also drops, which is particularly evident in the F_1 score of the Toronto dataset. Since there is a difference between the distributions of the Flickr and Foursquare datasets, and the trip lengths in the Tokyo dataset are more balanced and longer, we believe this is due to the stability and effectiveness of Mamba on long sequence modeling tasks.

Removing CA, we find the recommendation quality of the trip generator also worsened, demonstrating the necessity of incorporating "remaining context" into the trip generation.



(a) Toronto

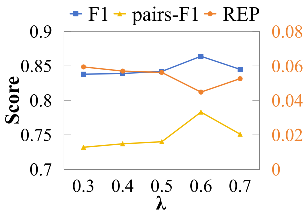
(b) Tokyo

Figure 4: Ablation studies.

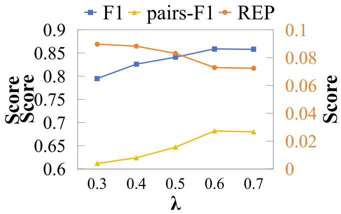
Finally, Bert-Trip+TMPH improves upon its base model, confirming the value of travel mode prediction, but still lags behind EKD-Trip due to lack of fusion between travel mode and transition pattern.

Effect of Multi-task Learning

We examine the impact of the trade-off hyperparameter λ between knowledge distillation and travel mode prediction by varying λ from 0.3 to 0.7. As shown in Figure 5, performance (in terms of F_1 and pairs-F_1) first improves and then declines, peaking at $\lambda = 0.6$. This suggests that balancing both objectives is essential, while overemphasizing either one harms performance.



(a) Effect of λ on Toronto



(b) Effect of λ on Tokyo

Figure 5: Parameter experiments.

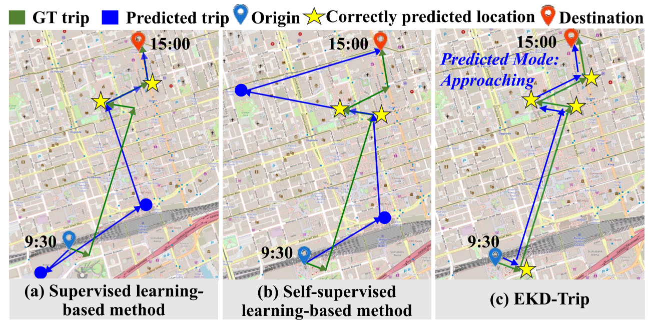


Figure 6: Case study of EKD-Trip.

Case Study

To intuitively demonstrate the effectiveness of EKD-Trip, we present a case study on the Toronto dataset. As shown in Figure 6, the user specifies the origin (the blue dot), destination (the red dot), start and end time (9:30 and 15:00 in this case), and desired number of POIs (6 in this case).

Figure 6(a) shows the trip generated by a supervised learning-based method (AR-Trip). While it correctly predicts the final two POIs, the first part significantly deviates from the ground-truth. This may result from its limited ability to model transition patterns. The latter part aligns better, possibly benefiting from the destination constraint—underscoring the importance of query intents.

Figure 6(b) illustrates the result from a self-supervised learning-based method (Bert-Trip). Ignoring the current query's constraints, the model outputs a trip with POIs scattered far apart, some even violating the time window, highlighting its lack of query-awareness.

Instead, Figure 6(c) shows the result from EKD-Trip. By fusing global transition patterns with query intent, and correctly inferring the travel mode (Approaching), it generates a coherent route closely aligned with the actual trajectory.

Related Work

Trip Recommendation. Trip recommendation aims to generate complete travel routes and is typically categorized into optimization-based and learning-based methods. The former treats the task as an orienteering problem (OP), scoring POIs to find a high-utility path (Chen, Ong, and Xie 2016; Lim et al. 2015; He, Qi, and Ramamohanarao 2019). However, such methods rely on heuristic scores and suffer from inefficiency due to the NP-hard nature of OP.

Learning-based methods have become dominant. Early work (e.g., RankMarkov (Chen, Ong, and Xie 2016)) used Markov models. Later, deep learning models including RNN-based (Yang et al. 2021), Transformer-based (Shu et al. 2024), and contrastive/self-supervised approaches (Zhou et al. 2021; Gao et al. 2022; Kuo, Chen, and Ku 2023) were proposed to better capture POI transitions using historical trajectories.

However, these methods either not fully capture the transition pattern nor comprehensively model user's query intent. Our method addresses both by explicit intent-enhanced knowledge distillation.

Recently, LLM-based recommenders (Chen et al. 2024) offer strong generalization, while our focus lies in enhancing accuracy via structured modeling.

Knowledge Distillation. Knowledge distillation (Hinton 2015) trains a compact student to mimic a more powerful teacher, often yielding better performance than direct training (Alkhulaifi, Alsahli, and Ahmad 2021). Applications span weather forecasting (He, Ji, and Lei 2024) and logistics (Zhang et al. 2024). Recent advances introduce privileged information (Lopez-Paz et al. 2016), where teachers access extra training-time data to enhance student learning (Aslam et al. 2023). We adopt this approach in trip recommendation, enabling the student to absorb trajectory-level patterns unavailable at inference.

Human Trajectory Modeling. Related tasks include prediction, generation, and recovery. Trajectory prediction forecasts the next POI based on past movements, using RNNs (Feng et al. 2018), Transformers (Lian et al. 2020; Sun et al. 2024), LLMs (Gong et al. 2024), and test-time adaptation methods such as AdaMove (Han et al. 2025). Generation methods (Gong et al. 2025; Zhu et al. 2024) simulate plausible trajectories via diffusion models. Recovery aims to impute missing locations, leveraging attention (Xia et al. 2021), GNNs (Sun et al. 2021; Deng et al. 2023), road-constrained models (Ren et al. 2021; Chen et al. 2023), and diffusion-based approaches (Liu et al. 2025b). Trip recommendation can be viewed as a special case of trajectory recovery. Unlike traditional recovery, which assumes moderate missing rates (e.g., 20% (Xia et al. 2021)), EKD-Trip handles the extreme case—only origin and destination are known—by aligning query and trip semantics via knowledge distillation and travel mode prediction.

Conclusion

In this paper, we propose EKD-Trip, an explicit intent-enhanced knowledge distillation framework. By introducing the privileged knowledge distillation (PKD) mechanism and the auxiliary task of travel mode prediction, EKD-Trip effectively integrates the implicit intent in trajectory knowledge with the explicit intent in the query, fully capturing user intent and improving the accuracy of trip recommendations. Experiments on four real-world datasets show EKD-Trip outperforms the best baseline by 13.70% in pairs-F_1. Future work will explore incorporating finer-grained POI semantics for better personalization.

Acknowledgments

This research is supported by National Natural Science Foundation of China (No. 62306033, 42371480), and the Science and Technology Development Fund, Macau SAR (0011/2025/RIB1, 001/2024/SKL).

References

Alkhulaifi, A.; Alsahli, F.; and Ahmad, I. 2021. Knowledge distillation in deep learning and its applications. *PeerJ Computer Science*, 7: e474.

Aslam, M. H.; Zeeshan, M. O.; Pedersoli, M.; Koerich, A. L.; Bacon, S.; and Granger, E. 2023. Privileged knowledge distillation for dimensional emotion recognition in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3338–3347.

Chen, A.; Ge, X.; Fu, Z.; Xiao, Y.; and Chen, J. 2024. TravelAgent: An AI assistant for personalized travel planning. *arXiv preprint arXiv:2409.08069*.

Chen, D.; Ong, C. S.; and Xie, L. 2016. Learning points and routes to recommend trajectories. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2227–2232.

Chen, L.; Gao, Y.; Fang, Z.; Miao, X.; Jensen, C. S.; and Guo, C. 2019. Real-time distributed co-movement pattern detection on streaming trajectories. *Proceedings of the VLDB Endowment*, 12(10): 1208–1220.

Chen, Y.; Zhang, H.; Sun, W.; and Zheng, B. 2023. Rntrajrec: Road network enhanced trajectory recovery with spatial-temporal transformer. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 829–842. IEEE.

Deng, L.; Zhao, Y.; Sun, H.; Yang, C.; Xie, J.; and Zheng, K. 2023. Fusing local and global mobility patterns for trajectory recovery. In *International Conference on Database Systems for Advanced Applications*, 448–463. Springer.

Feng, J.; Li, Y.; Zhang, C.; Sun, F.; Meng, F.; Guo, A.; and Jin, D. 2018. Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, 1459–1468.

Feng, S.; Meng, F.; Chen, L.; Shang, S.; and Ong, Y. S. 2024. Rotan: A rotation-based temporal attention network for time-specific next poi recommendation. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*, 759–770.

Gao, Q.; Trajcevski, G.; Zhou, F.; Zhang, K.; Zhong, T.; and Zhang, F. 2019. DeepTrip: Adversarially understanding human mobility for trip recommendation. In *Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems*, 444–447.

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

Gao, Q.; Wang, W.; Zhang, K.; Yang, X.; Miao, C.; and Li, T. 2022. Self-supervised representation learning for trip recommendation. *Knowledge-Based Systems*, 247: 108791.

Gao, Q.; Zhou, F.; Zhang, K.; Zhang, F.; and Trajcevski, G. 2021. Adversarial human trajectory learning for trip recommendation. *IEEE Transactions on Neural Networks and Learning Systems*, 34(4): 1764–1776.

Gong, L.; Guo, S.; Lin, Y.; Liu, Y.; Zheng, E.; Shuang, Y.; Lin, Y.; Hu, J.; and Wan, H. 2025. STCDM: Spatio-Temporal Contrastive Diffusion Model for Check-In Sequence Generation. *IEEE Transactions on Knowledge and Data Engineering*.

Gong, L.; Lin, Y.; Zhang, X.; Lu, Y.; Han, X.; Liu, Y.; Guo, S.; Lin, Y.; and Wan, H. 2024. Mobility-llm: Learning visiting intentions and travel preference from human mobility data with large language models. *Advances in Neural Information Processing Systems*, 37: 36185–36217.

Gou, J.; Yu, B.; Maybank, S. J.; and Tao, D. 2021. Knowledge distillation: A survey. *International journal of computer vision*, 129(6): 1789–1819.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Han, H.; Wang, S.; Ruan, S.; Yang, Q.; Liang, Y.; Yuan, Z.; Long, C.; Yuan, H.; and Zheng, Y. 2025. Adamove: Efficient test-time adaptation for human mobility prediction. In *2025 IEEE 41st International Conference on Data Engineering (ICDE)*, 3655–3667. IEEE.

He, J.; Ji, J.; and Lei, M. 2024. Spatio-temporal transformer network with physical knowledge distillation for weather forecasting. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, 819–828.

He, J.; Qi, J.; and Ramamohanarao, K. 2019. A joint context-aware embedding for trip recommendations. In *2019 IEEE 35th international conference on data engineering (ICDE)*, 292–303. IEEE.

Hinton, G. 2015. Distilling the Knowledge in a Neural Network. *arXiv preprint arXiv:1503.02531*.

Kuo, A.-T.; Chen, H.; and Ku, W.-S. 2023. Bert-trip: Effective and scalable trip representation using attentive contrast learning. In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*, 612–623. IEEE Computer Society.

Lian, D.; Wu, Y.; Ge, Y.; Xie, X.; and Chen, E. 2020. Geography-aware sequential location recommendation. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2009–2019.

Lim, K. H.; Chan, J.; Leckie, C.; and Karunasekera, S. 2015. Personalized tour recommendation based on user interests and points of interest visit durations. In *Proceedings of the 24th International Conference on Artificial Intelligence*, 1778–1784.

Liu, C.; Miao, H.; Xu, Q.; Zhou, S.; Long, C.; Zhao, Y.; Li, Z.; and Zhao, R. 2025a. Efficient multivariate time series forecasting via calibrated language models with privileged knowledge distillation. In *ICDE*.

Liu, K.; Ruan, S.; Zhang, L.; Long, C.; Wang, S.; and liang yu. 2025b. Generative Human Trajectory Recovery via

Embedding-Space Conditional Diffusion. In *Forty-second International Conference on Machine Learning*.

Lopez-Paz, D.; Bottou, L.; Schölkopf, B.; and Vapnik, V. 2016. Unifying distillation and privileged information. In *International Conference on Learning Representations*.

Rakesh, V.; Jadhav, N.; Kotov, A.; and Reddy, C. K. 2017. Probabilistic social sequential model for tour recommendation. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 631–640.

Ren, H.; Ruan, S.; Li, Y.; Bao, J.; Meng, C.; Li, R.; and Zheng, Y. 2021. Mtrajrec: Map-constrained trajectory recovery via seq2seq multi-task learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1410–1419.

Shu, W.; Xu, K.; Tai, W.; Zhong, T.; Wang, Y.; and Zhou, F. 2024. Analyzing and mitigating repetitions in trip recommendation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2276–2280.

Sun, H.; Yang, C.; Deng, L.; Zhou, F.; Huang, F.; and Zheng, K. 2021. Periodicmove: Shift-aware human mobility recovery with graph neural network. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 1734–1743.

Sun, T.; Fu, K.; Huang, W.; Zhao, K.; Gong, Y.; and Chen, M. 2024. Going where, by whom, and at what time: Next location prediction considering user preference and temporal regularity. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2784–2793.

Xia, T.; Qi, Y.; Feng, J.; Xu, F.; Sun, F.; Guo, D.; and Li, Y. 2021. Attnmove: History enhanced trajectory recovery via attentional network. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 4494–4502.

Yang, C.; Zhang, J.; Gao, X.; and Chen, G. 2021. Mattrip: Multi-functional attention-based neural network for semantic travel route recommendation. In *2021 IEEE International Conference on Web Services (ICWS)*, 145–154. IEEE.

Yang, D.; Zhang, D.; Zheng, V. W.; and Yu, Z. 2014. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(1): 129–142.

Yang, S. B.; Hu, J.; Guo, C.; Yang, B.; and Jensen, C. S. 2023. Lightpath: Lightweight and scalable path representation learning. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2999–3010.

Zhang, L.; Liu, Y.; Zeng, Z.; Cao, Y.; Wu, X.; Xu, Y.; Shen, Z.; and Cui, L. 2024. Package arrival time prediction via knowledge distillation graph neural network. *ACM Transactions on Knowledge Discovery from Data*, 18(5): 1–19.

Zhou, F.; Wang, P.; Xu, X.; Tai, W.; and Trajcevski, G. 2021. Contrastive trajectory learning for tour recommendation. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(1): 1–25.

Zhu, Y.; Yu, J. J.; Zhao, X.; Liu, Q.; Ye, Y.; Chen, W.; Zhang, Z.; Wei, X.; and Liang, Y. 2024. Controltraj: Controllable trajectory generation with topology-constrained diffusion model. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 4676–4687.