Contrastive Trajectory Learning for Tour Recommendation

FAN ZHOU, PENGYU WANG, XOVEE XU, and WENXIN TAI, University of Electronic Science and Technology of China
GOCE TRAJCEVSKI, Iowa State University

The main objective of Personalized Tour Recommendation (PTR) is to generate a sequence of point-of-interest (POIs) for a particular tourist, according to the user-specific constraints such as duration time, start and end points, the number of attractions planned to visit, and so on. Previous PTR solutions are based on either heuristics for solving the orienteering problem to maximize a global reward with a specified budget or approaches attempting to learn user visiting preferences and transition patterns with the stochastic process or recurrent neural networks. However, existing learning methodologies rely on historical trips to train the model and use the next visited POI as the supervised signal, which may not fully capture the coherence of preferences and thus recommend similar trips to different users, primarily due to the data sparsity problem and long-tailed distribution of POI popularity. This work presents a novel tour recommendation model by distilling knowledge and supervision signals from the trips in a self-supervised manner. We propose Contrastive Trajectory Learning for Tour Recommendation (CTLTR), which utilizes the intrinsic POI dependencies and traveling intent to discover extra knowledge and augments the sparse data via pre-training auxiliary self-supervised objectives. CTLTR provides a principled way to characterize the inherent data correlations while tackling the implicit feedback and weak supervision problems by learning robust representations applicable for tour planning. We introduce a hierarchical recurrent encoder-decoder to identify tourists' intentions and use the contrastive loss to discover subsequence semantics and their sequential patterns through maximizing the mutual information. Additionally, we observe that a data augmentation step as the preliminary of contrastive learning can solve the overfitting issue resulting from data sparsity. We conduct extensive experiments on a range of real-world datasets and demonstrate that our model can significantly improve the recommendation performance over the state-of-the-art baselines in terms of both recommendation accuracy and visiting orders.

CCS Concepts: • **Information systems** → **Location based services**; **Social recommendation**; • **Computing methodologies** → *Knowledge representation and reasoning;*

Additional Key Words and Phrases: Tour recommendation, trip planning, contrastive self-supervised learning

This work was supported by National Natural Science Foundation of China (Grant No. 62072077 and No. 62176043), NSF Grant SWIFT 2030249, and Sichuan Science and Technology Program (2020YFG0234).
Authors' addresses: F. Zhou, P. Wang, X. Xu (corresponding author), and W. Tai, University of Electronic Science and Technology of China, Chengdu, No. 4, Section 2, N Jianshe Rd, Chenghua District, Chengdu, Sichuan, China, 610054; emails: fan.zhou@uestc.edu.cn, p.y.wang@std.uestc.edu.cn, xovee@ieee.org, wxtai@std.uestc.edu.cn; G. Trajcevski, Iowa State University, 347 Durham, 613 Morrill Rd, Ames, IA, 50011-2100; email: gocet25@iastate.edu.

2157-6904/2021/11-ART4 $15.00
https://doi.org/10.1145/3462331

ACM Reference format:
Fan Zhou, Pengyu Wang, Xovee Xu, Wenxin Tai, and Goce Trajcevski. 2021. Contrastive Trajectory Learning for Tour Recommendation. *ACM Trans. Intell. Syst. Technol.* 13, 1, Article 4 (November 2021), 25 pages.
https://doi.org/10.1145/3462331

1 INTRODUCTION

The rapid development of the mobile Internet and the increased popularity of GPS-enabled devices enabled the generation of vast geo-tagged data from users of location-based social networks (e.g., Foursquare, Flickr, WeChat, Instagram, and Yelp) [1, 62]. The digital footprints left by users provided the opportunity of exploring human mobility patterns for a variety of downstream applications, spanning from next check-in prediction [14] and spatial crowdsourcing [10, 49] to next location recommendation [44, 62] and travel time estimation [32].

Tour recommendation (a.k.a. trip, travel, or itinerary planning) aims to generate a sequence of **point-of-interests (POIs)** for a particular user, given the user-specific constraints, e.g., duration time, starting and ending points, and the number of visiting places. This type of functionality is of great importance for people visiting unfamiliar cities and may facilitate other services such as intelligent map navigation and ride-sharing optimization. In the last decade, tour recommendation has received considerable attention from both industry [4, 26] and academia [5, 17, 24, 33, 35, 65]. However, planning reasonable and user-accommodating itineraries at individual level is a non-trivial task, primarily because of the different interests and trip-budgets of different tourists. Examples include: spatial distance constraints [5], the attractiveness of POIs [26], duration time [35], combining the distance-limits with diversity of POIs [46], and even incorporating the queuing time in attractions [33].

Existing studies in tour recommendation can be classified into two main categories: (1) solving the optimization problem given the trip budget and semantic information [3, 16, 20, 33, 35, 46]; and (2) learning the visiting preferences of tourists [4, 5, 8, 17, 38, 53, 61, 65]. Traditionally, tour recommendation approaches plan trips by maximizing the user demands through solving the **Orienteering Problem (OP)** [22], which is originated from the sport of navigation that requires participants to visit all control points as fast as possible. For example, modeling the constraints of POI categories and visiting order of tourists and solve the OP problem using a dynamic programming approach is presented in Reference [20]. CLIP [3] clusters POIs using a bottom-up approach and recommends a subset of POIs from each cluster through maximizing the obtained utility score. Later OP-based tour recommender systems such as TripBuilder [4], DAR-tree [46], PersTour [35], and PersQ [33] consider additional trip constraints, including POI popularity and coverage, semantic diversity, visit duration time, user demographics, queuing time, and so on. While maximizing the planned itinerary reward, these approaches cannot be scaled to larger datasets as OP is an NP-hard problem, and the computational cost of the exact solution increases exponentially with the number of candidate POIs. Besides, these methods do not explicitly consider users' interest in attractions and visiting patterns, thereby providing a low degree of personalization for individual tourists.

As massive traces become available from social media and ubiquitous location tracking devices, researchers begin to use data-driven approaches for personalized tour recommendations. Combining the information about locations, POI categories, and historical trajectories were used in Reference [5] to plan the tour based on POI ranking and transition patterns, modeling the sequential dependencies between POIs with classical **Markov chain (MC)**. Recent advances in neural networks have inspired several deep learning models for tour recommendation [11, 17, 24, 65]. For example, TRED [65] models the temporal and sequential patterns of user trips using **recurrent**

neural networks (RNN) and distinguishes the dependencies among attractions with the attention mechanism. It trains an end-to-end model to recommend tours given the starting and ending points, as well as the number of visiting POIs. DeepTrip [17] is an adversarial learning-based tour recommender system that regularizes user trajectories' latent space and generalizes users' complex visiting preference for improving tour planning performance. In addition to the capability to extract mobility patterns from vast amounts of trips, the deep tour recommendation methods exclude the effort of hand-crafted feature engineering.

Despite the promising results enabled by the recent approaches, there remain three key challenges in planning satisfactory tours: (1) Sequential models such as MC and RNN rely on the next POI/trip prediction loss to train the model or learn trajectory representations [5, 17, 65], which is prone to overfitting problem, especially when the trajectory data are sparse. This may happen due to the overemphasis of final output results in the underlying sequential models that take a whole trip as input, which may not fully explore the rich contexts and transition patterns inherent in the trips. (2) Previous models are either trained with a *single* objective function (e.g., the cross-entropy loss) [5] or enhanced by auxiliary data representation methods such as POI embedding [11], trip reconstruction [65], and adversarial latent space learning [17]. Nevertheless, these approaches usually fail to capture long-term dependencies between POIs that reflect users' real visiting interest due to the limitations of underlying sequential models, e.g., MC's memoryless property and gradient vanishing issue in RNNs. The myopic problem not only hinders learning meaningful motion transition patterns but also prevents the model from recommending diverse trips. (3) Existing methods are vulnerable to unsatisfied visits as there is usually no explicit feedback (e.g., ratings) over the visited POIs in the training data. Consequently, the model may not distinguish the confounding bias if merely trained by the sparse user trajectories.

To address the above issues, we propose a novel tour recommendation method **Contrastive Trajectory Learning for Tour Recommendation (CTLTR)**, which is inspired by the recent positive impacts of **self-supervised learning (SSL)** in **computer vision (CV)** [7, 25, 30, 48] and **natural language processing (NLP)** [13, 40, 51], showing a comparable performance as supervised approaches in a range of image recognition and NLP tasks [36]. CTLTR is a deep neural network based recommendation model, leveraging RNN as the basic building block but introducing complementary training objectives to improve the model performance. Specifically, CTLTR splits each trajectory into multiple subsequences in a recursive manner, which preserves the coherent motion patterns and significantly augments the trajectory training data. In this way, the model is encouraged to address the overfitting problem with more data while sustaining intrinsic in-trip contexts and transitions instead of solely relying on the final layer of RNNs. Besides, it enables overcoming the long-term dependency obstacles and eliminating the requirement of extra training data or labels. Moreover, we distill supervision signals from the tour data itself and enhance the data representation through mutual information estimation and maximization. Our method can discriminate the (augmented) trajectories from the same tour against others using the contrastive loss. We propose a new auxiliary training objective to enhance the recommendation accuracy and enrich trip representations for generating diverse tours through additional signals from self-supervision. Meanwhile, our model is more robust to noisy implicit feedback due to the augmented data and auxiliary training objectives that could mitigate the impact of confounding factors such as incidental or unwilling visits. The contributions of this article are threefold:

- We propose a generic and effective data-driven tour recommendation model CTLTR that captures semantic sub-trajectories and corresponding sequential patterns in an end-to-end manner. Our model is general and can be easily generalized to various mobility learning tasks.

- We introduce a concise trip data augmentation method, which is crucial in defining contrastive loss between positive and negative training samples, enabling us to overcome fundamental problems in previous tour recommendation approaches such as overfitting, robustness, and data sparsity.
- We present a mutual information-based self-supervised pre-training diagram to better capture tourists' transition patterns and interest preferences. To our knowledge, we are the first to employ SSL to explore human mobility patterns and generate high-quality representations to enhance the pure sequential methods.

In addition, we conducted comprehensive evaluations on several real-world tour recommendation datasets. The experimental results demonstrate that our model achieves significant improvements in terms of both tour planning accuracy and order of visits; specifically, CTLTR improves the $\mathbf{F}_1$ and pairs-$\mathbf{F}_1$ scores up to 6.7% and 9.8%, respectively, compared to the state-of-the-art baseline approaches.

The remainder of this article is organized as follows. In Section 2, we review the state-of-the-art models for personalized tour recommendation. In Section 3, we introduce necessary preliminaries and background information regarding mutual information and contrastive learning. We present the details of our proposed CTLTR model as well as how we utilize trajectory data augmentation and conduct model pre-training and fine-tuning in Section 4. We experimentally evaluate the benefits of CTLTR and show the results in Section 5. Finally, Section 6 concludes this article and points directions for future work.

2 RELATED WORK

We now review the related studies from two global perspectives and position our contribution in that context.

2.1 Tour Recommendation

Tour recommendation (TR) refers to customizing trajectory plans for users, generally including starting location, destination(s), and the number of places to visit—typically accompanied by other constraints such as specified times, budget, transportation means, and so on. Devising better itineraries catering to certain criteria/constraints for users is of interest in both offline and online mode, and in both personal settings as well as travel agencies planning and has received increased attention in the research community [34]. In addition, it is an integral component in other problem domains, such as spatial crowdsourcing [49], information diffusion in social networks [66], traffic forecasting [67, 68], and various mobility-related urban modeling tasks (e.g., routing delivery vehicles) [9].

Traditional tour recommendation algorithms [3, 4, 12, 16, 18, 20, 33, 35, 50, 59] are developed to solve the OP—a specialized instance of the *traveling salesman problem*—and its more complex variants [22]. The main objective is to plan an itinerary that maximizes a global reward under the given budget, e.g., cost, travel and duration time, and so on. Due to the NP-hardness of the complexity of OP, previous TR models seek efficient approximation to the optimal solution. For example, Reference [12] presents a construction of a POI graph based on the extracted users' photos from Flickr and plans the itineraries constrained by time and destinations using a recursive greedy algorithm. Different categories of venues and the order of user visits have been considered in Reference [20] when planning the trip, while POI popularity and topics have also been used for maximizing the profit of an itinerary in the TripBuilder [4].

However, these planning-based methods usually ignore user-specific preferences, which leads to generating the same tour for multiple users and lacks the capability to recommend personalized

tour itineraries. Complementary to this, approaches such as DAR-tree [46] involve the overheads of query processing relying on generating index structures that integrate heterogeneous data.

Later OP-based methods try to model personalized user interest with various features extracted from geotagged photos that consist of spatial and temporal travel sequences. For example, PersTour [35] recommends personalized itineraries considering user interest with a variable visit duration in POIs and solves the problem using an integer programming algorithm. Later, the authors proposed PersQ [33], which considers the queue time in attractions and uses Monte Carlo tree to search for the optimal tours. Balancing the quality of attractions on different visiting days was addressed in Reference [16], and Reference [26] focused on identifying the critical attributes of route attractiveness. Recently, TRAR [21] approach derives the attractive routes and recommends the tour that can maximize user experience along the trip. Although these studies incorporate tourists' personalized interest, they cannot be extended to large-scale datasets as a consequence of solving the NP-hard OP problem.

Another line of studies aims at learning tourist interest directly from the data using various machine learning techniques. Earlier efforts [19, 37, 38] exploit collaborative filtering to learn user interest, which, however, might not plan reasonable tours due to the substantial difference between tour recommendation and single POI recommendation. Later, researchers turn to model user interest and transition patterns with sequential models. For example, Reference [5] proposes to learn user historical trajectory using Markov-based models and demonstrates that learning-based methods outperform traditional heuristic trip recommendations.

Due to the impressive performance achieved by **deep neural networks (DNN)** in a broad range of tasks, a few recent works attempted to capture complex trajectory data and human mobility with various DNNs. DeepTrip [17] models historical check-in sequence with RNN and adopts an auxiliary network to learn the latent representations of tourists' trajectories, which are continuously enhanced through adversarial training. Pre-training the POI embedding using word2vec [40] to jointly learn the relationships between users and POIs via Bayesian pairwise ranking is discussed in Reference [24]. Location embedding and trajectory embedding have been studied in References [11, 63–65, 69], where the general idea is to capture users' transition patterns directly from the data and recommend tours in an end-to-end manner. Compared to planning-based methods, learning-based methods aim to learn a customized and unique tour itinerary for each tourist, according to the user's interest and preferences, as well as the spatial-temporal constraints [34]. Nevertheless, these works either learn local transitional patterns between attraction pairs with sequential models or encode the whole trajectory to learn the latent distribution, which may not fully capture the intrinsic tourists' interests due to the sparse and implicit user feedback. These works are mainly inspired by the recurrent neural models used in language modeling to recommend a sequence of POIs in a sequential generation manner. Unlike sufficient training corpus in language training, tour recommendation is severely restricted by the data sparsity problem and the weak supervision signals due to the lack of explicit rating/comment on the visited attractions.

Table 1 summarizes the main works in tour recommendations that are most closely related to ours work. We note that many of them rely on supervised training and seldom consider the sub-trajectories as well as relative distance and trajectory data augmentation. The second column of Table 1 indicates the fundamental techniques used in each paper, and the rest of the columns list different contexts exploited in generating/recommending routes in the respective works. It is also worthwhile to note that tour recommendation studied in this article is different from works such as References [6, 52], which focus on searching or planning the best (e.g., shortest) paths on the road networks.

Table 1. Summary of the Main Studies in Tour Recommendation

Reference	Technique	POI Location	User Interest	Sub-Trajectory	Relative Distance	POI Popularity	Trajectory Augmentation	Temporal -based
2010 [38]	Dynamic programming	✓	✓			✓		✓
2010 [12]	Greedy algorithm	✓	✓			✓		✓
2011 [50]	Greedy algorithm	✓	✓			✓		✓
2011 [19]	Collaborative filtering	✓	✓			✓		✓
2011 [8]	Bayesian learning model	✓	✓	✓		✓		✓
2012 [37]	Collaborative filtering	✓	✓			✓		✓
2013 [4]	Maximum coverage problem	✓	✓			✓		✓
2014 [20]	Dynamic programming	✓	✓			✓		✓
2014 [3]	Agglomerative clustering	✓	✓	✓				✓
2015 [59]	Collaborative filtering	✓	✓			✓		✓
2015 [35]	Greedy algorithm	✓	✓			✓		✓
2016 [5]	Markov chain	✓	✓			✓		✓
2017 [33]	Monte Carlo tree search	✓	✓			✓		✓
2017 [61]	Collaborative filtering	✓	✓			✓		✓
2018 [16]	Greedy algorithm	✓	✓			✓		✓
2019 [17]	Generative adversarial networks	✓	✓		✓	✓		✓
2019 [24]	POI embedding	✓	✓			✓		
2019 [11]	Knowledge graph embedding	✓	✓			✓		✓
2019 [26]	Shortest path algorithm	✓	✓		✓	✓		✓
2019 [52]	RNN with attention	✓	✓			✓		✓
2020 [21]	Gravity model	✓	✓			✓		✓
2020 [65]	RNN with attention	✓	✓			✓		✓
2020 [46]	Greedy algorithm	✓	✓			✓		✓
Current article	Self-supervised learning	✓	✓	✓	✓	✓	✓	✓

2.2 Self-Supervised Learning

While deep learning approaches have achieved outstanding results in many fields, current models heavily rely on a massive amount of labeled data [30]. Finding additional datasets or making better use of unlabeled data is always of interest to researchers [43]. Recently, many excellent SSL approaches have achieved comparable performance as the supervised models on a range of CV [7, 25, 27], audio [42], and NLP [31] tasks, where contrastive losses are used to distill extra knowledge from the data itself. The main idea is to formulate the task of finding similar and dissimilar parts of the data and maximize the mutual information between positive (similar) and negative (distinct) samples. For example, Reference [27] showed that maximizing the mutual information between an image and the local regions of the image improves the quality of representation learning. A study of sequential data with SSL, which achieved good performance in audio recognition, was presented in Reference [42]. Subsequently, Kong et al. [31] extended SSL to the field of NLP, obtaining negative samples from both training data and the randomly selected vocabulary tasks.

A few recent works focus on improving recommender systems using SSL. For example, SSL has been utilized to model the historical behavior of users and capture click/purchase correlations for sequential recommendation [39, 56, 57, 70], where the basic idea is to generate a multi-view of users history and maximize the mutual information. However, these works focus on recommending a single next item, leveraging rich historical user data for training the models. In contrast,

Table 2. Mathematical Notations Used in This Paper

Notation	Description
l	Identifier of the POI
l^{lat} and l^{lon}	geo-coordinates (i.e., latitude and longitude of l).
$l_{i,\tau} =< l^{\text{lat}}_{i,\tau}, l^{\text{lon}}_{i,\tau} >$	Geo-location $(l^{\text{lat}}, l^{\text{lon}})$ visited at time τ.
$\|L\|$ (*resp.* K)	Number of unique POIs in the dataset.
$\boldsymbol{h}_t$	The hidden state vector of the POI recommender.
$\mathcal{T}$	Trajectory $\mathcal{T} = \{l_i \| i \in [1, N]\}$ is a sequence of POIs recommended by the model.
N	Length of trip, i.e., the number of POIs in a trajectory.
o	Similarity score between hidden state $\boldsymbol{h}_t$ and l.
$\mathbf{u}(\eth)$, $\mathbf{u}(s)$, $\mathbf{u}(e)$	Representations of POI geographical distance, lower distance, and upper distance, respectively.
$\mathbf{v}(l)$, $\mathbf{u}(t)$	Embeddings of POI l and visiting time t.

tour recommendation requires planning a sequence of POIs that is more complex than the next item/POI recommendation tasks, in addition to the extreme sparsity of the data that can be used for training. In this spirit, we take the first attempt at learning human mobility and planning tours using SSL and contrastive trajectory learning.

3 PRELIMINARIES

In this section, we start with formally defining the tour recommendation problem and then provide the necessary background in mutual information and contrastive learning. Table 2 summarizes the frequently used notations.

Let $L = \{l_1, l_2, \ldots, l_K\}$ denotes the set of POIs that have been (potentially) visited by the users. Each particular visit is a pair (l_i, t_j), where $l_i \in L$ and t_j is the time of visiting that POI. We assume that each $l_i \in L$ is associated with a geographical location with a known coordinate, so, strictly speaking, a visit can be perceived as a triplet $(l^{\text{lat}}_i, l^{\text{lon}}_i, t_j)$. When there is no ambiguity, we will also use an alternate notation, omitting the explicit time parameter from the triplet and use it in the subscript to indicate a visit, as in $l_{i,\tau} =< l^{\text{lat}}_{i,\tau}, l^{\text{lon}}_{i,\tau} >$, indicating that the location $(l^{\text{lat}}_i, l^{\text{lon}}_i)$ was visited at time τ. Following previous work [5, 16, 17, 33, 45], we define the tour recommendation problem as:

Definition 3.1 (Tour Recommendation).
INPUT: A user-provided query consisting of the desired start point l_s and start time t_s, the length of the trip N (i.e., the number of POIs to visit), and the end point l_e at time t_e.
OUTPUT: The tour recommender system returns a tour route $\mathcal{T} =(l_1 = l_s, l_2, l_3, \ldots, l_N = l_e)$.

3.1 Background on Mutual Information Maximization

Mutual information (MI) is a Shannon entropy-based measurement of random variable dependencies [2], i.e., given two variables X and Y, the mutual information can be understood as how much knowing the X reduces the uncertainty in Y or vice versa:

$$I(X, Y) = H(X) - H(X|Y) = H(Y) - H(Y|X). \tag{1}$$

From the perspective of probability, mutual information is derived from a joint distribution and the product of two marginals on the random variables, which is defined as follows:

$$I(X, Y) = \sum_{y \in Y} \sum_{x \in X} p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right). \tag{2}$$

Consider a classification problem that aims to predict the label y by giving an input variable According to the Fano's inequality [54]:

$$p(y \neq \hat{y}) \geq \frac{H(Y \mid X) - 1}{\log N_y} = \frac{H(Y) - I(X, Y) - 1}{\log N_y}, \tag{3}$$

where y is the true label, $\hat{y}$ is the predicted label, and N_y is the number of samples. This implies that the lower bound of classification error is negatively related to the mutual information between the input and output variables, i.e., minimizing the classification error is equivalent to maximizing the mutual information.

3.2 Self-Supervised Learning by Contrasting Different Data Views

Contrastive SSL has recently received a great deal of attention in both academia and industry. The core idea behind SSL lies in pre-training the model on a large amount of data to realize self-supervised signals by measuring the distance between positive and negative samples without label supervision. This paradigm avoids the need for human-annotated labels while simultaneously improving the model's generalizability and robustness. Generally, it consists of three ingredients, i.e., the *anchor*, *positive*, and *negative* samples. The distance between the anchor x and a positive sample x^+ should be smaller than the distance between the anchor x and a negative sample x^- in the latent space of the learned representations. In equivalent terms:

$$f_\theta(x, x^+) \gg f_\theta(x, x^-), \tag{4}$$

where $f_\theta(\cdot, \cdot)$ denotes a similarity function (e.g., dot product or cosine similarity).

For one negative sample, the goal is to maximize the following expression:

$$\max \left[\frac{f_\theta(x, x^+)}{f_\theta(x, x^+) + f_\theta(x, x^-)} \right], \tag{5}$$

which is also known as the InfoNCE [23] loss:

$$\mathcal{L} = -\mathbb{E}_{(x, x^+)} \left[f_\theta(x, x^+) - \log \sum_{x_i \in N_{\text{neg}}} \exp f_\theta((x, x_i)) \right], \tag{6}$$

where N_{neg} denotes the set of negative samples. This contrastive approach has been widely used in a range of SSL-based language and visual recognition tasks [31, 42, 43].

Note that the InfoNCE is related to the cross entropy loss. If the variable $\tilde{Y}$ always includes all possible values of Y (i.e., $\tilde{Y} = Y$) and they are distributed uniformly, then maximizing InfoNCE is analogous to maximizing the standard cross entropy loss:

$$\mathbb{E}_{p(X, Y)} \left[f_\theta(x, y) - \log \sum_{\tilde{y} \in Y} \exp f_\theta(x, \tilde{y}) \right]. \tag{7}$$

This equation indicates that InfoNCE is related to maximizing $p_\theta(y|x)$, and it approximates the summation over Y's elements by utilizing the negative sampling. Based on the above equation, we can use specific X and Y to maximize the mutual information between different views of the

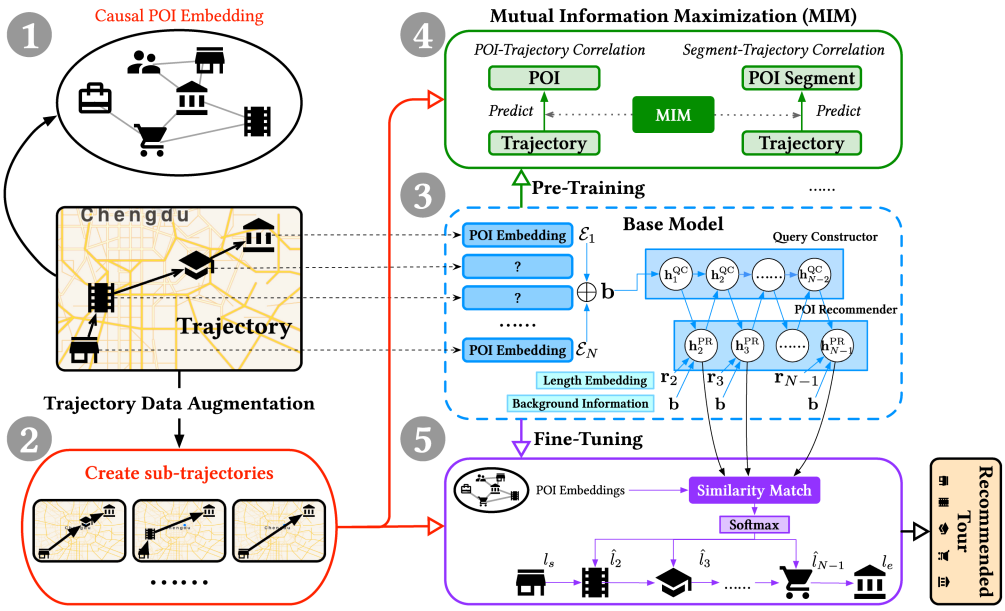


Fig. 1. An illustration of our proposed CTLTR framework, which is composed of the following five main modules: (1) It first encodes POIs into low-dimensional embeddings while also considering spatial and temporal contexts of POIs; (2) the designed trajectory data augmentation procedure greatly expands the pre-training samples by creating sub-trajectories; (3) a hierarchical Base model formed by two LSTM networks (*query constructor* and *POI recommender*); (4) two pretext tasks (POI-trajectory correlation and segment-trajectory correlation) based on mutual information maximization, enabling us to pre-train the CTLTR model without label information; and (5) a fine-tuned prediction layer for tour recommendation.

raw data, e.g., the POI-trajectory correlation and the segment-trajectory correlation modeled in CTLTR.

4 METHODOLOGY: CONTRASTIVE TRAJECTORY LEARNING

We now present the details of our proposed CTLTR, along with the trajectory data augmentation procedure and the self-supervised trajectory learning. CTLTR framework has the following two main procedures: (1) The Base model decomposes the trajectory recommendation task as a multi-round next POI planning problem and considers spatial and temporal contexts of POIs (cf. Section 4.1), and (2) the self-supervised trajectory learning combines pre-training and fine-tuning strategies with augmented trajectories, which models two kinds of pre-training strategies via mutual information maximization, i.e., POI-trajectory and segment-trajectory correlations (cf. Section 4.2). The overall model architecture is shown in Figure 1.

4.1 Base Model: Multi-Round Next POI Planning

The Base model serves as a basic supervised framework to encode the trajectories into latent representations containing semantic relationships and sequential visiting patterns between POIs. Then it decodes the latent representation for the personalized tour recommendation in an end-to-end manner by utilizing deterministic deep recurrent neural networks. Specifically, a query constructor has been used to generate time- and length-dependent queries, which embeds the background

knowledge of the current POI sequence at query time. A POI recommender is subsequently used to output the desired tour recommendations recursively.

4.1.1 Causal POI Representation. In this work, we follow previous models [24, 65] and use the Continuous Bag-of-Words [41], which was initially designed for word embedding, to learn the POI representation. It preserves the latent relationships among POI locations in a low-dimensional space. In particular, a sliding window of size 3 is used to intercept each trajectory and to predict the middle POI with the context POIs (i.e., the previous and next ones). The obtained sequential POI embedding vector denoted as $\mathbf{v}(l)$, is consistent with the downstream tour recommendation and can be regarded as recommending a single middle POI given the start and endpoints. Essentially, this word2vec-style POI embedding can be considered as the simplest self-supervised learning method [36], since we only explore the supervision signals from the trajectory data itself. Note that we just use the POI embedding as an initialization, which would be fine-tuned with self-supervised trajectory learning introduced later.

4.1.2 Spatial and Temporal Contexts of POIs. In addition to the POI embedding, we further encode the spatial and temporal contexts of POIs. Following Reference [17], we encode the spatial-temporal context of each location in a trajectory by incorporating the geographical and temporal constraints imposed by the start point and end points.

That is, the current-time geographical distance $\mathbf{u}(l_{i,\tau})$ of a particular POI visiting $l_{i,\tau}$ (i.e., a tourist visits l_i at time τ) is calculated by the following:

$$\mathbf{u}(l_{i,\tau}) = \frac{1}{2}\left(\frac{\mathrm{d}(l_{i,\tau}, l_{s,t_s})}{\mathrm{d}_{\max}}\mathbf{u}(s) + \frac{\mathrm{d}(l_{i,\tau}, l_{e,t_e})}{\mathrm{d}_{\max}}\mathbf{u}(e)\right), \tag{8}$$

where $\mathrm{d}(\cdot,\cdot)$ denotes the Euclidean distance between two locations and $\mathrm{d}_{\max}$ is the maximum distance between any pair of locations in the training data (i.e., all pairs from L). $\mathbf{u}(s)$ denotes the distance to the closest POI from the starting point l_s, and $\mathbf{u}(e)$ is the distance to the furthest POI from the end point l_e. The rationale behind Equation (8) is to account for the relative distance constraints imposed by the start and end POIs.

Subsequently, we map the temporal and distance information into a low-dimensional space and concatenate them with the causal POI embedding as a unified location representation $\mathcal{E}_{\tau_i} = \mathbf{v}(l_i) \oplus \mathbf{u}(l_{i,\tau}) \oplus \mathbf{u}(\tau)$, where $\oplus$ is the concatenation operation and $\mathbf{v}(l_i)$ and $\mathbf{u}(\tau)$ are the embeddings of the location l_i and of the visiting time τ. We note that other contexts and constraints, e.g., duration time and queuing time, can be encoded in a similar way.

4.1.3 Hierarchical Response Generator. We decompose the tour recommendation task into a multi-round next station planning problem. Different from traditional sequence-to-sequence models that construct the user query only from the past information, we further consider the number of remaining POIs that still need to be recommended, which can boost the recommendation performance through narrowing the searching space. Based on the current query and background information (the start and end POIs), our model predicts the next preferred POI by a hierarchical architecture to form the user query. This process can be interpreted as: A tourist has visited a sequence of POIs, and then s/he decides the next POI to visit. According to the current prediction along with previous predicted POIs, our model adjusts the query and carries out the next round of POI planning.

The planning will iterate until the complete trajectory is generated, i.e., when the length of the generated trajectory $|\mathcal{T}|$ meets the number of desired attractions N. As shown in Figure 1, the hierarchical response generator consists of two parts: a *query constructor* and a *POI recommender*. The former aims to model the existing trajectories and generate the corresponding hidden state

representation by utilizing recurrent neural networks. In CTLTR, we select the **long short-term memory (LSTM)** [28] as the basic recurrent unit to model the temporal dependencies among POI trajectories. Similarly, a POI recommender based on LSTM is used to generate the next recommended POI given the background knowledge and all past POIs as input.

The POI generating process is executed in a recursive manner. Specifically, for a running round at time $t \in [2, N-1]$, we concatenate the historical information $\mathbf{h}_{t-1}^{\mathrm{QC}}$ and the length embedding $\mathbf{r}_t$ (obtained by random initialization) to form the current query:

$$\mathbf{b} = \mathcal{E}_{\tau_s} \oplus \mathcal{E}_{\tau_e}, \tag{9}$$

$$\mathbf{h}_t^{\mathrm{QC}} = \mathrm{LSTM}\left(\mathbf{h}_t^{\mathrm{PR}}, \mathbf{h}_{t-1}^{\mathrm{QC}}\right), \tag{10}$$

$$\mathbf{h}_t^{\mathrm{PR}} = G_t\left(\mathbf{h}_{t-1}^{\mathrm{QC}} \oplus \mathbf{r}_t, \mathbf{b}\right), \tag{11}$$

where $\mathcal{E}_{\tau_s}$ and $\mathcal{E}_{\tau_e}$ are the unified location embeddings; $\mathbf{h}_t^{\mathrm{QC}}$ and $\mathbf{h}_t^{\mathrm{PR}}$ are the LSTM hidden state vectors of the query constructor and the POI recommender, respectively; and $G_t(\cdot,\cdot)$ denotes the gating unit in LSTM. The hierarchical response generator recommends the next POI by taking three important factors into account, i.e., (1) the background knowledge $\mathbf{b}$, (2) the historical travel information $\mathbf{h}_{t-1}^{\mathrm{QC}}$, and (3) the current length embedding $\mathbf{r}_t$. Next, the POI generator combines these information together through the gating mechanism to generate $\mathbf{h}_t^{\mathrm{PR}}$ for the next round of recommendation.

4.1.4 POI Prediction Layer. Once the hierarchical response generator produced enough tour hidden state vectors $\{\mathbf{h}_2^{\mathrm{PR}}, \mathbf{h}_3^{\mathrm{PR}}, \ldots, \mathbf{h}_t^{\mathrm{PR}}, \ldots, \mathbf{h}_{N-1}^{\mathrm{PR}}\}$, we compute the similarity score o_i between the hidden state vector $\mathbf{h}_t^{\mathrm{PR}}$ and the POI embedding $\mathbf{v}(l_i)$. The probability of observing l_i given $\mathbf{h}_t^{\mathrm{PR}}$ is derived by applying the softmax function on o_i:

$$o_i = \mathrm{similarity}\left(\mathbf{h}_t^{\mathrm{PR}}, \mathbf{v}(l_i)\right) = \mathbf{h}_t^{\mathrm{PR}} \odot \mathbf{v}(l_i), \tag{12}$$

$$P_i = \mathrm{softmax}(o_i) = \frac{\exp(o_i)}{\sum_{j=1}^{|L|} \exp(o_j)}, \tag{13}$$

where $\odot$ is the dot product, $|L|$ is the number of unique POIs in the dataset. POI with the highest probability from the POI distribution is selected as the next recommended POI. In this article, we use the Gumbel-Max technique [29], which enables a simple but effective way to sample probabilities from a categorical distribution.

4.2 Pre-training and Fine-Tuning CTLTR with Mutual Information Maximization

Now we describe the details of contrastive learning in CTLTR, which consists of two new distinct designs over the Base model. First, the previous supervised deep learning-based models are susceptible to the overfitting problem due to the shortage of available trajectory data. However, currently prevailing unsupervised or semi-supervised models cannot be directly applied here, as there is no extra unlabeled trajectory data. Therefore, designing a new trajectory data augmentation procedure becomes an urgent need for self-supervised trajectory pre-training. In CTLTR, we propose a new trajectory data augmentation approach by removing POIs in trajectories, which greatly increases the pre-training trajectory samples. For example, in Flickr@Edinburgh dataset, the number of trajectories increased from 2,681 to 156,374 after augmentation (cf. Section 5.1). Second, we proposed two novel pre-training strategies explicitly devised for contrastive trajectory data learning. In particular, two correlations of trajectories are considered in CTLTR, i.e., the POI-trajectory correlation and segment-trajectory correlation. We used self-supervised signals to minimize the pre-training losses via **mutual information maximization (MIM)**. Once the CTLTR model is

pre-trained, it can be used as fine-tuning on the tour recommendation problem. The *pre-training* and *fine-tuning* paradigm of CTLTR significantly improves the recommendation accuracy on all four datasets, which demonstrates the superiority of CTLTR to other supervised baselines in alleviating the overfitting and data sparsity problems (cf. Section 5.5).

4.2.1 Trajectory Data Augmentation. Various data augmentation techniques have been successfully applied in many unsupervised or semi-supervised learning tasks to improve model generalizability and robustness. As shown in Reference [7], a composition of data augmentation procedures plays a critical role in defining effective vision contrastive pre-training tasks. However, unlike popular image augmentation operations such as crop, rotate, and filtering, how to augment trajectory data, and to what extent trajectory data should be augmented are yet underexplored. In CTLTR, we adopt a simple but effective trajectory data augmentation strategy. On the one hand, our strategy greatly expanded the pre-training samples, and, on the other hand, we manually created various sub-trajectory instances based on real user trajectories. Specifically, for one complete trajectory whose length $N > 3$, we remove one or more POIs (at most $N-3$ POIs) in this trajectory except the start and end POIs. Each removal creates a new trajectory instance. The number of new trajectory instances created from an original trajectory of length N is as follows:

$$\sum_{i=1}^{N-3} C_N^i = \sum_{i=0}^{N-2} C_{N-2}^i - \left(C_{N-2}^0 + C_{N-2}^{N-2}\right) = 2^{N-2} - 2. \tag{14}$$

The idea behind our augmentation strategy is that for a specific complete tour trajectory, some tourists may not have enough time to visit all the POIs, or during the tour, the travel plans have been interrupted by unexpected events, e.g., bad weather conditions, certain POIs are closed, or the user has spent too much time at some points. As a consequence, they have to visit only a sub-trajectory due to time and money budgets. In Section 5, we show that this simple procedure—introducing the trajectory data augmentation into CTLTR—significantly improves the recommendation accuracy.

4.3 Trajectory Pre-Training

Based on the trajectory data augmentation mentioned above and the hierarchical response generator from the Base model, now we are ready to pre-train the CTLTR model in a *task-agnostic* manner. Specifically, we proposed two pre-training pretext tasks, i.e., mutual information maximization on both POI-trajectory (PT) correlation and the segment-trajectory (ST) correlation. The two relationships can be viewed as self-supervised signals with mutual information maximization and used to enhance the learned representations of trajectory data. We pre-train the CTLTR model by minimizing a combined loss function from the two correlations, which will be presented in the next two subsections. The pseudo code of CTLTR pre-training is shown in Algorithm 1.

4.3.1 Modeling POI-trajectory Correlation. Since a trajectory is composed of a sequence of POIs, an intuitive way is to maximize the mutual information between a single POI l_{j,t_j} and the whole trajectory $\mathcal{T}$ that contains l_{j,t_j}. The pretext task of POI-trajectory correlation is similar to that of a *Cloze problem*. For instance, giving a single POI, the task requires predicting the surrounding POIs. It is analogous to the problem of recommending a tour comprising a given scenic spot. However, giving a sequence of POIs that are missing one specific POI in the middle, the task needs to predict the missing POI by given the surroundings. That is, we maximize the mutual information between the POI and the trajectory, and thereby we could predict the other given one of them.

At each pre-training step, given a POI l_{j,t_j} and the trajectory $\mathcal{T}$, we mask the POI l_{j,t_j}, denoted as `[mask]`, in the trajectory $\mathcal{T}$. We predict the masked POI l_{j,t_j} using the surrounding context

$\widetilde{\mathcal{T}} = \{l_{1,t_1}, \ldots, \texttt{[mask]}, \ldots, l_{N,t_N}\}$. The pre-training network optimize the following loss:

$$\mathcal{L}_{\text{PT}}(\widetilde{\mathcal{T}}, l_{j,t_j}) = -\log \frac{\exp\left(f\left(\widetilde{\mathcal{T}}, l_j\right)\right)}{\sum_{\tilde{l}_j \in L \setminus l_j} \exp\left(f\left(\widetilde{\mathcal{T}}, \tilde{l}_j\right)\right)}, \tag{15}$$

where l_j denotes the masked POI, which can be considered as the positive sample and $\tilde{l}_j$ denotes a negative POI sampled from the POI set L. That is, we consider $\widetilde{\mathcal{T}}$ as the anchor and use a similarity function $f(\cdot,\cdot)$ to measure the mutual information as

$$f\left(\widetilde{\mathcal{T}}, l_j\right) = \sigma\left(\mathbf{h}_{\widetilde{\mathcal{T}}}^{\top} \cdot \mathbf{W}_{\text{PT}} \cdot \mathcal{E}_{\tau_j}\right), \tag{16}$$

where $\mathbf{W}_{\text{PT}}$ is a learnable parametric matrix and $\mathbf{h}_{\widetilde{\mathcal{T}}}$ is the jth hidden state vector of $\widetilde{\mathcal{T}}$ in the hierarchical response generator, and $\sigma(\cdot)$ is the sigmoid function.

4.3.2 Modeling Segment-trajectory Correlation. In addition, to maximize the mutual information between POI and trajectory, we extend the POI-trajectory correlation to segment-trajectory correlation, which models a sequence of POIs (segment) with its surrounding context, due to the following two reasons: (1) it might be loosely correlated between a single POI and a complete trajectory, and (2) some POIs might be tightly correlated that most tourists visit them simultaneously. Therefore, we propose to model the segment-trajectory correlation in a similar way, i.e., define the pretext task as a subsequence Cloze problem.

Consider a sequence of POIs $\{l_{j,t_j}, \ldots, l_{j+n,t_{j+n}}\}$ with length $n+1 \in [1, N-2]$. We mask the subsequence `[mask1,mask2,...]` in the original trajectory $\mathcal{T}$. Then, we predict the masked segment based on the surrounding context $\widetilde{\mathcal{T}}_s = \{l_{1,t_1}, \ldots, \texttt{[mask1,mask2,...]}, \ldots, l_{N,t_N}\}$. The model is also optimized by a similarity loss function based on mutual information maximization:

$$\mathcal{L}_{\text{ST}}\left(\widetilde{\mathcal{T}}_s, S_{j,n}\right) = -\log \frac{\exp\left(f\left(\widetilde{\mathcal{T}}_s, S_{j,n}\right)\right)}{\sum_{\tilde{l}_{j,t_j}}^{\tilde{l}_{j+n,t_{j+n}}} \exp\left(f\left(\widetilde{\mathcal{T}}_s, \widetilde{S}_{j,n}\right)\right)}, \tag{17}$$

where $S_{j,n}$ denotes the masked POI segment $\{l_{j,t_j}, \ldots, l_{j+n,t_{j+n}}\}$ (i.e., the positive sample), and $\widetilde{S}_{j,n}$ denotes the trajectory segment sampled from other trajectories (i.e., negative samples). Similarly to Equation (16), the mutual information between the context and the trajectory segment can be computed as

$$f\left(\widetilde{\mathcal{T}}_s, S_{j,n}\right) = \sigma\left(\mathbf{h}_{\widetilde{\mathcal{T}}_s}^{\top} \cdot \mathbf{W}_{\text{ST}} \cdot \mathbf{h}_{S_{j,n}}\right), \tag{18}$$

where $\mathbf{h}_{\widetilde{\mathcal{T}}_s}$ and $\mathbf{h}_{S_{j,n}}$ are the last hidden states of the surrounding context $\widetilde{\mathcal{T}}_s$ and the trajectory segment $S_{j,n}$, respectively.

4.3.3 Fine-tuning on Tour Recommendation Problem. Once the model has been pre-trained, it can be used for fine-tuning on the tour recommendation problem. We fine-tune the CTLTR model on the original dataset while pre-training it on both the augmented and original datasets. The pre-trained model parameters are used as the initialization of the fine-tuned model. We use the cross-entropy loss function to optimize the model. Specifically, for a certain trajectory $\mathcal{T}$, the loss is calculated by the following:

$$\mathcal{L}_{\text{fine\_tune}}(\mathcal{T}) = \frac{1}{N-2} \sum_{i=2}^{N-1} -l_i * \log\left(\hat{l}_i\right), \tag{19}$$

where N is the length of the trajectory, l_i is the ith ground truth POI, and $\hat{l}_i$ is the predicted POI.

ALGORITHM 1: CTLTR Pre-training

Input: The pre-processed data, the iteration counter k for early stop
Perform data augmentation operation (cf. Section 4.2.1)
/* Pre-training */
initialize model M
while $\mathcal{L}_{PT}$ *not continuously decline for k epochs* **do**
    Mask POI of the trajectories (cf. Section 4.3.1) in current batch
    Evaluate $\mathcal{L}_{\mathrm{PT}}$ according to Equation (15)
    Update the parameters by gradient descent
end
while $\mathcal{L}_{ST}$ *not continuously decline for k epochs* **do**
    Mask POI segment of the trajectories (cf. Section 4.3.2) in current batch
    Evaluate $\mathcal{L}_{\mathrm{ST}}$ according to Equation (17)
    Update the parameters by gradient descent
end
Output: Trained Model M

5 EXPERIMENTS

In this section, we report the results from the extensive experimental evaluations to test the performance of our CTLTR model. We compare it to the state-of-the-art tour recommendation methods. We also examine the interplay of the components of our model and various aspects of the CTLTR algorithm (efficiency, parameter tuning, and interpretability).

5.1 Datasets

We evaluated all methods based on the YFCC100M (Yahoo! Flickr Creative Commons 100M) dataset[1] [47] that consists of a publicly available curated dataset of almost 100 million photos in the world. Following References [17, 24, 35], we used the data extracted from four different cities: Toronto, Osaka, Glasgow, and Edinburgh. To ensure the robustness and versatility, only the photos with the highest accuracy of the geographical location were used in the experiment. Table 3 summarizes the statistics of the datasets. Specifically, each POI check-in in the track contains the tourist ID, timestamp, POI longitude and latitude, as well as the category information of the POIs. Our algorithm does not use user profile information to provide a more general trajectory recommendation. In each dataset, we only retain the trajectories whose check-ins are more than 3. In addition, we use leave-one-out cross-validation to evaluate all methods, exactly following the related References [5, 17, 35].

5.2 Baselines

We compare our CTLTR with following 12 models on trip recommendation:

- **Popularity [15]:** This is a relatively straightforward method that recommends tours based on the ranking of POI popularity.
- **PersTour and PersTour-L [35]:** PersTour is an orienteering-based method that recommends a sequence of POIs with a time budget. PersTour-L is a variant replacing the time budget with the constraint of trajectory length.

[1] https://bit.ly/yfcc100md.

Table 3. Statistics of Four Flickr Dataset

	Flickr			
City	**Edinburgh**	**Glasgow**	**Osaka**	**Toronto**
# photo	82,060	29,019	392,420	157,505
# visit	33,944	11,434	7,747	39,419
# trajectory	2,681	395	165	1,243
# trajectory after augmentation	156,374	1,544	350	37,506
# POI	29	29	29	30
# user	1,454	601	450	1,395
# POI/trip	6.75	5.13	6.95	6.50

- **POIRank [5]:** This recommends tours according to the POI ranks. It first ranks POIs with various features (e.g., popularity, average duration, etc.) using rankSVM and then connects them according to ranking scores to form the recommended tour.
- **Markov and Markov-Rank [5]:** Markov method models the POI-to-POI transition probabilities and recommends a tour by maximizing the transition likelihood. Markov-Rank is a method combining the advantages of Markov and POIRank, which leverages both POI ranking and Markov transition for tour recommendation. The ranking of POIs is learned by rankSVM with linear kernel and L2 loss. Trajectories recommended by Markov and Markov-Rank are trained using the maximum likelihood approach.
- **Path and Path-Rank [5]:** They eliminate the sub-tour problem in Markov and Markov-Rank by finding the best path using **Integer Linear Program (ILP)**, where the sub-tour constraints are adapted from the Traveling Salesman Problem. The only difference between the two methods is that Path-Rank considers the POI ranks in trajectory modeling.
- **PersQueue [33]:** This is a reinforcement learning based approach, which aims to maximize POI popularity and user interest preferences while minimizing the queuing time at attractions. It uses Monte Carlo tree to design the reward of a trajectory and recommend an optimal tour.
- **CATHI [69]:** This exploits the context of trajectories in an auto-encoder manner, composed by two encoders and two decoders and tuned with a variational attention mechanism. Since CATHI is not specified for tour recommendation, we adapt it to the problem by generating the sequential POIs after pre-training the trajectory data.
- **C-ILP [24]:** This uses word2vec to jointly learn the embeddings of users and POIs. It solves the trip recommendation problem by adapting the approximate large neighborhood search and ILP techniques. Note that the the corresponding code(s) for C-ILP were not publicly available, so we directly used the results published in the article.
- **DeepTrip [17]:** This leverages RNN-based autoencoder as its basic framework, and adopts an auxiliary neural network to learn the sequential POIs distribution in an adversarial learning manner.

5.3 Metrics

We evaluate the model performance using two commonly used metrics that are frequently used in tour recommendation studies [5, 17], i.e., F_1 and pairs-F_1 scores.

- **F_1 score** is defined as

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{20}$$

Table 4. Parameter Settings in Our Experiments

Parameter	Value	Parameter	Value
batch size	32	POI embedding size	64
initial learning rate	0.1	hidden size	64
dropout rate	0.8	time embedding dim	16
Gumbel temperature	0.4	length embedding dim	16
negative number	$L-1$	background dim	64

which is the harmonic mean of Precision and Recall of recommended POIs in a trip. F_1 score is suitable for assessing whether the preferred POIs are correctly recommended, which has been widely used in previous studies [5, 17, 35].

- **pairs-F_1 score:** Although F_1 score can evaluate the correctness of the POI-level recommendation, it ignores the visiting order of users. To address this issue, a specific metric called pairs-F_1 has been proposed in Reference [5]. It considers both POI preference and visiting order of users by measuring the F_1 score of every pair of POIs, i.e., whether they are adjacent or not in a trajectory, and is defined as

$$\text{pairs-F}_1 = \frac{2 \times \text{pairs-P} \times \text{pairs-R}}{\text{pairs-P} + \text{pairs-R}}, \tag{21}$$

where pairs-P and pairs-R denote the Precision and Recall of ordered POI pairs, respectively. The values of both pairs-F_1 and F_1 are between 0 and 1. The higher the value, the better the recommended tour, e.g., a value of 1 indicates that both POIs and their order in the planned trip are exactly the same as the ground truth.

5.4 Experimental Settings

We implement our model based on TensorFlow 1.14 with Python 3.7. The trajectories with fewer than three POIs are filtered out. For each dataset, we use leave-one-out cross-validation in both pre-training and training stages. The pre-training is early stopped when the loss does not decline after 10 epochs. We use the Adam optimizer for mini-batch gradient descent with the size of 32 in each mini-batch, and set the initial learning rate to 0.1 and use exponential decay to reduce the learning rate gradually. For two LSTM used in the Base model, we set the hidden size to 64. Other hyper-parameter settings are presented in Table 4.

5.5 Performance Comparison

Tables 5 and 6 report the F_1 and pairs-F_1 values of all the methods on the four datasets. According to the results, we have the following observations.

First, the deep learning-based approaches, including CTLTR, achieve better performance across all datasets compared to orienteering problem-based and traditional machine learning-based approaches. This demonstrates the advantages of neural networks in learning human visiting preferences and planning tours for users. Among the baselines, DeepTrip usually performs better due to its specifically designed adversarial trajectory learning model for discriminating user preference. In addition, CATHI performs well on tour recommendations, although it was originally proposed to learn the context of trajectories. However, C-ILP, as an orienteering problem-based approach, combines the effectiveness of check-in embedding and integer linear programming, but it is too simple to capture the intrinsic POI transition patterns, while also requiring significantly more computational cost to search the optimal solutions.

Table 5. F_1 Score Comparisons among Different Baselines in Four Cities

	Flickr Dataset			
Model	***Edinburgh***	***Glasgow***	***Osaka***	***Toronto***
Popularity	0.701 ± 0.160	0.745 ± 0.166	0.663 ± 0.125	0.678 ± 0.121
PersTour	0.656 ± 0.223	0.801 ± 0.213	0.686 ± 0.231	0.720 ± 0.215
PersTour-L	0.651 ± 0.143	0.660 ± 0.102	0.686 ± 0.137	0.643 ± 0.113
POIRank	0.700 ± 0.155	0.768 ± 0.171	0.745 ± 0.173	0.754 ± 0.170
Markov	0.645 ± 0.169	0.725 ± 0.167	0.697 ± 0.150	0.669 ± 0.151
Markov-Rank	0.659 ± 0.174	0.754 ± 0.173	0.715 ± 0.164	0.723 ± 0.185
Path	0.678 ± 0.149	0.732 ± 0.168	0.706 ± 0.150	0.688 ± 0.138
Path-Rank	0.697 ± 0.152	0.762 ± 0.167	0.732 ± 0.162	0.751 ± 0.170
PersQueue	0.470 ± 0.196	0.586 ± 0.231	0.507 ± 0.186	0.536 ± 0.187
CACHI	0.772 ± 0.123	0.815 ± 0.127	0.758 ± 0.091	0.807 ± 0.106
C-ILP	0.769 ± 0.000	0.853 ± 0.000	0.763 ± 0.000	0.818 ± 0.000
DeepTrip	0.833 ± 0.142	0.831 ± 0.172	0.834 ± 0.166	0.811 ± 0.163
CTLTR	**0.853 ± 0.151**	**0.874 ± 0.147**	**0.889 ± 0.142**	**0.874 ± 0.141**

Simple methods, such as Popularity and POIRank, perform well on the four datasets, even compared with specifically designed machine learning approaches. This surprising result reflects that tourists are inclined to choose popular attractions when traveling to a new city. PersTour usually outperforms PersTour-L, which suggests the time budget is a more relevant constraint than trajectory length when modeling tour recommendation as an orienteering problem. Meanwhile, the two methods jointly model the POI popularity and user interest preference, which sometimes (e.g., on Edinburgh) result in lower performance than Popularity and POIRank. We conjecture that there may be a conflict between POI popularity and user interest that requires reconciliation, which has not been well addressed in previous methods.

Third, several Markov-based approaches perform very closely on four datasets. After scrutinizing the subtle difference between these models, we find that taking the POI rank into account would improve the recommendation performance, further proving the role of POI popularity in enriching the simple Markov transition model, even though its effect is insignificant. As models that are specifically tailored for eliminating sub-tours in the planned trajectory, Path and Path-Rank slightly outperform the corresponding base models (i.e., Makov and Markov-Rank), since the existence of sub-tours would deteriorate the recommendation performance. Another unexpected observation is that PersQueue does not show competitive results in both metrics. As a customized model for reducing the queuing time in attractions, this method is more suitable for modeling and recommending tours with time constraints rather than the travel length constraint.

Moreover, all methods, including ours, did not perform well on the pairs-F_1 metric, implying that the visiting orders are more difficult to be captured compared to POI correctness in the recommended tour. This phenomenon also indicates that tour recommendation is a non-trivial task as no methods can recommend the tours fully matching the ground truth. Notably, deep learning models usually perform significantly better than traditional methods in capturing the visiting orders, primarily due to the widely employed pre-training strategy. It enables us to well capture the long- and short-transition patterns and overcome the memory-less issue of Markov-based approaches.

Finally, we note that our proposed CTLTR model consistently outperforms all baselines on the data in four cities. More specifically, on average, CTLTR achieves 4.5% and 6.8% improvements

Table 6. Pairs-F_1 Score Comparisons among Different Baselines in Four Cities

	Flickr Dataset			
Model	***Edinburgh***	***Glasgow***	***Osaka***	***Toronto***
Popularity	0.436 ± 0.259	0.507 ± 0.298	0.365 ± 0.190	0.384 ± 0.201
PersTour	0.417 ± 0.343	0.643 ± 0.366	0.468 ± 0.376	0.504 ± 0.354
PersTour-L	0.359 ± 0.207	0.352 ± 0.162	0.406 ± 0.238	0.333 ± 0.163
POIRank	0.432 ± 0.251	0.548 ± 0.311	0.511 ± 0.309	0.518 ± 0.296
Markov	0.417 ± 0.248	0.495 ± 0.296	0.445 ± 0.266	0.407 ± 0.241
Markov-Rank	0.444 ± 0.263	0.545 ± 0.306	0.486 ± 0.288	0.512 ± 0.303
Path	0.400 ± 0.235	0.485 ± 0.293	0.442 ± 0.260	0.405 ± 0.231
Path-Rank	0.428 ± 0.245	0.533 ± 0.303	0.489 ± 0.287	0.514 ± 0.297
PersQueue	0.320 ± 0.243	0.384 ± 0.224	0.363 ± 0.283	0.336 ± 0.257
CACHI	0.515 ± 0.180	0.523 ± 0.236	0.516 ± 0.199	0.559 ± 0.205
DeepTrip	0.660 ± 0.246	0.782 ± 0.257	0.755 ± 0.268	0.751 ± 0.242
CTLTR	**0.729 ± 0.277**	**0.807 ± 0.267**	**0.834 ± 0.236**	**0.850 ± 0.210**

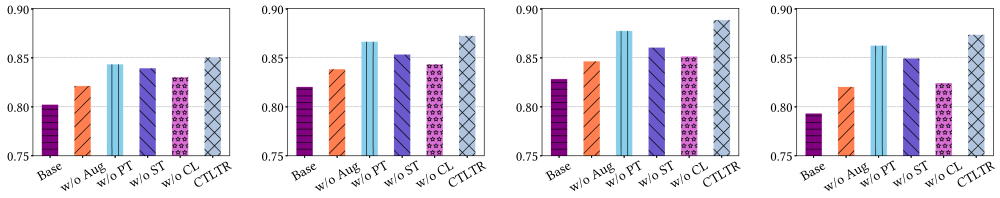
over the best baseline in terms of F_1 and pairs-F_1, respectively. The main superiority of our model lies in its trajectory data augmentation and contrastive trajectory learning, which enables us to better capture indistinct users' visiting patterns. Compared to the previous deep learning-based tour recommendation models, CTLTR extracts extra supervision signals from the user trajectories themselves and is particularly useful for sparse trajectory data. These results justify our motivation; i.e., the self-supervised learning paradigm can help improve sequential decision-making performance with careful multi-view trajectory designs, even with sparse historical data. From this perspective, we empirically observe the applicability of contrastive learning in sequence recommendation/planning problems beyond single item recommendation that have recently been studied extensively [39, 56, 70].

5.6 Ablation Study

We now investigate the effect of individual components in CTLTR. To this end, we conduct an ablation study through implementing the following variants:

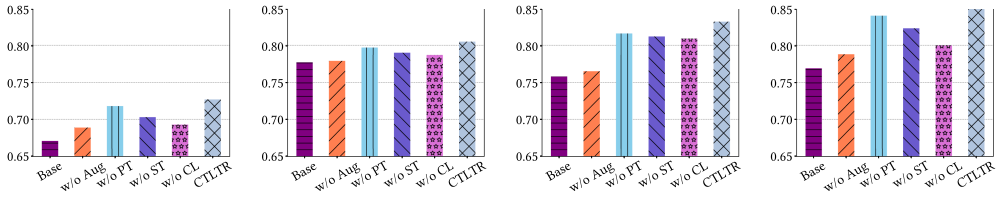
- **CTLTR-Base:** This is a basic model that only uses the hierarchical response generator described in Section. 4.1.3 for tour recommendation, i.e., there are no data augmentation operations and pre-training with MIM.
- **CTLTR w/o Aug:** This does not incorporate the augmented data for training, i.e., contrastive trajectory pre-training is performed on the original data.
- **CTLTR w/o PT:** This does not model POI-trajectory mutual information during pre-training.
- **CTLTR w/o ST:** This is another variant without modeling segment-trajectory correlations.
- **CTLTR w/o CL:** This is a variant omitting both POI-level (PT) and segment-level (ST) contrastive learning. Note that this method keeps the augmented trajectory data during training.

As presented in Figures 2 and 3, we have the following insights that enable us to understand how contrastive learning facilitates the learning of user visiting patterns. Overall, several main components of CTLTR contribute differently to the tour recommendation results. For example, the Base model's performance is competitive compared with the previous baselines, although it achieves the lowest scores among the variants. Recall that the Base model employs a simple hierarchical response generator to recursively reduce the searching space. This result suggests that



(a) Edinburgh (b) Glasgow (c) Osaka (d) Toronto

Fig. 2. F_1 results of ablation study.



(a) Edinburgh (b) Glasgow (c) Osaka (d) Toronto

Fig. 3. Pairs-F_1 results of ablation study.

the hierarchical model borrowed from language generation can boost tour planning performance even without data augmentation and contrastive trajectory learning.

Moreover, we can see that data augmentation plays a crucial role in the proposed model, as it significantly enriches the training data that are usually sparse and extremely imbalanced. We have employed a straightforward trajectory augmentation method that intercepts the subsequences of original trajectories, which is essentially a subtrajectory resampling strategy for balancing the trajectory distributions. Besides, it provides a multi-view of original visiting histories and enhances the model robustness in learning user interest preferences. However, we also note that the data augmentation substantially changes the trajectory distribution that may result in inductive bias. This also raises an open problem, i.e., how to de-bias the model while preserving data augmentation benefits, which is left for our future work.

As the main contribution of this work, contrastive trajectory learning indeed benefits user preference learning. The two pre-training strategies, i.e., PT and ST, help learn better POI and trajectory representations for downstream tasks, e.g., tour planning in this article. Among the two correlation modeling methods, ST (segment-trajectory) mutual information is relatively more important, which is not unexpected, since it captures the coherence of visiting orders compared to POI-level contrastive learning (i.e., POI-trajectory correlation). When looking at the difference between CTLTR w/o Aug and CTLTR w/o CL, we can see that the mutual information learning would manifest the advantages of contrastive learning on the augmented data rather than on the original trajectories (we note that the contrastive learning is performed on the original trajectories in CTLTR w/o Aug). This result is in line with the observations in computer vision and graph-structured data, i.e., self-supervised data learning requires manually designed and task-specified data augmentation [48, 55, 58].

5.7 Model Analysis

5.7.1 Influence of Negative Samples. To further determine the effect of contrastive learning on self-supervised learning, we performed a series of experiments to find a better sampling strategy. During the sample processing of contrastive learning, we take different quantities of negative

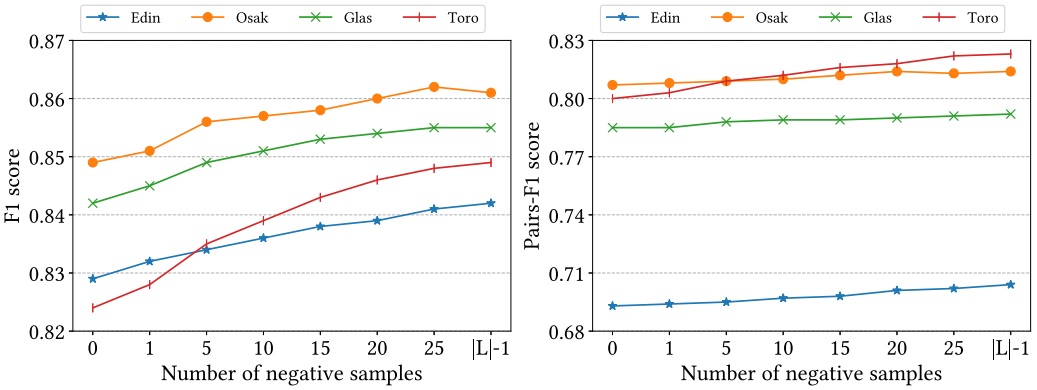


Fig. 4. Effects of negative samples. Note that when # of negative samples equals to zero, the model degenerates to CTLTR w/o CL. When # of negative samples equals to $|L| - 1$, the model is CTLTR w/o ST.

Table 7. The Effect Analysis of Gumbel Noise

		Flickr Dataset			
Metric	**System**	**Edinburgh**	**Glasgow**	**Osaka**	**Toronto**
F_1 score	Normal greedy	0.845 ± 0.169	0.867 ± 0.168	0.882 ± 0.148	0.865 ± 0.152
	+Noise	0.853 ± 0.151	0.874 ± 0.147	0.889 ± 0.142	0.874 ± 0.141
Pairs-F_1	Normal greedy	0.714 ± 0.281	0.795 ± 0.283	0.818 ± 0.256	0.831 ± 0.218
	+Noise	0.729 ± 0.277	0.807 ± 0.267	0.834 ± 0.236	0.850 ± 0.210

samples to figure out their influence on four datasets based on CTLTR w/o ST. Specifically, in the pre-training stage (cf. Section 4.3.1), after masking the POI, we change the negative sample space to affect the number of negative samples. Since we use incorrect POIs as negative samples, we change their number from 0 to $|L| - 1$ (recall that $|L|$ is the number of POIs). Both F_1 score and Pairs-F_1 score indicate that the more negative samples, the better the performance of contrastive learning. The results (cf. Figure 4) demonstrated that increasing the number of negative samples within a certain limit could effectively improve the performance of downstream tasks [7, 25]. We conjecture that a quantity of negative samples could help the model recognize the difference between negative samples and anchor and reduce their MI, thus improving the prediction accuracy (cf. Equation (3)).

5.7.2 The Effect of Gumbel Noise. Recall that CTLTR leverages Gumbel noise [60] to relieve the overfitting problem when selecting POIs caused by data sparsity, which allows us to sample from discrete distribution and estimate the maximum *a posteriori* more easily. To investigate the influence of the Gumbel noise, we conduct a comparison experiment to study the effect of Gumbel noise. To ensure the results' reliability, we ran 10 times experiments and reported the averaged scores, as illustrated in Table 7. Compared with the method that directly selects the next POI with the highest probability, adding Gumbel noise yields an increase of 0.7% and 1.6% in terms of F_1 score Pairs-F_1 score, respectively, on the Osaka dataset.

In addition, the method of selecting the POI with maximum probability after the Gumbel noise is more stable, e.g., its standard errors on F_1 and Pairs-F_1 decrease 0.6% and 2.0% in Osaka dataset. This demonstrates the effectiveness of Gumbel-Max technique in improving the sequence recommendation using the reparameterization trick, as it provides a more efficient and robust approach to sample from a categorical distribution.

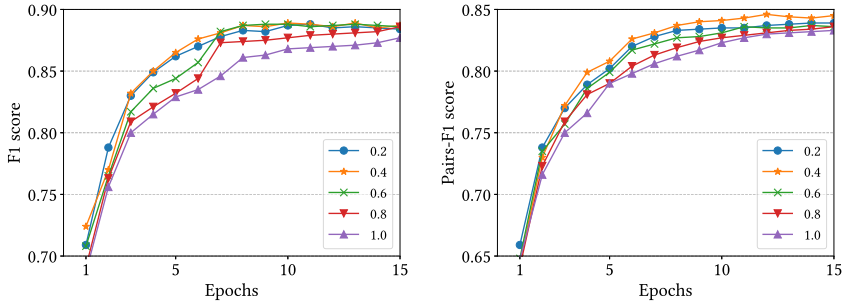


Fig. 5. Influence of temperature value (Osaka dataset).

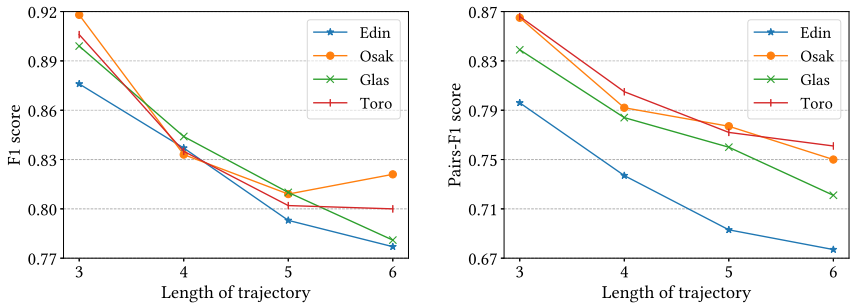


Fig. 6. Influence of trajectory length.

Moreover, the temperature factor controls the softness of the Gumbel-softmax. The higher the value, the smoother the generated distribution and the closer to the uniform distribution. In contrast, the lower the temperature, the closer to the discrete one-hot distribution. To find the optimal temperature value, we vary the temperature value in $\{0.2, 0.4, 0.6, 0.8, 1\}$ on Osaka. The results are shown in Figure 5. Therefore, we empirically set the temperature value to 0.4 in our model.

5.7.3 Influence of Trajectory Length. In our model, the trajectory length is specified by the query. Figure 6 reports the influence of trajectory, which shows that trajectory length is an essential factor affecting the recommendation performance, i.e., the longer the trip length, the lower the values of the two metrics. This result is intuitive, since our model, as well as previous methods, recommend trips iteratively, which may accumulate errors if the initial generated POIs are incorrect. It also raises an open problem in the trip recommendation, i.e., how to alleviate the negative impact of inaccurate POIs on the final generated trip, which is left as our future work.

5.7.4 Time Efficiency. Figure 7 illustrates the training time of the different models. We note that the approaches for solving orienteering problem (e.g., PersTour and PersQueue) are omitted, as they are too time consuming. Markov-based and rank-based methods require more time to estimate the pairwise transition distribution and explicitly rank the POIs. DeepTrip is efficient in learning user transition patterns and visiting preferences; however, it needs to learn trajectory distributions in an adversarial learning manner, which is computation intensive and sometimes cannot converge, primarily because of the sparse and imbalanced data property. As for our models, the CTLTR-Base is fast due to its simple architecture. After data augmentation and contrastive trajectory learning, CTLTR may incur the extra computational cost, which is insignificant in comparison to posterior inference in DeepTrip and the ranking cost in rank-based approaches.

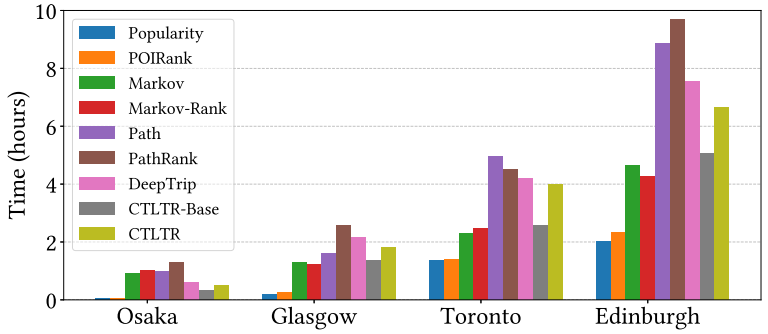


Fig. 7. Efficiency comparisons among different models.

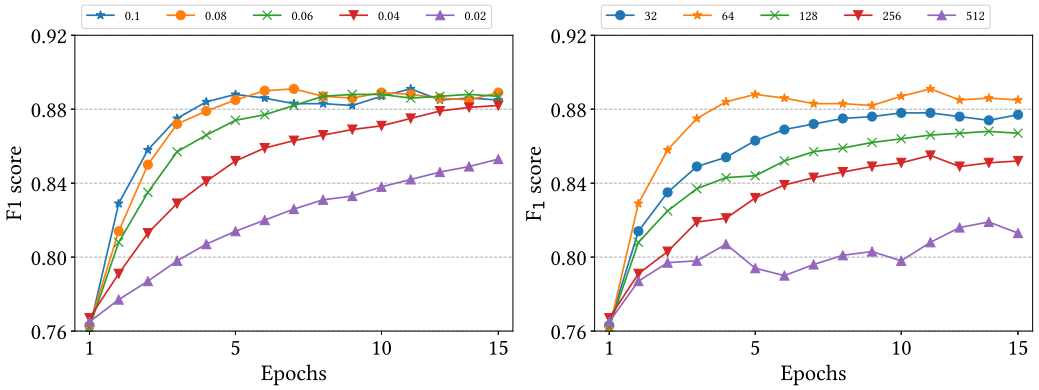


Fig. 8. Parameter tuning in CTLTR. Left: F_1 score with initial learning rate. Right: F_1 score with hidden size.

5.7.5 Training Details. We now discuss the impact of two parameters on training process—the initial learning rate and the LSTM hidden size—reflected in terms of F_1 scores. For the initial learning rate, as illustrated in the left of Figure 8, we fix the hidden size as 64 and vary the initial learning rate from $\{0.02, 0.04, 0.06, 0.08, 0.1\}$ on Osaka dataset. For the hidden size shown in the right of Figure 8, we fix the initial learning rate as 0.1 and tune the hidden size from $\{32, 64, 128, 256, 512\}$. After balancing the overfitting problem and the efficiency, we empirically select 0.1 and 64 as the initial learning rate and hidden size, respectively.

5.7.6 Discussion. As noted, the intuitive way of trajectory data augmentation based on subsequence sampling essentially provides a single view of the original data; however, augmenting the trajectory data to suppose multi-view contrasts provides more robust and permutation invariant representations. For each trajectory, our model samples negative POIs randomly from the other trajectories, introducing bias due to the imbalanced POI visiting frequency, and we note that correcting the bias using the unweighted sampling method requires prior knowledge of the POI/trajectory distribution. Another subtle observation is that the positive samples are already very close and the negative samples are already far away, thus further optimizations are not needed.

6 CONCLUSION AND FUTURE WORK

In this article, we proposed CTLTR, a self-supervised trip recommendation model based on the contrastive method. We adopted the hierarchical recurrent neural network to construct our base

model and devise data augmentation operations as the preliminary of contrastive learning. Based on the mutual information maximization principle, we designed two self-supervised learning objectives to learn human transition patterns and user interests over tours. Extensive experiments conducted on four real-world datasets demonstrated the effectiveness of our method in recommending more accurate tours in terms of both POI correctness and visiting orders, compared with the existing related methods. In addition, we examined the performance of our method through ablation study and parameters tuning.

We have several extensions planned for future work. We are interested in exploring more systematic methods for human trajectory augmentation beyond subsequences used in this article. As mentioned, our first challenge is how to de-bias the model while preserving the benefits of data augmentation. Another interesting challenge is to learn more discriminative data representations in the latent space by incorporating "hard" (both positive and negative) samples during contrastive learning.

In addition, we will attempt to address two global and complementary categories of challenges. One of them will investigate the problem of preventing the trip recommendation from forming a closed-loop in the deep neural network. The other kind of challenge will investigate the trade-offs between learning-based approaches and querying-based approaches in terms of overheads arising from incorporating multiple contexts.

REFERENCES

[1] Jie Bao, Yu Zheng, David Wilkie, and Mohamed F. Mokbel. 2015. Recommendations in location-based social networks: A survey. *GeoInformatica* 19, 3 (2015), 525–565.

[2] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R. Devon Hjelm. 2018. Mine: Mutual information neural estimation. arXiv:1801.04062. Retrieved from https://arxiv.org/abs/1801.04062.

[3] Paolo Bolzoni, Sven Helmer, Kevin Wellenzohn, Johann Gamper, and Periklis Andritsos. 2014. Efficient itinerary planning with category constraints. In *SIGSPATIAL*. 203–212.

[4] Igo Brilhante, Jose Antonio Macedo, Franco Maria Nardini, Raffaele Perego, and Chiara Renso. 2013. Where shall we go today? Planning touristic tours with TripBuilder. In *CIKM*. 757–762.

[5] Dawei Chen, Cheng Soon Ong, and Lexing Xie. 2016. Learning points and routes to recommend trajectories. In *CIKM*. 2227–2232.

[6] Lisi Chen, Shuo Shang, Christian S. Jensen, Bin Yao, Zhiwei Zhang, and Ling Shao. 2019. Effective and efficient reuse of past travel behavior for route recommendation. In *SIGKDD*. 488–498.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *ICML*. 1597–1607.

[8] An-Jung Cheng, Yan-Ying Chen, Yen-Ta Huang, Winston H. Hsu, and Hong-Yuan Mark Liao. 2011. Personalized travel recommendation by mining people attributes from community-contributed photos. In *MM*. 83–92.

[9] Henry Crosby, Theodoros Damoulas, and Stephen A. Jarvis. 2019. Embedding road networks and travel time into distance metrics for urban modelling. *Int. J. Geogr. Inf. Sci.* 33, 3 (2019), 512–536.

[10] Yue Cui, Liwei Deng, Yan Zhao, Bin Yao, Vincent W. Zheng, and Kai Zheng. 2019. Hidden POI ranking with spatial crowdsourcing. In *SIGKDD*. 814–824.

[11] Amine Dadoun, Raphaël Troncy, Olivier Ratier, and Riccardo Petitti. 2019. Location embeddings for next trip recommendation. In *WWW Companion*. 896–903.

[12] Munmun De Choudhury, Moran Feldman, Sihem Amer-Yahia, Nadav Golbandi, Ronny Lempel, and Cong Yu. 2010. Constructing travel itineraries from tagged geo-temporal breadcrumbs. In *WWW*. 1083–1084.

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*. 4171–4186.

[14] Jie Feng, Yong Li, Chao Zhang, Funing Sun, Fanchao Meng, Ang Guo, and Depeng Jin. 2018. DeepMove: Predicting human mobility with attentional recurrent networks. In *WWW*. 1459–1468.

[15] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized ranking metric embedding for next new POI recommendation. In *IJCAI*. 2069–2075.

[16] Zachary Friggstad, Sreenivas Gollapudi, Kostas Kollias, Tamás Sarlós, Chaitanya Swamy, and Andrew Tomkins. 2018. Orienteering algorithms for generating travel itineraries. In *WSDM*. 180–188.

[17] Qiang Gao, Goce Trajcevski, Fan Zhou, Kunpeng Zhang, Ting Zhong, and Fengli Zhang. 2019. DeepTrip: Adversarially understanding human mobility for trip recommendation. In *SIGSPATIAL*. 444–447.
[18] Qiang Gao, Fan Zhou, Kunpeng Zhang, Fengli Zhang, and Goce Trajcevski. 2021. Adversarial human trajectory learning for trip recommendation. In *IEEE Transactions on Neural Networks and Learning Systems*.
[19] Yong Ge, Qi Liu, Hui Xiong, Alexander Tuzhilin, and Jian Chen. 2011. Cost-aware travel tour recommendation. In *SIGKDD*. 983–991.
[20] Aristides Gionis, Theodoros Lappas, Konstantinos Pelechrinis, and Evimaria Terzi. 2014. Customized tour recommendations in urban areas. In *WSDM*. 313–322.
[21] Jiqing Gu, Chao Song, Wenjun Jiang, Xiaomin Wang, and Ming Liu. 2020. Enhancing personalized trip recommendation with attractive routes. In *AAAI*. 662–669.
[22] Aldy Gunawan, Hoong Chuin Lau, and Pieter Vansteenwegen. 2016. Orienteering problem: A survey of recent variants, solution approaches and applications. *Eur. J. Oper. Res.* 255, 2 (2016), 315–332.
[23] Michael Gutmann and Aapo Hyvärinen. 2010. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *AISTATS*. 297–304.
[24] Jiayuan He, Jianzhong Qi, and Kotagiri Ramamohanarao. 2019. A joint context-aware embedding for trip recommendations. In *ICDE*. 292–303.
[25] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*. 9729–9738.
[26] Daniel Herzog, Sherjeel Sikander, and Wolfgang Wörndl. 2019. Integrating route attractiveness attributes into tourist trip recommendations. In *WWW Companion*. 96–101.
[27] R. Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. In *ICLR*.
[28] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neur. Comput.* 9, 8 (1997), 1735–1780.
[29] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical reparameterization with gumbel-softmax. In *ICLR*.
[30] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised contrastive learning. arXiv:2004.11362. Retrieved from https://arxiv.org/abs/2004.11362.
[31] Lingpeng Kong, Cyprien de Masson d'Autume, Wang Ling, Lei Yu, Zihang Dai, and Dani Yogatama. 2020. A mutual information maximization perspective of language representation learning. In *ICLR*.
[32] Yaguang Li, Kun Fu, Zheng Wang, Cyrus Shahabi, Jieping Ye, and Yan Liu. 2018. Multi-task representation learning for travel time estimation. In *SIGKDD*. 1695–1704.
[33] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. 2017. Personalized itinerary recommendation with queuing time awareness. In *SIGIR*. 325–334.
[34] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. 2019. Tour recommendation and trip planning using location-based social media: A survey. *Knowl. Inf. Syst.* 60, 3 (2019), 1247–1275.
[35] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2015. Personalized tour recommendation based on user interests and points of interest visit durations. In *IJCAI*. 1778–1784.
[36] Xiao Liu, Fanjin Zhang, Zhenyu Hou, Zhaoyu Wang, Li Mian, Jing Zhang, and Jie Tang. 2020. Self-supervised learning: Generative or contrastive. arXiv:2006.08218. Retrieved from https://arxiv.org/abs/2006.08218.
[37] Eric Hsueh-Chan Lu, Ching-Yu Chen, and Vincent S. Tseng. 2012. Personalized trip recommendation with multiple constraints by mining user check-in behaviors. In *SIGSPATIAL*. 209–218.
[38] Xin Lu, Changhu Wang, Jiang-Ming Yang, Yanwei Pang, and Lei Zhang. 2010. Photo2Trip: Generating travel routes from geo-tagged photos for trip planning. In *MM*. 143–152.
[39] Jianxin Ma, Chang Zhou, Hongxia Yang, Peng Cui, Xin Wang, and Wenwu Zhu. 2020. Disentangled self-supervision in sequential recommenders. In *SIGKDD*. 483–491.
[40] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *ICLR (Workshop)*.
[41] Tomas Mikolov, Ilya Sutskever, Kai Chen, G. S. Corrado, and J. Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*. 3111–3119.
[42] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748. Retrieved from https://arxiv.org/abs/1807.03748.
[43] Lars Schmarje, Monty Santarossa, Simon-Martin Schröder, and Reinhard Koch. 2020. A survey on semi-, self-and unsupervised techniques in image classification. arXiv:2002.08721. Retrieved from https://arxiv.org/abs/2002.08721.
[44] Ke Sun, Tieyun Qian, Tong Chen, Yile Liang, Quoc Viet Hung Nguyen, and Hongzhi Yin. 2020. Where to go next: Modeling long- and short-term user preferences for point-of-interest recommendation. In *AAAI*. 214–221.
[45] Kendall Taylor, Kwan Hui Lim, and Jeffrey Chan. 2018. Travel itinerary recommendations with must-see points-of-interest. In *WWW Companion*. 1198–1205.

[46] Xu Teng, Goce Trajcevski, Joon-Seok Kim, and Andreas Züfle. 2020. Semantically diverse path search. In *MDM*. 69–78.
[47] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Commun. ACM* 59, 2 (2016), 64–73.
[48] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive multiview coding. In *ECCV*. 776–794.
[49] Yongxin Tong, Zimu Zhou, Yuxiang Zeng, Lei Chen, and Cyrus Shahabi. 2020. Spatial crowdsourcing: A survey. *VLDB J.* 29, 1 (2020), 217–250.
[50] Pieter Vansteenwegen, Wouter Souffriau, Greet Vanden Berghe, and Dirk Van Oudheusden. 2011. The city trip planner: An expert system for tourists. *Expert Syst. Appl.* 38, 6 (2011), 6540–6546.
[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.
[52] Jingyuan Wang, Ning Wu, Wayne Xin Zhao, Fanzhang Peng, and Xin Lin. 2019. Empowering A* search algorithms with neural networks for personalized route recommendation. In *SIGKDD*. 539–547.
[53] Ling-Yin Wei, Yu Zheng, and Wen-Chih Peng. 2012. Constructing popular routes from uncertain trajectories. In *SIGKDD*. 195–203.
[54] Hanwei Wu, Ather Gattami, and Markus Flierl. 2020. Conditional mutual information-based contrastive loss for financial time series forecasting. In *ICAIF*. 9:1–9:7.
[55] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.
[56] Xin Xin, Alexandros Karatzoglou, Ioannis Arapakis, and Joemon M. Jose. 2020. Self-supervised reinforcement learning for recommender systems. In *SIGIR*. 931–940.
[57] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Aditya Menon, Lichan Hong, Ed H. Chi, Steve Tjoa, Evan Ettinger, et al. 2020. Self-supervised learning for deep models in recommendations. arXiv:2007.12865. Retrieved from https://arxiv.org/abs/2007.12865.
[58] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph contrastive learning with augmentations. In *NeurIPS*.
[59] Chenyi Zhang, Hongwei Liang, Ke Wang, and Jianling Sun. 2015. Personalized trip recommendation with POI availability and uncertain traveling time. In *CIKM*. 911–920.
[60] Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *ACL*. 4334–4343.
[61] Pengpeng Zhao, Xiefeng Xu, Yanchi Liu, Victor S. Sheng, Kai Zheng, and Hui Xiong. 2017. Photo2Trip: Exploiting visual contents in geo-tagged photos for personalized tour recommendation. In *MM*. 916–924.
[62] Shenglin Zhao, Irwin King, and Michael R. Lyu. 2016. A survey of point-of-interest recommendation in location-based social networks. arXiv:1607.00647. Retrieved from https://arxiv.org/abs/1607.00647.
[63] Ting Zhong, Shengming Zhang, Fan Zhou, Kunpeng Zhang, Goce Trajcevski, and Jin Wu. 2020. Hybrid graph convolutional networks with multi-head attention for location recommendation. *World Wide Web* 23, 6 (2020), 3125–3151.
[64] Fan Zhou, Xin Liu, Ting Zhong, and Goce Trajcevski. 2021. MetaMove: On improving human mobility classification and prediction via metalearning. (unpublished). https://doi.org/10.1109/TCYB.2021.3049533
[65] Fan Zhou, Hantao Wu, Goce Trajcevski, Ashfaq A. Khokhar, and Kunpeng Zhang. 2020. Semi-supervised trajectory understanding with POI attention for end-to-end trip recommendation. *Trans. Spatial. Algor. Syst.* 6, 2 (2020), 13:1–13:25.
[66] Fan Zhou, Xovee Xu, Goce Trajcevski, and Kunpeng Zhang. 2021. A survey of information cascade analysis: Models, predictions, and recent advances. *Comput. Surv.* 54, 2, Article 27 (2021), 36 pages. https://doi.org/10.1145/3433000
[67] Fan Zhou, Qing Yang, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Ashfaq Khokhar. 2020. Reinforced spatiotemporal attentive graph neural networks for traffic forecasting. *IEEE IoT J.* 7, 7 (2020), 6414–6428.
[68] Fan Zhou, Qing Yang, Ting Zhong, Dajiang Chen, and Ning Zhang. 2020. Variational graph neural networks for road traffic prediction in intelligent transportation systems. *IEEE Transactions on Industrial Informatics* 17, 4 (2020), 2802–2812.
[69] Fan Zhou, Xiaoli Yue, Goce Trajcevski, Ting Zhong, and Kunpeng Zhang. 2019. Context-aware variational trajectory encoding and human mobility inference. In *WWW*. 3469–3475.
[70] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*. 1893–1902.

Received December 2020; revised March 2021; accepted April 2021