Beyond Relevance: Factor-level Causal Explanation for User Travel Decisions with Counterfactual Data Augmentation

HANZHE LI, Nanjing University of Aeronautics and Astronautics, Nanjing, China
JINGJING GU, Nanjing University of Aeronautics and Astronautics, Nanjing, China
XINJIANG LU, Baidu Business Intelligence Lab, Baidu Research, Beijing, China
DAZHONG SHEN, Shanghai Artificial Intelligence Laboratory, Shanghai, China
YUTING LIU, Nanjing University of Aeronautics and Astronautics, Nanjing, China
YANAN DENG, Nanjing University of Aeronautics and Astronautics, Nanjing, China
GUOLIANG SHI, Nanjing University of Aeronautics and Astronautics, Nanjing, China
HUI XIONG, Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

Point-of-Interest (POI) recommendation, an important research hotspot in the field of urban computing, plays a crucial role in urban construction. While understanding the process of users' travel decisions and exploring the causality of POI choosing is not easy due to the complex and diverse influencing factors in urban travel scenarios. Moreover, the spurious explanations caused by severe data sparsity, i.e., misrepresenting universal relevance as causality, may also hinder us from understanding users' travel decisions. To this end, in this article, we propose a factor-level causal explanation generation framework based on counterfactual data augmentation for user travel decisions, named Factor-level Causal Explanation for User Travel Decisions (FCE-UTD), which can distinguish between true and false causal factors and generate true causal explanations. Specifically, we first assume that a user decision is composed of a set of several different factors. Then, by preserving the user decision structure with a joint counterfactual contrastive learning paradigm, we learn the representation of factors and detect the relevant factors. Next, we further identify true causal factors by constructing counterfactual decisions with a counterfactual representation generator, in particular, it can not only augment the dataset and mitigate the sparsity but also contribute to clarifying the causal factors from other false causal factors that may cause spurious explanations. Besides, a causal dependency learner is proposed to identify causal factors for each decision by learning causal dependency scores. Extensive experiments conducted on three real-world datasets demonstrate the superiority of our approach in terms of check-in rate, fidelity, and downstream tasks under different behavior scenarios. The extra case studies also demonstrate the ability of FCE-UTD to generate causal explanations in POI choosing.

This is a substantially extended and revised version of Reference [14], which appears in the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI'20).

This work is supported in part by the National Natural Science Foundation of China under Grant (No. 62072235).

Authors' addresses: H. Li, J. Gu (Corresponding author), Y. Liu, Y. Deng, and G. Shi, Nanjing University of Aeronautics and Astronautics, Nanjing, China; e-mails: lihanzhe@nuaa.edu.cn, gujingjing@nuaa.edu.cn, yuting_liu@nuaa.edu.cn, dyn@nuaa.edu.cn, shiguoliang@nuaa.edu.cn; X. Lu, Baidu Business Intelligence Lab, Baidu Research, Beijing, China; e-mail: luxinjiang@baidu.com; D. Shen, Shanghai Artificial Intelligence Laboratory, Shanghai, China; e-mail: dazh.shen@gmail.com; H. Xiong, Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China; e-mail: xionghui@ust.hk.

ACM 1046-8188/2024/04-ART128
https://doi.org/10.1145/3653673

CCS Concepts: • **Information systems** → **Data mining;**

Additional Key Words and Phrases: Causal explanation generation, urban travel decisions, counterfactual data augmentation, contrastive learning

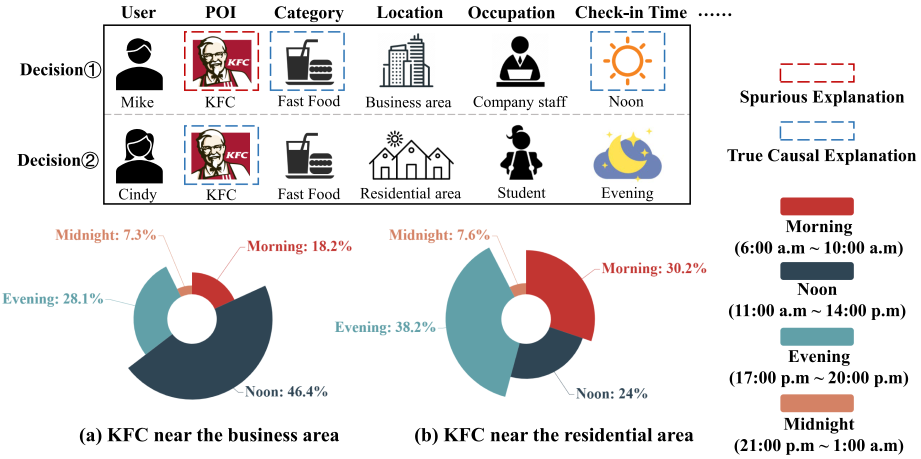
ACM Reference Format:
Hanzhe Li, Jingjing Gu, Xinjiang Lu, Dazhong Shen, Yuting Liu, YaNan Deng, Guoliang Shi, and Hui Xiong. 2024. Beyond Relevance: Factor-level Causal Explanation for User Travel Decisions with Counterfactual Data Augmentation. *ACM Trans. Inf. Syst.* 42, 5, Article 128 (April 2024), 31 pages. https://doi.org/10.1145/3653673

1 INTRODUCTION

With the rapid development of smart mobile devices, **Location-Based Social Networks (LBSNs)** have become ubiquitous in our daily lives. People use LBSN to visit a variety of locations (i.e., **Point-of-Interest (POI)**) in different categories, generating a large number of check-in logs (i.e., check-in behavior). For example, Foursquare had recorded 10 billion check-ins at 93 million POIs from more than 50 million users in this platform by 2016 [11]. The increasing log data becomes the basis for exploring user preferences and recommending POIs interested but never visited for users. As a result, POI recommendation has been extensively researched and achieved remarkable results in recent years [10, 54, 63]. However, how to interpret recommendation results and understand the process of decision-making is still an unignorable challenge, which is crucial for the transparency, persuasiveness, and trustworthiness of recommendation systems [30, 61].

In the literature, some research utilized external data to generate explanations for traditional recommendations, such as user ratings and reviews [4, 13, 36]. However, these efforts are limited by insufficient external data and incomplete feature selection, which cannot provide a universal solution. Besides, some works generated explanations by mining the relevance between items from the sufficient data in the traditional recommendation scenario and achieved promising results [1, 2, 6, 41]. However, most of the above works ignored a situation that some items with strong relevance are not the reason (i.e., true causal explanation) for user decisions. And if we treat them as explanations, then the spurious explanations will affect the validity and reliability of the model and lead to difficulties in the execution of downstream tasks. Moreover, the data sparsity in city travel is serious owing to the difficulty of collecting sufficient check-in logs and the limitation of the scope of user activities, which further aggravates the aforementioned issue.

Actually, a user decision is composed of a set of factors, including user and item attributes. In addition to being influenced by user and POI attributes, user decisions in city travel are also susceptible to complex spatio-temporal factors (e.g., user query and check-in time, and geographical distances), which is different from traditional recommendation scenarios (e.g., e-commerce). Figure 1 shows the check-in time distribution of KFC in two different areas and two user decisions. In *Decision1*, Mike is a company employee and had lunch at the KFC near the company. The category factor *Fast Food* of KFC and the check-in time *Noon* may be the true reason (i.e., true causal explanation) why he checked-in this KFC, instead of the POI brand *KFC* (i.e., spurious explanation). While in *Decision2*, Cindy is a student who went to KFC after school in the evening, and the decision happened probably because she likes KFC. The check-in time distribution of KFC also shows that visits to KFC in the business area are mostly concentrated at noon, while those in the residential area are mainly concentrated in the evening, which is consistent with the analysis above. In other words, the process of decision-making is caused by different spatio-temporal factors to varying degrees in different scenarios. Therefore, it is important to explore reasons for user decisions and identify those true causal factors, which can further help us explore user preferences and contribute to business planning. Indeed, mining causal factors of user travel decisions faces



(a) KFC near the business area

(b) KFC near the residential area

Fig. 1. Examples of user travel decision. The top part of this figure represents two user decisions that consists of several factors. And the bottom part shows the check-in time distribution of KFC in two different areas corresponding to user decisions, where different colors mean different time slots. Note that the red dashed box indicates spurious explanations and the blue dashed box indicates true causal explanations.

two challenges: (1) Complex and diverse spatio-temporal factors in user travel scenarios make it difficult to mine causal relationships of user decisions. (2) Severe data sparsity in user travel scenarios hinders the generation of true causal explanations. And spurious explanations will mislead us and have an impact on model performance.

To this end, in this article, we propose a factor-level causal explanation generation framework based on counterfactual data augmentation for user travel decisions, named **Factor-level Causal Explanation for User Travel Decisions (FCE-UTD)**. Specifically, FCE-UTD is mainly composed of three parts. In the *Input Preparation* module, each factor included in a user decision is embedded initially. In the *Relevant Factors Learning* module, a joint counterfactual contrastive learning paradigm is designed to optimize the embeddings of all factors in a pre-training manner. In particular, the relevant factors will also be detected with a self-projection attention mechanism to score each factor. Then in the *Causal Explanation Learning* module, we further identify causal factors from those relevant factors based on a universal knowledge: *Causality has strong relevance.* A counterfactual representation generator is designed to construct counterfactual decisions that can not only augment the dataset and alleviate the sparsity but also contribute to clarifying the causal factors from other spurious explanations. Along this line, a simple causal dependency learner is proposed to rank the causal dependency for each factor in a decision, and, finally, factor-level causal explanations would be generated by outputting the intersection of top factors with the highest causal dependency and relevant factors. Overall, the primary contributions are summarized as follows:

- To the best of our knowledge, we are the first to generate factor-level causal explanations for user travel decisions, which are susceptible to complex spatio-temporal factors. As a result, a novel approach, named FCE-UTD, is proposed to mine the causal dependency among varying factors and user decisions, which can avoid generating spurious explanations by jointly using real and counterfactual data.
- A novel joint counterfactual contrastive learning paradigm is designed with a self-projection attention mechanism to learn the embedding of all factors and mine their relevant dependency for user decisions.

- We design a counterfactual representation generator to generate counterfactual decisions, alleviating the data sparsity and contributing to exploring more potential causal factors with a simple causal dependency learner.
- We conduct extensive experiments on three real-world datasets to evaluate our FCE-UTD framework. The results demonstrate the superiority in terms of check-in rate, fidelity, and downstream tasks (i.e., recommendation) under different behavior scenarios.

2 RELATED WORK

Explainable recommendation systems not only output recommendation results but also generate explanations to clarify why such items are recommended [61]. Traditional matrix factorization methods focus on providing accurate recommendations to users while failing to interpret the hidden decision logic, which leads to the explicit factor model [62] and tensor factorization–based methods [7]. Recently, different deep neural models are widely induced into recommendation systems, where the explanations can also be explored due to their powerful expressivity and flexibility. For example, Reference [39] modeled user preferences and item attributes based on user reviews by convolutional neural networks and attention mechanisms, and generated explanations. Similarly, Reference [50] utilized the convolution operations and attention mechanism to highlight the relevant semantic information from reviews, which can uncover user preferences and improve explainability for recommendation. Reference [9] proposed a deep model based on attentive multi-view learning to mitigate the tradeoff between accuracy and explainability. Reference [44] incorporated auxiliary knowledge with memory networks for sequence to sequence modeling and obtained explanation based on the annotations produced by attention mechanism over memory. Reference [20] designed a context-prediction task that maps user or item IDs onto words to be generated by the explanation task and then presented a personalized Transformer to make recommendations and generate explanations simultaneously based on IDs. Furthermore, researchers have been exploring **knowledge graphs (KG)** that contain rich information about users and items to generate more intuitive explanations for recommended items. Reference [16] constructed a knowledge graph to mine the attribute-level preferences for recommendation and then generated explanations based on attributes that have an impact on prediction. Reference [53] proposed a policy-guided path reasoning method to reason over knowledge graph and generate explanations with reasoned paths. Reference [24] proposed to integrate explainable rule induction in knowledge graphs with a rule-guided recommendation model and translate the mined inductive rules into explanations. Unlike previous KG-based explainable recommendation works, Reference [32] analyzed user's reviews and ratings on items to construct a sentiment-aware knowledge graph that can reason more convincing explanations with a sentiment-aware policy learning methods. As for the scenarios of urban POI travel, the related literature is limited. Reference [51] developed a topic model for explainable hotel recommendations that generates a topical word cloud explanation on hotel features. Reference [14] studied user decision profiling with a scalar projection maximization objective and generated explanations based on the identified key factors. In summary, different models generated explanations for POI travel in different ways, i.e., the topic model and the identification of key factors. However, these methods mostly described how to learn the relevance between corresponding items and therefore might be mislead by spurious explanations. Conversely, our approach focuses on mining causality from relevance, which aims to reduce the impact of spurious explanations and identify true causal explanations.

Contrastive learning is a kind of self-supervised learning, which was widely used in Computer Vision and **Natural Language Processing (NLP)** fields. It learns quality discriminative representations by constructing positive and negative instances [59]. Reference [31] proposed InfoNCE loss, which maximizes the mutual information between positive sample pairs and minimizes the mutual information between negative sample pairs, to learn latent feature representations.

Reference [38] constructed triplets, which consists of two matching face images and a non-matching one, to learn feature representations of images by separate the positive pair from the negative by a distance margin in an end-to-end architecture.

Due to the flexibility and promising performance of contrastive self-supervised learning, it has recently become a research hotspot among recommendation methods based on self-supervised learning [17]. Reference [49] first applied contrastive learning to graph-based recommendation. It generated two different graph views based on the user–item intersection graph, and performed the node self-discrimination task on positive and negative node pairs in different views respectively, to learn more generalized representations. Reference [23] perturbed the L-hop ego-network of each node with a stochastic edge dropout and obtain two augmented neighborhood subgraphs, then maximized agreement between node representations learned on the two subgraphs. Reference [40] utilized item ratings and corresponding review semantics to generate feature-enhanced edges and construct a review-aware user–item graph with these edges, then designed a contrastive objective that maximizes the mutual information between the review representation and the corresponding interaction representation. To capture local and global collaborative relations in user–item intersections, Reference [52] constructed two views, including a user–item interaction graph and a learnable hypergraph, then proposed a hypergraph-enhanced cross-view contrastive learning architecture based on the two graphs.

Recently, contrastive learning has been applied to POI recommendations. For example, Reference [55] adopted a random sampling methods to augment user check-in sequences and then proposed a contrastive self-supervised learning framework with the generated sequences to improve the POI recommendation. Different from Reference [55], which only considered random sampling to augment check-in sequences, Reference [22] substituted POIs with highly correlated POIs to maintain the correlations in check-in sequences and enhance the robustness. Reference [8] first modelled users' intent distributions from all user check-in sequences via clustering and then fused the learnt intents into a POI recommendation model with a new contrastive objective to improve model robustness. Reference [35] extracted preference proxies from check-in sequences and utilized them to improve POI embedding quality via a contrastive objective. Moreover, Reference [65] proposed a **Bidirectional Encoder Representations from Transformers (BERT)** based model with four auxiliary self-supervised objectives to learn user and POI representations. For our work, the proposed contrastive learning module that is based on mined relevant factors can learn more accurate and robust representations effectively.

Counterfactual perspective aims to answer a question related to the factual world (i.e., the observational data) and the counterfactual world: "What would...if...?" [47]. In short, it is to apply a perturbation to the original data and observe how results change. Many researchers designed explainable and robust models from counterfactual perspective, which have achieved remarkable results [18, 19, 42, 43, 48]. For example, Reference [43] designed a counterfactual explainable recommendation framework, which generates explanations based on counterfactual changes on item aspects. Reference [48] introduced a framework to eliminate popularity bias in recommendation, which adopted counterfactual reasoning to estimate the direct effect from items to ranking scores, and removed it to eliminate bias.

In addition, another important application of the counterfactual perspective is to augment data to alleviate the data sparsity. Reference [47] designed heuristics and learning-based methods with counterfactual perspective to enrich user behavior sequences and improve the recommendation performance. Reference [56] generated extra training samples via changing the users' feature-level preferences, to alleviate the data sparsity for improving review-based recommendation. Moreover, Reference [58] generated extra informative training samples with a learning-based intervention method to mitigate the exposure bias caused by data sparsity. Reference [29] adopted

Table 1. Factors in POI Travel Behaviors

Type	POI Travel
User-related	User identifier, Frequency of visiting different POI categories
POI-related	POI identifier, Category, Brand, POI popularity
Spatio-temporal related	Check-in time, The day of the week, Time session of query, Location distance between query and check-in POI

reinforcement learning to build counterfactual generators for generating high-quality counterfactual intersections and learned a recommender from factual and counterfactual interactions to remove spurious correlations. Our work could also be classified as this category, and we achieve the idea of a counterfactual perspective in urban POI travel by constructing factor-level counterfactual user decisions and generating causal explanations.

3 PROBLEM DEFINITION

In this section, we formally define the prediction problem of user decision check-in rate and its factor-level causal explanation generation problem. We start by defining some basic concepts and notations. The types of factors and the mathematical notations used in this article are listed in Table 1 and Table 2, respectively. Note that we use bold for representation vectors and calligraphic fonts for sets to achieve a clearer description.

Definition 1 (Factor). A factor f denotes an item that has an impact on the user's decision process, with a concrete explanation. We define the set of all factors as $\mathcal{F}$ and the factor lookup table as E.

Similarly to previous work [14], we define well-designed factors with practical meanings to guarantee the interpretability of user decisions as well as generated causal explanations. To consider all aspects of influence as much as possible, we define three types of factors as shown in Table 1, i.e., user-related, POI-related, and spatio-temporal related.

First, user-related factors include the user identifier and the frequency of different POI categories they frequently visited. The former is used to model distinct impacts of different users and the latter can reflect their preferences. Second, POI-related factors contain POI identifier, category, brand, and POI popularity. The POI identifier can distinguish the impact of different POIs, which is similar to the usage of user identifiers. Besides, the POI popularity is defined in a similar way to Reference [14], we divide the continuous popularity into six levels according to the standard scores z of log-scaled popularity: (i) *strongly unpopular* if $z \leq -1$, (ii) *unpopular* if $z \in (-1, -0.5]$, (iii) *weakly unpopular* if $z \in (-0.5, 0]$, (iv) *weakly popular* if $z \in (0, 0.5]$, (v) *popular* if $z \in (0.5, 1]$, and (vi) *strongly popular* if $z > 1$. Third, spatio-temporal related factors are the time relevant to user decision, e.g., arriving hour, the day of the week, and the time session of user query POI. In addition, the location distance between user check-in and query POI is also considered. We divide the distance into five levels: (i) *1 km and less*, (ii) *1 km–3 km*, (iii) *3 km–7 km*, (iv) *7 km–15 km*, and (v) *15 km and more*.

Definition 2 (Decision). In the POI travel scenario, a decision D represents a user check-in and consists of a set of factors, i.e., $D = \{f_1, \ldots, f_n\}$. We define the set of all decisions as $\mathcal{D}$. For convenience, we use F to denote all factors in D.

Definition 3 (Causal Relation). Suppose there are two variables A and B. If A leads to B, then there is a causal relation $A \Rightarrow B$, where A and B are cause and effect, respectively.

It is worth noting that, given a decision $D = \{f_1, \ldots, f_n\}$, all factors in D are considered as its potential causes.

Table 2. Mathematical Notations

Symbols	Description
n	The number of factors that compose a decision;
m	The number of sampling counterfactual representations for each decision;
$D = \{f_1, \ldots, f_n\}$	A user travel decision composed of n different factors;
f_i, F^D	The ith factor and all factors in D;
F^D_{rel}	The relevant factors of D;
$\hat{r}(D), r(D)$	The empirical and predictive check-in rate for decision D;
W_1, W_2, W_3	The trainable matrices parameters of MLP in sparse likelihood estimator;
Δ^{n-1}	The n-dimensional probability simplex;
L_d, L_c	The BCE loss for optimizing decision structure learner and the counterfactual contrastive triplet learning paradigm;
δ, γ	The margin in L_c and the hyper-parameter to control the contribution of L_c;
k_1, τ	The replacement number in counterfactual contrastive learner and the corresponding hyper-parameter;
k_2	The number of highest causal dependency pairs selected;
G	The VAE-based counterfactual representation generator;
$d(\cdot, \cdot)$	The Euclidean distance function;
$R(\cdot)$	The check-in rate prediction model in the relevant factors learning module;
$Q_\theta(z\|x), p_\phi(x\|z)$	The inference network and generation network in G;
$p(z)$	The prior distribution in G;
λ	The trading-off hyper-parameter in G;
$(\widetilde{F_i}^D, \widetilde{Y}_i^D)$	The counterfactual decision–label pairs of D;
$(\widehat{F}_i^D, \widehat{Y}_i^D)$	The augmented decision–label pairs of D;
$(\widehat{f}_{ij}^D, \widehat{Y}_i^D), T$	The unique factor–label pairs extracted from $\mathcal{S}^D$ and their number;
H	The one-hot vector of the T unique factor–label pairs;
θ^D	The trainable causal dependency of the T unique pairs;
$\boldsymbol{f_i}, \hat{\boldsymbol{f}}_{\boldsymbol{i}}$	The factor embedding and the attention factor embedding of f_i;
$\boldsymbol{F}, \hat{\boldsymbol{F}}$	The factor embedding matrix and the attention embedding matrix of decision D;
$\boldsymbol{E}$	The embedding lookup table of all factors;
$\boldsymbol{P_{ij}}, \boldsymbol{P}$	The scalar projection of f_j on f_i and the scalar projection matrix;
$\hat{\boldsymbol{P}}_{\boldsymbol{i}:}$	The representation after applying softmax to each row of $\boldsymbol{P}$;
$\boldsymbol{l}, \hat{\boldsymbol{l}}$	The dense likelihood vector obtained via MLP and the sparse likelihood vector obtained by applying sparsemax on $\boldsymbol{l}$;
$\boldsymbol{d}$	The aggregated relevant embedding of D;
$\boldsymbol{f}'$	The sum of all factor embeddings in D;
$\boldsymbol{F_p}, \boldsymbol{F_n}$	The positive and negative counterfactual samples obtained by transforming the original decision in counterfactual contrastive learner;
$\boldsymbol{F}_{\boldsymbol{i}}^{\boldsymbol{c}}$	The ith counterfactual representation for D obtained by sampling from G;
$\widetilde{\boldsymbol{F}_{\boldsymbol{i}}}^D$	The factors embeddings of mapped counterfactual representations (i.e., counterfactual decisions) for D;
$\mathcal{D}, \mathcal{F}$	The set of all user travel decisions and all factors;
$\mathcal{D}^-$	The set of negative decisions;
$\mathcal{E}$	The causal explanation for a decision D;
$\mathcal{S}^D$	The set of augmented decision–label pairs of decision D;
C	The factor parts of top-k_2 causal dependency pairs;

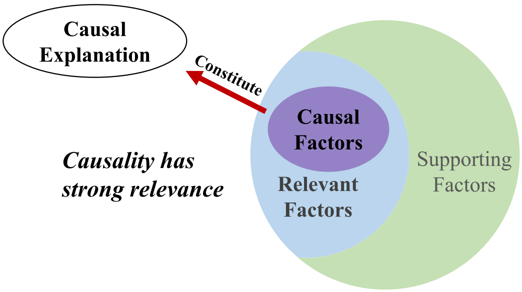


Fig. 2. Illustration for the relationship between various factors. The factors with high relevance to a decision are regarded as relevant factors, while the rest that have minor impact are supporting factors. Besides, causal factors have strong relevance and are a subset of relevant factors. They constitute the true causal explanation of a decision.

Definition 4 (Causal Explanation of Decision). Given a decision $D = \{f_1, \ldots, f_n\}$, if there exists a factor f_i in $\{f_1, \ldots, f_n\}$ that is the cause of D, then f_i is regarded as a causal factor and a part of the causal explanation of decision D. Noting that the causal explanation may consist of one or more factors.

To discover the causes of one decision, i.e., the true causal explanation, in its potential causes, we can utilize likelihood estimation to model the probability of each candidate pair (f_i, D) being a causal explanation.

Problem 1 (Check-in Rate Prediction). *Given the particular set of factors related to a decision D, the problem of check-in rate prediction aims to predict the probability of D, which can also be seen as a classification problem.*

For each decision D, we denote its predictive check-in rate as $r(D)$, where $r(D) > 0$. Note that for historical decisions made by users, i.e., users' check-in history, we define their empirical check-in rate $\hat{r}(D)$ as 1 and call them *positive decision instances*. Conversely, we define the empirical check-in rate $\hat{r}(D)$ of *negative decision instances* as 0, which will be explained in detail later.

Problem 2 (Causal Explanation Generation). *Given a positive decision D, we generate a causal explanations $\mathcal{E}$, which consists of one or more factors $\{f_1', \ldots, f_m'\}$, indicating $\mathcal{E} \Rightarrow D$.*

4 METHODOLOGY

In this section, we introduce our factor-level causal explanation generation framework in detail. First, our work is actually inspired by the following conjecture.

Conjecture 1 (Causality Has Strong Relevance). *Relevance reflects the degree to which two variables are associated with each other, which is only a necessary but insufficient condition for causality. However, in turn, having causality is inevitably accompanied by strong relevance, which implies a special kind of relevance.*

The relationship among relevant, causal, and other supporting factors can be illustrated in Figure 2. Note that the term "relevance" is also expressed as "correlation," and we use the former for better understanding. Along this line, for each decision, we tend to identify relevant factors first, and further mine causal factors from them to obtain the true causal explanation. Specifically, the overview of FCE-UTD framework is illustrated in Figure 3, which consists of three parts, i.e., Input Preparation Module, Relevant Factors Learning Module, and Causal Explanation Learning Module, as follows:

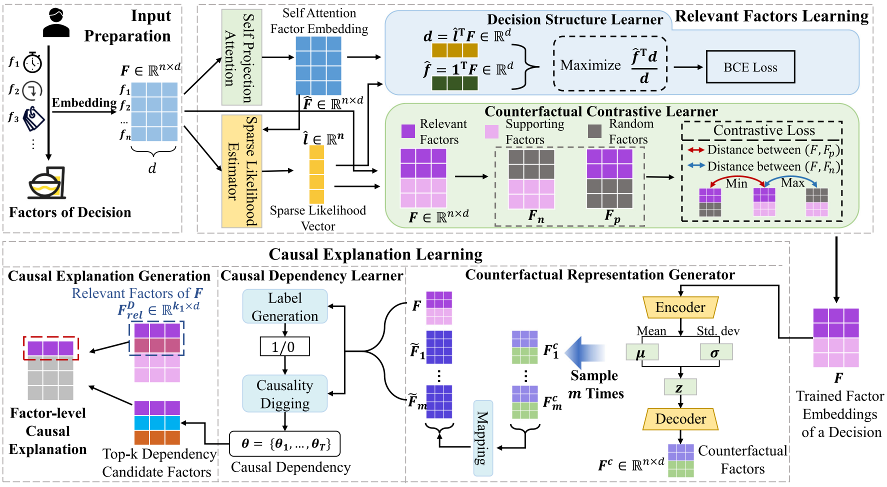

Input Preparation
f_1
f_2
f_3
Embedding
$F \in \mathbb{R}^{n \times d}$
f_1
f_2
...
f_n
d
Factors of Decision
Self Projection Attention
Self Attention Factor Embedding
$\hat{F} \in \mathbb{R}^{n \times d}$
Sparse Likelihood Estimator
$\hat{l} \in \mathbb{R}^n$
Sparse Likelihood Vector
Relevant Factors Learning
Decision Structure Learner
$d = \hat{l}^T F \in \mathbb{R}^d$
$\hat{f} = 1^T F \in \mathbb{R}^d$
Maximize $\frac{\hat{f}^T d}{d}$
BCE Loss
Counterfactual Contrastive Learner
Relevant Factors
Supporting Factors
Random Factors
$F \in \mathbb{R}^{n \times d}$
F_n
F_p
Contrastive Loss
Distance between (F, F_p)
Distance between (F, F_n)
Min
Max
Causal Explanation Learning
Causal Explanation Generation
Relevant Factors of F
$F^D_{rel} \in \mathbb{R}^{k_1 \times d}$
Factor-level Causal Explanation
Top-k Dependency Candidate Factors
Causal Dependency Learner
Label Generation
1/0
Causality Digging
$\theta = \{\theta_1, ..., \theta_T\}$
Causal Dependency
Counterfactual Representation Generator
F
$\widetilde{F}_1$
$\widetilde{F}_m$
Mapping
F^c_1
F^c_m
Sample m Times
Encoder
Mean
Std. dev
μ
σ
z
Decoder
$F^c \in \mathbb{R}^{n \times d}$
Counterfactual Factors
F
Trained Factor Embeddings of a Decision

Fig. 3. Framework overview of FCE-UTD. Three modules are included: Input Preparation, Relevant Factors Learning, and Causal Explanation Learning.

- *Input Preparation Module*. This module aims to initialize the embedding for each factor and compose decisions. We take a user decision composed of several factors as input, and the d-dimensional embedding of factors is obtained through the embedding layer.
- *Relevant Factors Learning Module*. This module aims to learn factor embeddings and identify relevant factors. Specifically, We first developed *Self-Projection Attention* mechanism to update the factor embedding matrix $\boldsymbol{F}$ into $\hat{\boldsymbol{F}}$ with the pairwise scalar projection matrix as the attention matrix. Then $\boldsymbol{F}$ and $\hat{\boldsymbol{F}}$ are input to *Sparse Likelihood Estimator* to evaluate the likelihood of each factor to be a relevant factor. Finally, to learn the accurate and robust embeddings of all factors and representations of user decisions, two additional sub-modules are designed with two different objectives: *Decision Structure Learner* aims to predict the user decision based on the embeddings of relevant factors, where the **Binary Cross-Entropy (BCE)** loss function is used with the user decision labels as the supervision. *Counterfactual Contrastive Learner* turns to distinguish the different impacts of relevant and supporting factors on the representation of user decisions, where the contrastive loss function is used with the counterfactual user decision samples as contrastive samples.
- *Causal Explanation Learning Module*. This module is used to mine the true causal explanations with the relevant factors as potential candidates. Specifically, with the representation $\boldsymbol{F}$ of a user decision from the trained Relevant Factors Learning module, we first developed a *Counterfactual Representation Generator* to generate m counterfactual representations. It adds continuous noise on each factor embedding of $\boldsymbol{F}$ with a pre-trained **Variational Auto-Encoder (VAE)** to obtained counterfactual representation $\{\boldsymbol{F}^c_1, \ldots, \boldsymbol{F}^c_m\}$, and maps each noised factor embedding into the real-world factor embedding, which results in the real counterfactual decisions $\{\widetilde{\boldsymbol{F}_1}, \ldots, \widetilde{\boldsymbol{F}_m}\}$. Then, in *Causal Dependency Learner*, a simple causality digging sub-module is designed to infer the causal dependencies of each factor for a user decision by using the sum of the causal dependency scores of each factor to model the possibility of counterfactual decisions. In particular, the counterfactual decisions of the counterfactual representations are produced by the trained Relevant Factors Learning module. Finally, in *Causal Explanation Generator*, we use the intersection of the top factors with

the highest causal dependency and the original relevant factors to generate the causal explanation of this user decision.

Next, we will introduce the two main modules in our framework, i.e., Relevant Factors Learning Module and Causal Explanation Learning Module.

4.1 Relevant Factors Learning Module

A user decision consists of several factors that jointly influence the decision-making process. However, intuitively, it is unlikely that all factors will be crucial for a decision, thus we distinguish between the concepts of relevant and supporting factors based on the degree of influence (Noting that, we denote “relevant factors” as those factors with relatively strong relevance for convenience). Specifically, we learn the representations of all factors, and compute an aggregated relevant embedding for each decision, which is considered as the weighted combination of relevant factor embeddings. Then we preserve decision structures by maximizing the sum of scalar projections of each factor embedding on the aggregated embedding, which can emphasize the impacts of relevant factors and reduce the impacts of supporting factors.

4.1.1 Self-Projection Attention. As described in our previous work [14], a relevant factor should be supported by a lot of other factors that project a large scalar on the relevant factor. To evaluate the probability of each factor becoming a relevant factor, a lightweight self-projection attention is introduced to compute a projected embedding for each factor, which reflects the contribution of other factors to the current factor for learning the likelihood later.

Specifically, given a decision $D = \{f_1, \ldots, f_n\}$ and its factor embedding matrix $\boldsymbol{F} = [\boldsymbol{f}_1, \ldots, \boldsymbol{f_n}]^T$, we first compute the scalar projection of $\boldsymbol{f_j}$ on $\boldsymbol{f_i}$ to obtain the pairwise scalar projection matrix $\boldsymbol{P} \in \mathbb{R}^{n \times n}$,

$$\boldsymbol{P_{ij}} = \boldsymbol{f}_{\boldsymbol{i}}^T \boldsymbol{f_j} / |\boldsymbol{f_i}|. \tag{1}$$

Then we normalize $\boldsymbol{P}$ by applying softmax function to each row,

$$\hat{\boldsymbol{P}}_{\boldsymbol{i}:} = softmax(\boldsymbol{P}_{\boldsymbol{i}:}), i = \{1, \ldots, n\}. \tag{2}$$

Next, for each factor $\boldsymbol{f_i}$, the attention embedding that indicates the sum of the impacts of other factors on it can be formulated as

$$\hat{\boldsymbol{f}}_{\boldsymbol{i}} = \sum_{j=1}^{n} \hat{\boldsymbol{P}}_{\boldsymbol{ij}} \boldsymbol{f_j}, \tag{3}$$

and the attention embedding matrix is $\hat{\boldsymbol{F}} = \hat{\boldsymbol{P}}\boldsymbol{F} \in \mathbb{R}^{n \times d}$.

4.1.2 Sparse Likelihood Estimator. We evaluate the likelihood of each factor being a relevant factor from sparse likelihood perspective. To integrate information about each factor itself and the influence of other factors on it, we concatenate the original embedding of the factor and the projection attention embedding, $\boldsymbol{F} \oplus \hat{\boldsymbol{F}}$, and then input it into **multi-layer perception (MLP)** with Dropout and ReLU activation to obtain a dense likelihood vector $\boldsymbol{l} \in \mathbb{R}^n$,

$$\boldsymbol{l} = MLP(\boldsymbol{F} \oplus \hat{\boldsymbol{F}}). \tag{4}$$

We use three-layer MLP here, and the trainable matrices parameters are $W_1 \in \mathbb{R}^{2d \times d}$, $W_2 \in \mathbb{R}^{d \times d}$, and $W_3 \in \mathbb{R}^{d \times 1}$, respectively. Now, $\boldsymbol{l}$ contains information about itself and other factors and indicates those factors with a large impact (i.e., large scalar projection).

Since $\boldsymbol{l}$ is a dense vector, we expect to sparse it to obtain relevant factors. We use the convenient sparsemax [26] to normalize $\boldsymbol{l}$, which will output sparse probabilities,

$$\hat{\boldsymbol{l}} = sparsemax(\boldsymbol{l}) = \underset{\boldsymbol{p} \in \Delta^{n-1}}{\arg\min} |\boldsymbol{p} - \boldsymbol{l}|, \tag{5}$$

where $\Delta^{n-1} = \{\boldsymbol{p} \in \mathbb{R}^n | \mathbf{1}^T \boldsymbol{p} = 1, \boldsymbol{p} \geq 0\}$ is the n-dimensional probability simplex and sparsity is ensured by Euclidean projection onto Δ^{n-1} [14]. The vector $\hat{\boldsymbol{l}}$ is a sparse normalized vector, which indicates relevant factors and their contribution to the decision. Note that we cannot explicitly specify the number of relevant factors, so we further adopt an L2 regularizer on the unnormalized $\boldsymbol{l}$ before sparsemax to control it flexibly. The larger L2 weight is, the more relevant factors can be identified.

4.1.3 Decision Structure Learner. After obtaining the sparse likelihood vector $\hat{\boldsymbol{l}}$, we can compute the aggregated relevant embedding $\boldsymbol{d}$ of decision D,

$$\boldsymbol{d} = \sum_{i=1}^{n} \hat{\boldsymbol{l}}_i \boldsymbol{f}_i = \hat{\boldsymbol{l}}^T \boldsymbol{F}. \tag{6}$$

Intuitively, this embedding $\boldsymbol{d}$ should preserve the user decision structure as much as possible, i.e., the information in all factors in D. Therefore, we hope to maximize the scalar projection of all factors in D on the aggregated relevant embeddings for each real user decision, i.e.,

$$\max_{\boldsymbol{F}} \boldsymbol{f'}^T \boldsymbol{d}/|\boldsymbol{d}|, \boldsymbol{f'} = \sum_{i=1}^{n} \boldsymbol{f}_i. \tag{7}$$

Noting that the sparse likelihood $\hat{\boldsymbol{l}}$ puts more weight on the relevant factors, this objective will enhance the influence of relevant factors, and reduce the influence of supporting factors.

Specifically, to train representations for all factors based on Equation (7), we need both positive and negative decision instances [45]. As stated in Problem 1, we define historical decisions made by users as positive instances, and the empirical check-in rate $\hat{r}(D)$ is 1 (i.e., label = 1). For each positive decision instance, we generate several negative decision instances by replacing POI-related factors (e.g., replacing the POI with the same category) and define their empirical check-in rate as 0 (i.e., label = 0). We further define $r(D)$ as the predictive check-in rate, obtained by $r(D) = \sigma(\boldsymbol{f'}^T \boldsymbol{d}/|\boldsymbol{d}|)$. Along this line, we should maximize $r(D)$ for positive instances, while minimizing $r(D)$ for negative instances due to the spurious combination of factors. Therefore, we apply a binary cross-entropy loss function that makes the prediction distribution close to the empirical distribution, thus optimizing the factor representations,

$$L_d = \frac{1}{|\mathcal{D} \cup \mathcal{D}^-|} \sum_{D}^{\mathcal{D} \cup \mathcal{D}^-} -[\hat{r}(D) log(r(D)) + (1 - \hat{r}(D)) log(1 - r(D))], \tag{8}$$

where $\mathcal{D}^-$ is the negative decision set. Obviously, the objective maximizes the scalar projection $\boldsymbol{f'}^T \boldsymbol{d}/|\boldsymbol{d}|$ on positive decision instances while minimizes $\boldsymbol{f'}^T \boldsymbol{d}/|\boldsymbol{d}|$ on the negative.

4.1.4 Counterfactual Contrastive Learner. To mitigate the impact of severe data sparsity on the causal analysis of decisions in urban travel scenarios, we construct counterfactual distribution decision samples to learn more accurate and robust factor representations. Briefly, a counterfactual decision sample is first generated by transforming the factor representations $\boldsymbol{F} \in \mathbb{R}^{n \times d}$ of a decision. Then a contrastive learning paradigm is proposed to optimize $\boldsymbol{F}$.

Specifically, inspired by Reference [60], we introduce an inductive bias before describing how to construct counterfactual decision samples: With the description in Section 4.1.3, we can define the relevant and supporting factors based on the sparse likelihood vector $\hat{\boldsymbol{l}}$. We believe that the replacement of relevant factors will have a large impact on the occurrence of a decision, because changes of several factors that are most important will influence the user decision-making process and the original semantics of the decision will change. We define these decision samples with relevant factors replaced as *negative counterfactual samples*, i.e., with a large semantic change compared to original decision samples. On the contrary, if the supporting factors are replaced, then it will have a smaller impact on the occurrence of the decision and its semantics should remain unchanged. Thus, we define the decision samples whose supporting factors are replaced as *positive counterfactual samples*. For convenience, we select k_1 factors with the highest (lowest) likelihood probability according to $\hat{\boldsymbol{l}}$, then replace them randomly from $\mathcal{F}/F$, and regard them as negative (positive) counterfactual samples. Note that F represents all factors that compose the original decision, and k_1 is determined by a hyper-parameter: $k_1 = \tau * n$, where τ is a ratio hyper-parameter to control the replacement number.

To learn a more accurate and robust factor representations, we propose a targeted and effective contrastive learning paradigm. Intuitively, a robust decision representation should rely primarily on its relevant factors rather than supporting factors with minor impacts. Therefore, the positive counterfactual decision sample should be close to the original decision sample in the latent space, while the negative counterfactual decision sample should be far from the original one. Therefore, we perform contrastive triplet learning on original decisions and counterfactual samples (i.e., positive and negative),

$$L_c = max\{d(\boldsymbol{F}, \boldsymbol{F_p}) - d(\boldsymbol{F}, \boldsymbol{F_n}) + \delta, 0\}, \tag{9}$$

where $d(\cdot,\cdot)$ is the Euclidean distance between original factor embeddings $\boldsymbol{F}$ and factor embeddings of counterfactual samples (i,e., $\boldsymbol{F_p}$ and $\boldsymbol{F_n}$), and δ is the margin that widens the gap between $d(\boldsymbol{F}, \boldsymbol{F_p})$ and $d(\boldsymbol{F}, \boldsymbol{F_n})$.

4.1.5 Factors Learning Objectives. We train factor representations under the supervision of two objective L_d and L_c,

$$O = L_d + \gamma L_c, \tag{10}$$

where γ is a parameter to control the contribution of L_c. After training the above objective, we not only obtain accurate and robust factor representations but also can distinguish relevant and supporting factors.

4.2 Causal Explanation Learning Module

In this section, we introduce how to mine the true causal explanations of users' travel decisions after obtaining the relevant factors (i.e., factors with strong relevance to the decision) in detail.

From the previous description, not all relevant factors are true causal factors. Other spurious explanations, referring to the relevance between two variables that appear to be causal but are not actually, may damage the recommendation results for urban traveling. As a result, the model validity and reliability will not be guaranteed [29], which may lead to difficulties in the execution of downstream tasks. Therefore, here, we turn to remove the spurious explanations and find the real causal factors from relevant factors mined by the Relevant Factors Learning Module in Section 4.1. To achieve this, we first generate counterfactual decision representations based on the original decision distribution with a pre-trained Variational Auto-Encoder. Note that the counterfactual decision representations here are different from those generated in Section 4.1.4. Then, for a decision, we learn the causal dependency for each factor with a logistic regression model. Finally, we infer the true causal explanation from the top factors with the highest causal dependency and original relevant factors.

4.2.1 Counterfactual Representation Generator. To obtain the true causal explanation for a decision, we first need to explore influence of factors in the decision more deeply. Inspired by the recent success of counterfactual data augmentation techniques in the field of NLP [66] and recommendation systems [47], we explore the relationship between one decision and various factors from a counterfactual perspective. Formally, the counterfactual inference aims to answer questions related to "*what if*" in our cases: "What would happen if we replace factors in a decision?" Specifically, given factors in a decision and their representations, we replace some factors (e.g., time or location) and obtain extra counterfactual decisions to explore the impact of each factor on this decision. However, an intractable problem is that the number of factors in the urban travel scenario is huge, and the amount of possible counterfactual decisions is very large if each factor is randomly replaced, making the process of training impractical.

Considering that if we replace factors randomly in a decision, then the generated counterfactual decision data will be far from the semantics of original decisions. As previous work [28, 43] suggested, we tend to obtain counterfactual decisions that are similar but different from the original decision, which can maintain a greater degree of semantics. Given the rapid development of generative models, we adopt VAE, which has been widely used in several fields [5, 12] in recent years, to learn the latent distribution of original decision data. Specifically, we first pre-train a VAE-based counterfactual representation generator to generate counterfactual representations based on the original decision D. Briefly, with the trained Relevant Factors Learning Module, we can obtain the factor embedding matrix $\boldsymbol{F}$ for each user decision D, then we input it to the inference network $Q_\theta(z|F)$ (i.e., encoder) to obtain an approximate posterior Gaussian distribution of the latent embeddings z. Afterwards, several embeddings z are sampled from this distribution with different Gaussian noise, and the conditional generation distribution is computed by the generation network $p_\phi(F|z)$ (i.e., decoder). The training objective of the counterfactual representation generator is

$$\begin{aligned} &\max_{\theta,\phi}\{\lambda\mathbb{E}_{z\sim Q_\theta(z|F)}[\ln p_\phi(F|z)] - (1-\lambda)KL[Q_\theta(z|F)\|p(z)]\}, \\ &\qquad\qquad\qquad\qquad \mathbb{E}_{z\sim Q_\theta(z|F)}[\ln p_\phi(F|z)] = MSE(\boldsymbol{F}, \boldsymbol{F^c}), \end{aligned} \tag{11}$$

where $p(z)$ is the prior distribution, and Gaussian distribution is generally used, $\boldsymbol{F^c}$ is the counterfactual representation generated by decoder. The first part (i.e., reconstruction loss) in Equation (11) aims to improve the reconstruction quality of generated representations, and the second part (i.e., Kullback–Leibler divergence) measures the distance between the posterior and prior distribution, which constrains the similarity between the generated representation and the original decision representation, reflecting the generation ability of the counterfactual representation generator. Here λ is a trading-off parameter to balance the above two parts. For simplicity, we denote the counterfactual representation generator as G in next sections.

With pre-trained the VAE model, for each original decision D, we input its factor embeddings $\boldsymbol{F}$ into the VAE encoder and sample m latent embeddings z with different noises, then generate m counterfactual representations $\{\boldsymbol{F^c_1}, \ldots, \boldsymbol{F^c_m}\}$ through VAE decoder. Since factors in generated counterfactual representations are in latent space, so they cannot represent real-world factors now. For interpretability, we compute the cosine similarity between each counterfactual factor and those in $\mathcal{F}/F$ and map it to the nearest real-world factor. Note that we keep the original user and POI identifiers, since we should keep the most essential factors in a decision to maintain the essential semantics. The mapped counterfactual representations $\{\widetilde{\boldsymbol{F_1}}, \ldots, \widetilde{\boldsymbol{F_m}}\}$ are called counterfactual decisions. Specifically, given a decision D, we denote the factors of each counterfactual decision and their embeddings as $\widetilde{F_i}^D$ and $\widetilde{\boldsymbol{F_i}}^D$, respectively.

4.2.2 *Causal Dependency Learner.*

After obtaining counterfactual decisions, we will integrate them with the original decision and learn the probability of each factor becoming a causal factor (i.e., causal dependency).

Specifically, for each original decision D, we first input its counterfactual decisions $\widetilde{F}^D$ into the check-in rate prediction model trained by Equation (10) in Section 4.1 to predict the label for each

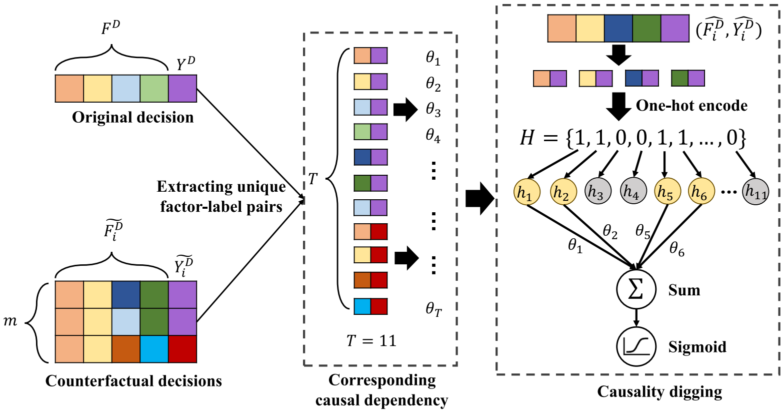


Fig. 4. Illustration of the causality digging sub-module. We first extract unique pairs from the original decision and the corresponding m counterfactual decisions. Then, each pair is assigned a corresponding causal dependency score θ_i. Finally, we utilized the original decision and the counterfactual decisions to infer causal dependencies with a logistic regression.

counterfactual decision, and we denote the obtained label set as $\widetilde{Y}^D = \{\widetilde{Y}_1^D, \ldots, \widetilde{Y}_m^D\}$

$$\widetilde{Y}_i^D = R(\widetilde{\boldsymbol{F}}_i^D), \tag{12}$$

where $R(\cdot)$ stands for our check-in rate prediction model in Section 4.1. Recall that the label indicates whether the decision will happen. Then, for each original decision D, we obtain the set of augmented decision–label pairs:

$$\mathcal{S}^D = \{(\widetilde{F}_i^D, \widetilde{Y}_i^D)\}_{i=1}^m \cup \{(F^D, Y^D)\}, \tag{13}$$

where m is the number of counterfactual decisions, F^D and Y^D are factors that compose D and the corresponding label, respectively. To identify causal factors that constitute the true causal explanation from relevant factors in a decision, we define a trainable causal dependency θ^D to model the causality of each factor. For each decision D, we will learn causal dependencies through the proposed causality digging model after obtaining the augmented decision–label set $\mathcal{S}^D$. For convenience, we define each pair in $\mathcal{S}^D$ as $(\widehat{F}_i^D, \widehat{Y}_i^D)$, which can denote pairs in the original decision or counterfactual decisions.

Inspired by References [3, 57], we tend to learn causal dependencies between $\widehat{Y}_i^D$ and each factor $\widehat{f}_{ij}^D$ that composes $\widehat{F}_i^D$ with likelihood estimation, where $j = 1, \ldots, n$. Then, we infer the causal explanation of the decision with causal dependencies. The idea is further illustrated in Figure 4. Specifically, given a decision D and its decision–label pair set $\mathcal{S}^D$, we first extract unique factor–label pairs $(\widehat{f}_{ij}^D, \widehat{Y}_i^D)$ from all decision–label pairs and denote the number of unique pairs as T. Afterwards, we denote $\theta^D = \{\theta_1^D, \ldots, \theta_T^D\}$ as the trainable causal dependency of the corresponding T unique pairs. Note that the reason for extracting unique factor–label pairs is that if the same pairs appear in different decisions (i.e., original and counterfactual decisions), then they represent the same causal dependency. For simplicity, we hide the superscript D of θ. To infer causal dependencies, we propose to use the sum of the causal dependency scores of each factor in $\widehat{F}_i^D$ to model the possibility of the current decision with logistic regression. And the frequency of input factors in augmented decisions will determine the score of causal dependencies.

In particular, for unique factor–label pairs extracted from $\widehat{F}_i^D$ with corresponding label $\widehat{Y}_i^D$, we first represent them with one-hot encoding and denote them as $H = \{0, 1\}^T$. And we model the possibility of $\widehat{Y}_i^D$ as

$$P(\widehat{Y}_i^D|\widehat{F}_i^D) = \sigma(\boldsymbol{\theta}^T H), \tag{14}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function. The possibility of each decision–label pair in $\mathcal{S}^D$ should be close to 1 from the maximum likelihood perspective, so we infer causal dependencies by maximizing their occurrence probability.

4.2.3 Causal Explanation Generator. After training the logistic regression model in Equation (14), we can distinguish false and true causal factors for an original decision and generate causal explanations through the learned causal dependencies. Specifically, we choose factors with the highest causal dependency θ and generate causal explanations by several steps:

- First, we pick out factor–label pairs in $\mathcal{S}^D$ that have the same label as the original decision (i.e., $\widehat{Y}_i^D = Y^D = 1$) and sort these pairs according to their causal dependencies in descending order. Note that these factor–label pairs include those in the original decision.
- Second, we choose the top-k_2 pairs and extract their factor parts $C = \{\widehat{f}_1, \ldots, \widehat{f}_{k_2}\}$.
- Third, we take the intersection of C and the set of relevant factors F_{rel}^D of the original decision, and denote them as $\mathcal{E}$. For convenience, we set the number of relevant factors as k_1, similar to that in Section 4.1.4. If there are one or more factors in $\mathcal{E}$, then we regard them as causal factors of this original decision D, and we will construct $\mathcal{E} \Rightarrow D$ as the causal explanation of D.

Note that we set $k_2 = 1$ by default. And we treat those factors in F_{rel}^D, which do not constitute causal explanations as false causal factors, which may cause spurious explanations. The model fidelity [61] will be reported to show what percentage of decisions can be explained by our method and further compared to the **association rule mining model (AR)** [33].

In summary, the whole pipeline of our FCE-UTD framework can be found in Algorithm 1.

5 EXPERIMENT

In this section, we conduct extensive experiments on third real-world datasets to evaluate our FCE-UTD framework. Specifically, our experiments aim to answer the following questions:

- **Q1:** How about the effectiveness of FCE-UTD on check-in rate prediction task and the coverage of explanation for user decisions? (Section 5.2)
- **Q2:** How about the effectiveness of each components in FCE-UTD? (Sections 5.3–5.4)
- **Q3:** How do learned causal factor representations perform in downstream recommendation tasks on both regular and **out-of-distribution (OOD)** datasets? (Section 5.5)
- **Q4:** How about the quality of generated explanations and whether our study can provide explanations for user travel decisions? (Sections 5.6 and 5.7)

5.1 Experimental Setups

Datasets. To demonstrate the effectiveness and generalization of our FCE-UTD model, we not only apply it in POI travel scenarios but also in traditional recommendation scenarios. Specifically, we use two POI recommendation datasets, i.e., **Shanghai (SH)** and **New York (NY)** and a movie recommendation dataset, i.e., **movielens 1m (ml1m)**. In particular, we consider the rating behavior in ml1m as check-in behavior for exhibiting the model generalization ability of FCE-UTD.

- SH was produced by a third-party map service platform from Shanghai, which contains user map query and POI check-in records. Similar to the dataset used in Reference [14], each query consists of an anonymous user identifier, a time stamp, a location and some queried POIs. For each query, we constructed a positive decision instance if the user visited at least one of queried POIs in the following three days, and constructed negative decision instances

ALGORITHM 1: Factor-level Causal Explanation For User Travel Decisions

```
Input: user decision set $\mathcal{D}$, the negative decision set $\mathcal{D}^-$, factor lookup table $E$,
check-in rate prediction model $R$, counterfactual representation generator $G$,
counterfactual sample times $m$, causal dependency $\boldsymbol{\theta}$;
Output: causal explanation $\mathcal{E} = \{f_i'\}$;
Initialize $E$, $P$, $G$, $R$;

### Relevant Factors Learning Module
repeat
    for $D = \{f_1, \ldots, f_n\} \in \mathcal{D} \cup \mathcal{D}^-$ do
        Extract the factor embeddings $\boldsymbol{F} = [\boldsymbol{f_1}, \ldots, \boldsymbol{f_n}]$ of $D$ from factor lookup table $E$;
        Train $E$ and $R$ using $\boldsymbol{F}$ and the corresponding label $Y^D$ (i.e., empirical check-in rate $\hat{r}(D)$) by
          Equations (1)–(10);
    end
until $E$ and $R$ converge;
Obtain the sparse likelihood vector and relevant factors $F_{rel}^D$ for each decision $D$;

### Causal Explanation Learning Module
Pre-train $G$ using the representations $\boldsymbol{F}$ of each decision in $\mathcal{D}$ by Equation (11);
for $D \in \mathcal{D}$ do
    Sample $m$ counterfactual representations $\{\boldsymbol{F_1^c, \ldots, F_m^c}\}$ with $G$;
    for $i$=1 to $m$ do
        Map factors in $\boldsymbol{F_i^c}$ onto real-world factors to obtain counterfactual decisions $\widetilde{\boldsymbol{F}}_i^D$;
    end
    Predict the label for all counterfactual decisions and obtain $\widetilde{Y}^D = \{\widetilde{Y}_1^D, \ldots, \widetilde{Y}_m^D\}$;
    Train causal dependency $\boldsymbol{\theta}$ with original and counterfactual decisions;
    Identify causal factors based on the factors with highest $\theta$ and the relevant factors $F_{rel}^D$;
    Generate the causal explanation $\mathcal{E}$ with causal factors;
end
```

with those have not been visited. We filtered a query if no queried POIs were visited. Note that we computed the POI popularity based on the frequency of visits to POIs.

- NYC was produced based on the public Foursquare check-in dataset, which contains POIs and user check-in records. Each check-in was considered as a positive decision instance, and we constructed the negative by replacing POIs with those of the same categories. The POI popularity was computed in the same way for SH dataset. We did not consider distance factors due to unknowable decision location.
- ml1m was produced based on MovieLens 1M dataset and contains records of user rating behavior. We constructed user-related factors based on user's age, gender, occupation, and the most frequently watched movie genres. Furthermore, movie-related factors include genres and popularity, which is computed based on the frequency of being rated and processed in a similar way as POI popularity. Finally, each rating behavior was regarded as a positive decision instance, and we generated negative instances by replacing movies randomly.

Note that we applied the 10-core setting to the above datasets. The data statistics are listed in Table 3. Meanwhile, we divided the training/validation/test sets by the proportion of 7:1:2 for each dataset.

Baselines. To evaluate the effectiveness, we compared our FCE-UTD model in terms of representation learning performance with the following baselines:

Table 3. Statistics of SH, NYC, and Ml1m

Description	SH	NYC	ml1m
time spanning	01/07/18~ 30/09/18	03/04/12~ 14/02/13	01/1996~ 03/2009
# of users	28877	1019	1000
# of items	20081	5086	2568
sparsity	99.87%	97.99%	98.99%
# of positive $\mathcal{D}$	72888	103584	26078
# of negative $\mathcal{D}^-$	437045	517920	130390
# of factors per D/D^-	17	13	14

- *BPR* [37], a classic pairwise learning framework for implicit feedback data. Specifically, we employed matrix factorization as the internal predictor.
- *Learnsuc* [45], which represented behavior records as multi-type itemsets and learned the success of behaviors by preserving itemset structures.
- *SVDGCN* [34] replaced neighborhood aggregation with a truncated SVD, which only exploits K-largest singular values and vectors for downstream tasks.
- *UltraGCN* [25], a GNN-based method that skipped explicit message passing and directly approximated the limit of infinite message passing layers to learn the node representations.
- *HCCF* [52], a self-supervised learning framework that jointly captured local and global collaborative signals with hypergraph-enhanced contrastive learning.
- *DICE* [64] extracted cause-specific data and train different embeddings with them to achieve disentanglement between interest and conformity.
- *UKGC* [21] separated geographical and functional attributes of POIs through a urban knowledge graph, and introduced counterfactual inference to alleviate the geographical bias in POI recommendation. Note that we adapted the graph construction method to the ml1m dataset according to the genres and other context.
- *PROUD* [14] can be seen as a variant of the relevant factors learning part in FCE-UTD that learned factor embeddings without the counterfactual contrative learning paradigm.
- *AR* [33], a post hoc explanation model that aimed to discovering association rules of from all users' interactions. Here we used it to generate explanations and compared it with our method in terms of model fidelity.

Metrics. We adopted Precision (Pre), Recall, F1, and AUC to evaluate the performance about check-in rate prediction task, and we used model fidelity to evaluate our causal explanation framework. **Implementation.** We implemented our model with PyTorch. We used the Adam optimizer with the learning rate $lr = 0.01/0.005/0.001$ for SH/NYC/ml1m and set batch size $B = 512$ to train FCE-UTD. The number d of dimensions was fixed to 64 for all methods and set $\tau = 0.5, \gamma = 1.0, \lambda = 0.1, \delta = 1.2$. The default number of counterfactual decisions is $m = 100$. We set the dropout rate in Equation (4) as 0.2. For the VAE-based counterfactual representation generator, both the encoder and decoder are two-layer MLP. We set the latent dimensions of encoder and decoder as 32, and the dimension of z as 48. Besides, we employed an early stopping if the F1 on validation set did not increase in 5/10/5 epochs for SH/NYC/ml1m.

For the AR model, we ranked based on support value by default to generate factor-level explanations and referred to the settings in Reference [33] to set the optimal parameters for all datasets: support = 0.001 for finding the frequent itemset, min threshold = 0.001 for extracting association rules and length = 2 for filtering. Finally, we accepted the all extracted rules based on the support value as explanations.

Table 4. Accuracy Evaluation on Check-in Rate Prediction

Method	**SH**				**NYC**				**ml1m**			
	Pre	Recall	F1	AUC	Pre	Recall	F1	AUC	Pre	Recall	F1	AUC
BPR	0.4585	0.5658	0.4614	0.8009	0.7418	0.6619	0.6871	0.8341	0.7767	0.7899	0.7772	0.9043
Learnsuc	0.3488	0.3409	0.3448	0.6767	0.2330	0.5616	0.3293	0.6082	0.5378	0.5832	0.5542	0.7785
SVDGCN	0.4224	0.5374	0.4396	0.8138	0.7249	0.6911	0.7009	0.8750	0.7127	0.6707	0.6850	0.8770
UltraGCN	0.7751	0.6480	0.7011	0.8919	0.3123	0.6517	0.4173	0.7041	0.5156	0.6120	0.5502	0.8204
DICE	0.7418	0.6944	0.6994	0.8947	0.7287	0.6555	0.6730	0.8676	0.7149	0.7393	0.7165	0.8964
UKGC	0.5679	0.6401	0.5748	0.8575	0.6078	0.5043	0.4510	0.6617	0.5289	0.6943	0.5806	0.7784
HCCF	0.5597	0.6506	0.5833	0.8294	0.4933	0.6412	0.5241	0.8513	0.8045	0.8098	0.8013	0.9200
PROUD	0.7879	0.7979	0.7929	0.9647	0.7560	0.6701	0.7104	0.9220	0.9333	0.9022	0.9175	0.9822
FCE-UTD	**0.8527***	**0.8091***	**0.8304***	**0.9723***	**0.7941***	**0.7194***	**0.7549***	**0.9277***	**0.9492***	**0.9126***	**0.9306***	**0.9841***
p-value	9.81e-5	2.35e-2	1.43e-5	3.62e-2	2.15e-3	2.96e-3	5.36e-4	3.38e-4	6.40e-4	8.62e-3	5.77e-4	1.48e-2

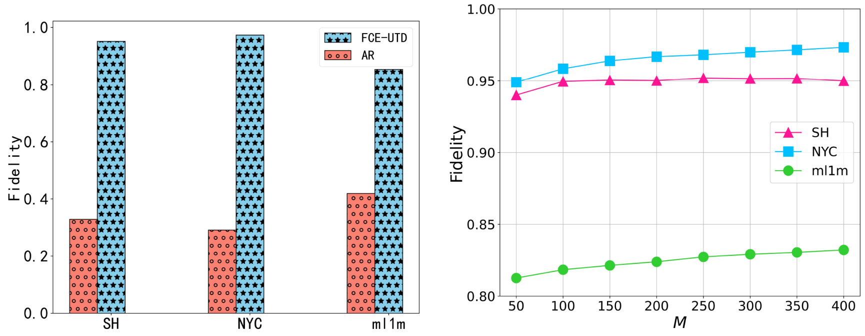
The best result is highlighted in bold.

5.2 Experimental Results

5.2.1 Check-in Rate Prediction. We first evaluate the performance of check-in rate prediction task for distinguishing positive and negative user decision instances, which can reflect the effectiveness of FCE-UTD in preserving decision structures as well as factor embedding learning. The results are reported in Table 4.

Based on the results, we can find that FCE-UTD significantly outperforms all baselines in all three datasets and four metrics, especially the Precision, reflecting the superiority of our model in determining true positive decision instances without sacrificing Recall, which demonstrates the effectiveness and generalization of our framework. Moreover, we also witness the following interesting findings. First, UKGC, HCCF, UltraGCN, and SVDGCN are GNN based and related to collaborative filtering. The first two methods achieve similar performance on SH due to the serious data sparsity and they both propagate information over multiple graphs, enabling them to learn sufficient information. Notably, HCCF outperforms better than UKGC in NYC and ml1m, because UKGC is influenced by the degree of contextual richness, while HCCF effectively integrates explicit and implicit user–item relationship to alleviates data sparsity and learns more accurate representations through a contrastive learning paradigm. UltraGCN performs better than the above two on SH, mostly because it respectively filters uninformative user–item and item–item relationships, which avoids introducing too much noise in sparse dataset. However, UltraGCN can only exploit the first-order neighborhood and loses the ability to capture high-order collaborative signals, making it difficult to take full advantage of its strengths in small but less sparse datasets (i.e., NYC and ml1m). In addition, SVDGCN performs well on NYC and ml1m, attributed to the ability of its truncated SVD to extract effective features. However, its performance on SH is unsatisfactory due to the significant impact of data sparsity. Second, DICE performs well, since it learns representations whose conformity are eliminated and can better learn users' real preferences. Third, Learnsuc does not work well, because positive and negative decision instances are only partially different in POI-related factors, which makes the behavior modeling difficult. After that, BPR performs better in NYC and ml1m than SH, since the smaller data sparsity makes the task easier. Finally, the comparison of FCE-UTD and PROUD can be seen as a part of *ablation experiments*, which indicates the effectiveness of our counterfactual contrastive learning module in factor representation learning. In addition, We also conduct one-sample t-tests of FCE-UTD with the strongest baseline PROUD, and p-value < 0.05 indicates that the improvements of FCE-UTD over the strongest baseline PROUD are statistically significant.

5.2.2 Model Fidelity. The purpose of explainable models is to generate explanations for most decisions, so we report the model fidelity in Figure 5(a) to show what percentage of decisions can



(a) Model fidelity.

(b) Impact of hyper-parameter m.

Fig. 5. The results of model fidelity and impact of hyper-parameter m.

be explained with our FCE-UTD framework. Specifically, we set the number of causal factors that constitute explanation as 1 (i.e., $k_2 = 1$). We can clearly see that FCE-UTD reaches the highest fidelity in all datasets, which shows the superiority of counterfactual representation generator in our causal explanation framework. We can generate many counterfactual decisions and fully explore the influence between factors and decisions to extract more potential causal relations for generating causal explanations. Besides, AR gets the worst performance, which exposes AR cannot mine enough association rules with insufficient user–item interactions. If a user visited a POI or rated a movie only once, then it will be difficult to match global association rules with this decision, so explanations can only be generated for a few decisions.

5.3 Ablation Study

To further validate the effectiveness of the various components of FCE-UTD, we design several simplified variants of FCE-UTD for the Relevant Factors Learning and Causal Explanation Learning modules and conduct experiments on the check-in rate prediction task and model fidelity.

- *FCE-UTD-SPA:* This method is a variant of FCE-UTD that removes the *Self-Projection Attention* part. Specifically, we replace $\hat{\boldsymbol{F}}$ with $\boldsymbol{F}$ in Equation (4) to indicate that only the information of the factor itself is considered, regardless of contributions of other factors.
- *FCE-UTD-SLE:* This method is a variant of FCE-UTD that removes the *Sparse Likelihood Estimator* part. Specifically, we replace the sparsemax with softmax function and remove the corresponding L2 regularization loss on the unnormalized $\boldsymbol{l}$.
- *FCE-UTD-RSS:* This method is a variant of FCE-UTD that removes both the *Self-Projection Attention* and *Sparse Likelihood Estimator* parts.
- *FCE-UTD-DSC:* This method is a variant of FCE-UTD that removes the *Decision Structure Learner* in Section 4.1.3 and the corresponding objective L_d in Equation (10).
- *FCE-UTD-CCL:* This method is a variant of FCE-UTD that removes the *Counterfactual Contrastive Learner* in Section 4.1.4 and the corresponding objective L_c in Equation (10).
- *FCE-UTD-CRG:* This method is a variant of FCE-UTD that uses a fixed standard deviation and mean to replace those learned by the encoder in VAE.

The results of the ablation study are shown in Tables 5 and 6. The best results are highlighted in bold. For the check-in rate prediction task, we have the following findings. First, FCE-UTD-SPA and FCE-UTD-SLE outperform FCE-UTD-RSS in all three datasets and on four metrics, indicating that both the self-projection attention mechanism and sparse likelihood estimator can help the model better learn the relevance of each factor in a decision and further preserve decision structures. Specifically, the self-projection attention can effectively learn the contribution matrix between

Table 5. The Results of Ablation Study on Check-in Rate Prediction

Method	**SH**				**NYC**				**ml1m**			
	Pre	Recall	F1	AUC	Pre	Recall	F1	AUC	Pre	Recall	F1	AUC
FCE-UTD-SPA	0.8467	0.7898	0.8173	0.9690	0.7780	0.6906	0.7317	0.9087	0.9166	0.8900	0.9031	0.9809
FCE-UTD-SLE	0.8241	0.7931	0.8083	0.9679	0.7882	0.6931	0.7376	0.9269	0.9066	0.8408	0.8725	0.9563
FCE-UTD-RSS	0.6522	0.7014	0.6759	0.9314	0.4607	0.7064	0.5577	0.8889	0.9079	0.7576	0.8260	0.9303
FCE-UTD-DSC	0.1422	0.9995	0.2490	0.4947	0.2089	0.4902	0.2930	0.6858	0.1799	0.7418	0.2896	0.5531
FCE-UTD-CCL	0.7879	0.7979	0.7929	0.9647	0.7560	0.6701	0.7104	0.9220	0.9333	0.9022	0.9175	0.9822
FCE-UTD	**0.8527**	**0.8091**	**0.8304**	**0.9723**	**0.7941**	**0.7194**	**0.7549**	**0.9277**	**0.9492**	**0.9126**	**0.9306**	**0.9841**

Table 6. The Results of Ablation Study on Model Fidelity

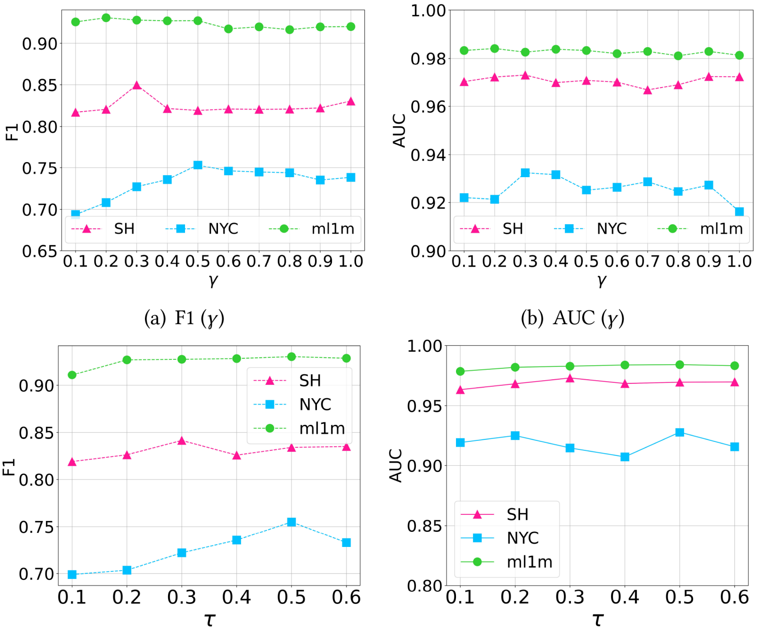
Method	**SH**	**NYC**	**ml1m**
FCE-UTD-CRG	0.3792	0.1931	0.3241
FCE-UTD-SPA	0.4582	0.6872	0.7123
FCE-UTD-SLE	0.7332	0.6135	0.7130
FCE-UTD-CCL	0.8678	0.7737	0.6470
FCE-UTD	**0.9514**	**0.9734**	**0.8536**

factors, while sparse likelihood estimation can effectively enhance the impacts of relevant factors and reduce the impacts of supporting factors, thus better preserving decision structures. Second, the FCE-UTD-DSC performs extremely badly, demonstrating that the model does not work at all without the objective in the decision structure learner. In addition, FCE-UTD performs better than FCE-UTD-CCL, indicating that the counterfactual contrastive learner is able to learn more accurate and robust factor representations. Finally, FCE-UTD consistently outperforms all variants, which suggests the significance of each component in the Relevant Factors Learning module.

For the performance of each variant in terms of model fidelity, we witness the following findings. Note that we omit FCE-UTD-RSS and FCE-UTD-DSC, since they are ineffective to learn the relevance of each factor for further generating causal explanations. First, FCE-UTD-CRG performs worst among all variants, owing to the fixed standard deviation and mean cannot generate specific couterfactual representations for different user decisions, resulting in the failure of causality mining. Second, both FCE-UTD-SPA and FCE-UTD-SLE do not work well, which verifies the importance of considering mutual contributions between factors when identifying relevant factors, and the necessity of using the sparsemax function to emphasize the influence of relevant factors, respectively. Moreover, the comparison of FCE-UTD and FCE-UTD-CCL demonstrates that the counterfactual transformation in counterfactual contrastive learner can help to learn more generative factor representations. Finally, FCE-UTD performs best, illustrating that each part of FCE-UTD is crucial for the generation of explanations.

5.4 Impacts of Hyper-Parameters

Here, we first evaluate the impacts of hyper-parameters γ and τ in Section 4.1.4 with F1 and AUC, which consider both Precision and Recall metrics and provide a more comprehensive evaluation. Specifically, γ controls the contribution of counterfactual contrastive objective, Figure 6(a) and (b) show the results by varying γ. The best result is achieved with $\gamma = 0.3$ for SH, $\gamma = 0.4/0.5$ for NYC and $\gamma = 0.2 \sim 0.5$ for ml1m. The too-small γ will limit the effectiveness of counterfactual contrastive learning, while too-large γ will make the ability to preserve decision structures reduced and introduce more noise.



(a) F1 (γ) (b) AUC (γ)

(c) F1 (τ) (d) AUC (τ)

Fig. 6. Impact of hyper-parameter γ, τ in terms of F1 and AUC.

The results of replacement ratio τ for constructing counterfactual samples in counterfactual contrastive learner are reported in Figure 6(c) and (d). The best results are reached with $\tau = 0.3/0.5/0.5$ for SH, NYC, and ml1m, respectively. The effect of counterfactual contrastive objective will be affected when τ is small, while the large τ will introduce too much noise that comes from the random replacement of factors, and influence the accuracy of representation learning.

In addition, we also evaluate the number of counterfactual decisions m with the model fidelity as the metric. As shown in Figure 5(b), when m increases, model fidelity increases as well. When m is small, the random noise generated by the counterfactual representation generator increase the probability that the model will extract factors other than the original ones as explanations, thus reducing model fidelity. But when m becomes larger, rich counterfactual decisions can help the model better explore the causality between each factor and the decision. Meanwhile, as m increases, counterfactual decisions containing the original factors also increases, which can statistically offset such noise.

5.5 Application–Recommendation Task

To demonstrate the mined factor-level causal explanations can assist downstream tasks, we conduct additional experiments of recommendation task on all three original and corresponding OOD datasets. To construct OOD datasets, we partition all decisions based on the categories of visited POIs for SH and NYC and based on user occupations for ml1m. Moreover, we separate training and test sets according to the partition, aiming to ensure a different distribution between training and test sets. Note that we conduct the following experiments with the optimal parameters in previous experiments. Specifically, we first mine causal factors for each decision in test set. After that, for each user and item pair (u_i, v_j) in the test set that needs to be scored and ranked, we extract identifier factors of the user and the item and then combine these factors with mined causal factors by

Table 7. Recommendation Performance on Original and OOD SH Datasets

Method	**SH**				**OOD SH**			
	Rec@1	Rec@5	N@1	N@5	Rec@1	Rec@5	N@1	N@5
BPR	0.0019	0.0081	0.0035	0.0056	0.0013	0.0020	0.0024	0.0019
WRMF	0.0026	0.0075	0.0055	0.0060	0.0015	0.0042	0.0038	0.0039
NGCF	0.0025	0.0042	0.0041	0.0060	0.0013	0.0016	0.0029	0.0037
UltraGCN	0.0024	0.0091	0.0050	0.0068	0.0008	0.0026	0.0048	0.0034
SVDGCN	0.0024	0.0048	0.0041	0.0043	0.0014	0.0021	0.0022	0.0020
FCE-UTD-E	0.0029	0.0045	0.0044	0.0041	0.0078	0.0078	0.0124	0.0087
FCE-UTD	**0.0111**	**0.0117**	**0.0189**	**0.0131**	**0.0093**	**0.0094**	**0.0157**	**0.0109**

Rec@k means Recall@k and N@k means NDCG@k.
The best result is highlighted in bold.

Table 8. Recommendation Performance on Original and OOD NYC Datasets

Method	**NYC**				**OOD NYC**			
	Rec@1	Rec@5	N@1	N@5	Rec@1	Rec@5	N@1	N@5
BPR	0.0019	0.0073	0.0130	0.0098	0.0009	0.0053	0.0060	0.0080
WRMF	0.0046	0.0180	0.0307	0.0272	0.0020	0.0063	0.0100	0.0092
NGCF	0.0040	0.0124	0.0287	0.0559	0.0023	0.0066	0.0100	0.0212
UltraGCN	0.0045	0.0155	0.0337	0.0283	0.0015	0.0050	0.0130	0.0092
SVDGCN	0.0078	0.0170	0.0559	0.0351	0.0024	0.0053	0.0128	0.0098
FCE-UTD-E	0.0007	0.0048	0.0040	0.0068	0.0004	0.0018	0.0040	0.0034
FCE-UTD	**0.0169**	**0.0448**	**0.0990**	**0.0665**	**0.0136**	**0.0269**	**0.0702**	**0.0406**

a simple summation to form user and item representations. Note that we only use causal factors that are user related or item related. Finally, we calculate the score for (u_i, v_j) pair by taking the inner product of their representations,

$$x_{ij} = \boldsymbol{u}_i \cdot \boldsymbol{v}_j, \boldsymbol{u}_i = \boldsymbol{u}_{ID} + \sum^{F_{u_i}} \boldsymbol{f}, \boldsymbol{v}_j = \boldsymbol{v}_{ID} + \sum^{F_{v_j}} \boldsymbol{f}, \tag{15}$$

where F_{u_i} and F_{v_j} are sets of user-related and item-related causal factors. Finally, we adopt the Recall@k and NDCG@k metrics to test the recommendation performance, and results on original and OOD datasets are shown in Tables 7–9, respectively. Note that we generate only one causal factor for each decision by default, and the FCE-UTD-E is a variant that replaces causal factors with relevant factors. Additionally, the WRMF [15] and NGCF [46] in results are two classical recommendation models. Specifically, WRMF is a pointwise latent factor model that distinguishes user observed and unobserved check-in data with different confidence values. NGCF is a graph-based framework for collaborative filtering that adopts three GNN layers on the user–item graph to refine user and item representations via at most three-hop neighbors' information.

As shown in Tables 7–9, for small-scale original datasets (i.e., NYC and ml1m), SVDGCN can achieve better results, which shows that the truncated SVD can extract effective features and reduce the noise from vectors with small singular values. Meanwhile, NGCF performs better in NDCG@5. This is mainly because NGCF is capable of effectively modeling higher-order

Table 9. Recommendation Performance on Original and OOD ml1m Datasets

Method	**ml1m**				**OOD ml1m**			
	Rec@1	Rec@5	N@1	N@5	Rec@1	Rec@5	N@1	N@5
BPR	0.0271	0.0892	0.1296	0.1051	0.0317	0.0958	0.1498	0.1154
WRMF	0.0383	0.1291	0.1822	0.1511	0.0374	0.1200	0.1761	0.1428
NGCF	0.0345	0.1121	0.1427	0.2524	0.0405	0.1223	0.1508	0.2147
UltraGCN	0.0451	0.1398	0.2176	0.1690	0.0412	0.1246	0.1387	0.1531
SVDGCN	0.0465	0.1544	0.2150	0.1838	0.0460	0.1340	0.1566	0.1594
FCE-UTD-E	0.0235	0.0458	0.1143	0.0647	0.0312	0.0447	0.0881	0.0656
FCE-UTD	**0.1021**	**0.1859**	**0.4858**	**0.2693**	**0.1115**	**0.1823**	**0.3727**	**0.2958**

connectivity in small user–item interaction graph by stacking multiple embedding propagation layers, leading to a good ranking ability over a longer range. Additionally, in large-scale dataset SH, UltraGCN has achieved good results due to its simplified model structure that can filter out user–item relationships with limited information. In OOD SH, WRMF proves to be more effective than other more complex models in the case of imbalanced data distribution, possibly because its simple structure helps avoid overfitting. For OOD NYC and OOD ml1m, NGCF gets better results, especially on NDCG@5, which is mainly attributed to the strong ranking ability of the learned higher-order connectivity. Finally, our method achieves the best performance on both original and OOD datasets, which demonstrates the superiority of the mined causal factors. Compared with using representations of relevant factors (i.e., FCE-UTD-E), our method utilizes causal factors, which can effectively avoid misleading some spurious relationships and enable user and item representations to contain some causal orientation. Moreover, experiments on ml1m and OOD ml1m further validate the generalization of our method.

5.6 Average Causal Effect

We evaluate the quality of generated causal explanations with **Average Causal Effect (ACE)** [27, 57] in the absence of the ground-truth explanations. The ACE can be used to measure the effect from a particular operation, and we first give the definition of Average Causal Effect according to Reference [27].

Definition 5 (Average Causal Effect). The Average Causal Effect of a binary random variable x (treatment) on another random variable y (outcome) is defined as

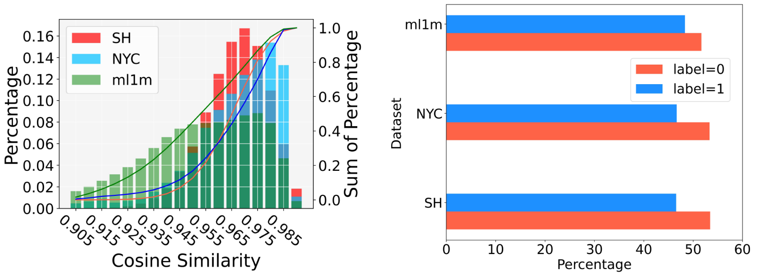
$$ACE = \mathbb{E}[y|do(x=1)] - \mathbb{E}[y|do(x=0)], \tag{16}$$

where $do(\cdot)$ denotes an external intervention that forces a variable to take a specific value. In our case, we use ACE to evaluate the causality inside our check-in rate prediction model. Specifically, to compute ACE, we first keep the number of causal factors that constitute explanation as 1(i.e., $k_2 = 1$), which means that each decision explanation $\mathcal{E}$ consists of one factor. Then, given the causal explanation of decision D, i.e., $\mathcal{E} \Rightarrow D$, and m corresponding counterfactual decisions $\{(\widetilde{F}_i^D, \widetilde{Y}_i^D)\}_{i=1}^m$ of D, if $\mathcal{E}$ belongs to $\widetilde{F}_i^D$, then we regard it as setting x to 1 (i.e., $do(x=1)$), and 0 otherwise. Recall that $\widetilde{F}_i^D$ is obtained by counterfactual representation generator in Section 4.2.1, which cannot be observed in the original decision. Moreover, we also define y as a binary random variable, where $y = 1$ if the $\widetilde{Y}_i^D = Y^D = 1$ occurs, and 0 otherwise. Finally, we can compute average

Table 10. The Results of Average Causal Effect

Method	SH	NYC	ml1m
FCE-UTD-CRG	0.2803	0.3312	0.2367
FCE-UTD	**0.4583**	**0.4770**	**0.3878**

The best result is highlighted in bold.



(a) Cosine similarity of mapping.

(b) Label distribution of counterfactual decisions.

Fig. 7. Cosine similarity of mapping and the label distribution of counterfactual decisions.

ACE on all generated explanations,

$$\mathbb{E}[y|do(x=1)] = P(y=1|do(x=1)) = \frac{\#Pairs((f \in \widetilde{F}_i^D) \wedge (\widetilde{Y}_i^D = 1))}{\#Pairs(f \in \widetilde{F}_i^D)}, \tag{17}$$

$$\mathbb{E}[y|do(x=0)] = P(y=1|do(x=0)) = \frac{\#Pairs((f \notin \widetilde{F}_i^D) \wedge (\widetilde{Y}_i^D = 1))}{\#Pairs(f \notin \widetilde{F}_i^D)}, \tag{18}$$

where f is the only causal factor in $\mathcal{E}$. To verify the effectiveness of our counterfactual representation generator for generating accurate causal explanations, we compare FCE-UTD with FCE-UTD-CRG, which uses fixed standard deviation and mean. The results in three datasets are shown in Table 10. Since ACE values only apply to models that are relevant for modeling causality, we do not report ACE values for the (AR).

We can find that FCE-UTD achieves the best results on all datasets, which indicates our counterfactual representation generator can generate different counterfactual factors for each user decision and construct counterfactual representations that conform to the inherent causal logic of original users' decisions. This helps the causal dependency learner to mine accurate causal explanations. Moreover, FCE-UTD-CRG performs worse, mainly because it uses fixed standard deviation and mean, which fails to yield effective decision-specific counterfactual representations and hinders the learning process of causal dependency learner.

5.7 Case Studies

5.7.1 Quality of Counterfactual Decisions. As stated in Section 4.2.1, to generate causal explanations for user travel decisions, we should first ensure the quality and generation ability of the decisions generated by the counterfactual representation generator. Since the generated counterfactual representations need to be mapped to real-world factors, one of the important premise is the accuracy of mapping (i.e., the cosine similarity between the counterfactual and real-world factors).

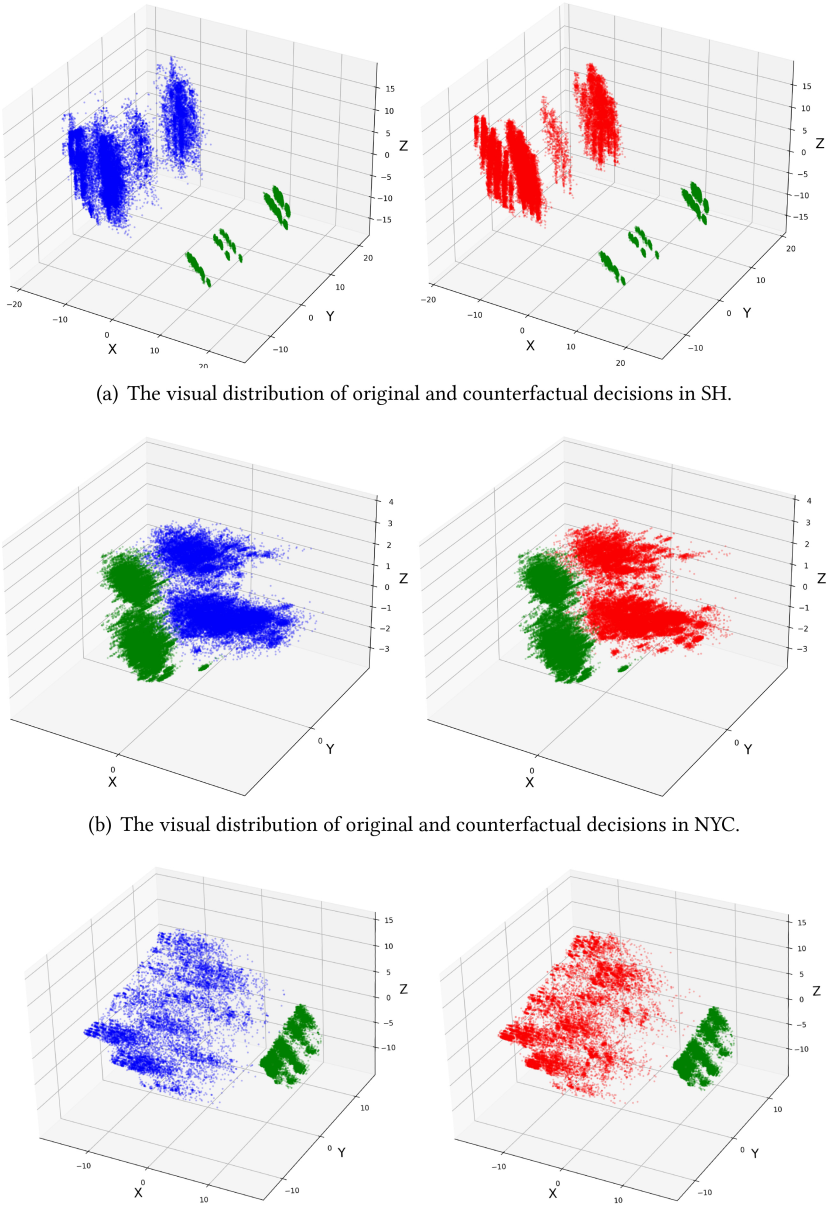
Note that a high similarity means the generated counterfactual representation is close to the real space, while a low similarity means it is far from the real space. We show the mapping of generated counterfactual representations in Figure 7(a). For each counterfactual decision, we compute the average cosine similarity for factors in the mapping process. As shown in Figure 7(a), the decisions generated by counterfactual representation generator have good mapping performance across all datasets, which indirectly demonstrates the high quality of counterfactual representations. In SH and NYC datasets, the matching similarity is concentrated between 0.945 and 0.985, while the similarity in ml1m is concentrated between 0.925 and 0.980. We can notice that there are some decisions with low mapping similarity in NYC and ml1m. This is because the number of factors that comprise decisions in the two datasets is less than those of SH, thus increasing the variance.

Furthermore, since we need to generate different decisions from the original to explore the influence of factors on decisions, the generation ability of counterfactual decisions is also important. As described in Section 4.2.2, to mine the potential causality of each factor in a decision, we compute labels of the counterfactual decisions by means of the trained check-in rate prediction model $R(\cdot)$. To verify the intermediate process of counterfactual decision generation, we present the label distribution of counterfactual decisions, the result is shown in Figure 7(b). It can be seen that the distribution of positive and negative labels is relatively balanced, indicating that representations generated through the counterfactual representation generator will not deviate significantly from the original decision and introduce excessive noise, which facilitates the generation of explanations.

In addition, the distribution of original decisions and counterfactual decisions (i.e., positive and negative) based on Principal Component Analysis is shown in Figure 8. We can see clearly that counterfactual decisions are different from the original, but the counterfactual positive decisions and the negative are close to each other, indicating that they belong to the same distribution and have only a small difference, which reflects that our generated decisions have better generation ability along with better generation quality.

5.7.2 Explanation For User Travel Decisions. In this part, we further illustrate the ability of FCE-UTD to generate causal explanations and how to distinguish spurious explanations and true causal relations with a case study in Figure 9. We first randomly select three different travel decisions and compute the relevance and causal dependencies of factors in each decision. Then we present the differences between causal dependencies and relevance using heat maps and highlight factors with large differences with blue dashed boxes, aiming to explain decisions and distinguish spurious explanations. In addition, we further incorporate users' query and check-in histories to verify and analyze the generated explanations. Concretely, we list the brands of POIs in users' query and check-in histories that belong to the same categories as the users' decisions. Then we present the time distribution histograms of query or check-in behavior to illustrate users' daily preferences and behavioral patterns. Notably, the red font numbers in the heat map indicate causal dependencies, and the black indicate relevance. Besides, to explore the deeper reasons for decisions, we omit identifiers of user and POI. Specifically, for the three user decisions, we have the following findings:

- *D46345 is a check-in decision of User8928 to go to Yonghui Market, which has a high relevance in* Brand(0.2299) *but a high causal dependency in* Query Time Session(0.2198). It demonstrates that the user did not make this decision for a specific brand but probably because he had a need to go to the market to buy necessities. To further verify and analyze this explanation, we checked User8928's query and check-in histories. We found that the user both queried and visited various markets, and he usually queried these POIs at night after a day's work. It is consistent with the explanation drawn from the difference between relevance and causal dependencies. Therefore, if there is a need to recommend POIs for this user, then the recommendation system should recommend markets more often in the evening, instead of focusing on specific brands.
- *D3000 is a check-in decision of User27627 to go to Starbucks, which has a strong relevance and a relatively high causal dependency in* Check-in Time(0.4008, 0.212)*, and a high causal*



(a) The visual distribution of original and counterfactual decisions in SH.

(b) The visual distribution of original and counterfactual decisions in NYC.

(c) The visual distribution of original and counterfactual decisions in ml1m.

Fig. 8. The visual distribution of original and counterfactual (positive and negative) decisions. Green dots indicate original decisions, blue dots indicate positive counterfactual decisions (i.e., label = 1), and red dots indicate negative counterfactual decisions (i.e., label = 0).

dependency in Category(0.2599). We can combine the Category and Check-in Time factors to infer that the real reason for the decision is not only the corresponding check-in time but also the user's preference to drink coffee. To further verify and analyze this explanation, we checked User27627's query and check-in histories. We discovered that the user both queried

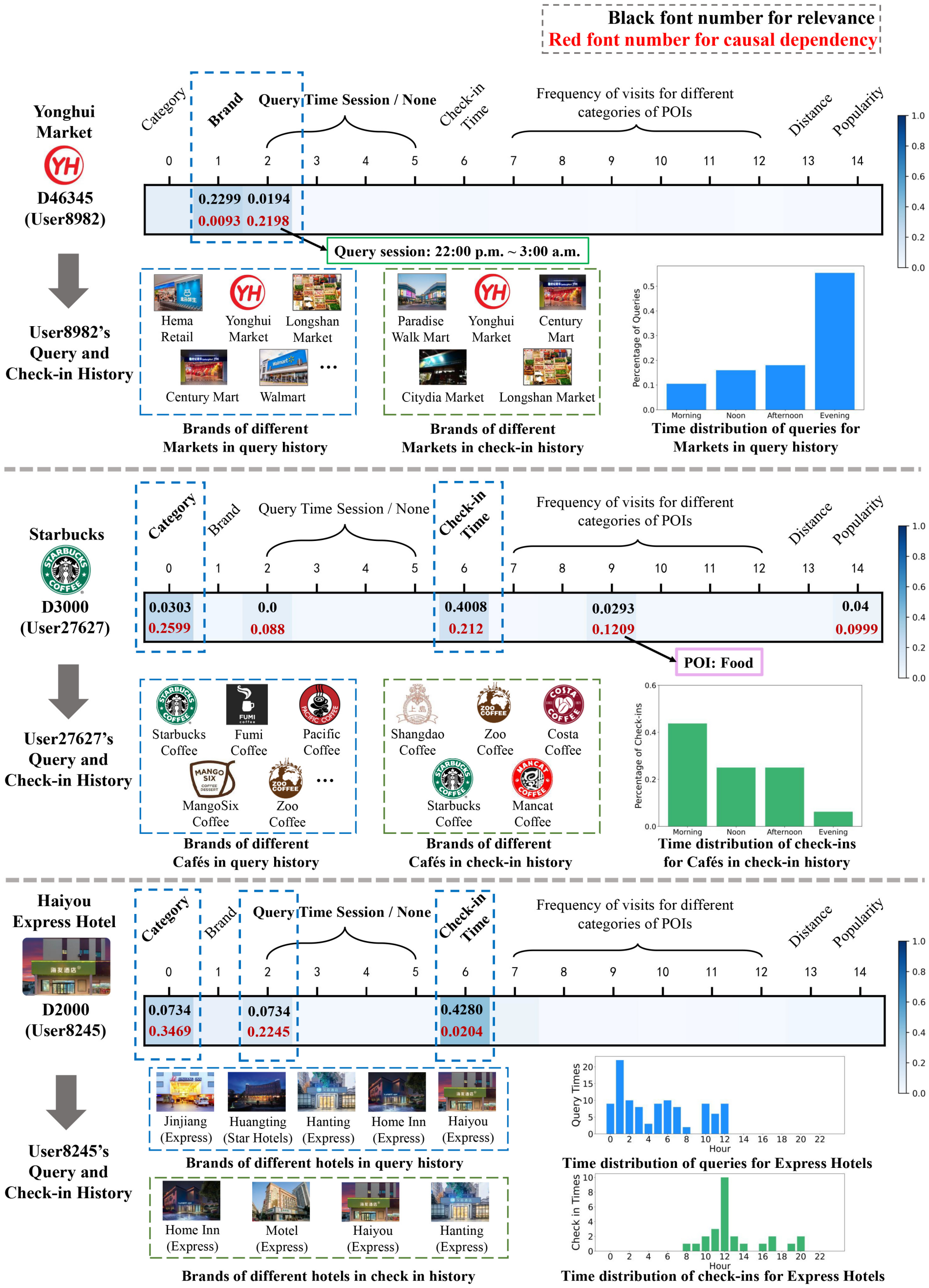


Fig. 9. A case study of the difference between factor relevance and causal dependencies for three decisions by different users.

and visited many different cafes, and he frequently visited POIs with food categories, indicating that this user likes coffee and often explores different cafes or restaurants. Furthermore, the time distribution of the user's check-ins shows that the user has visited cafes several times in the morning, noon, and afternoon, demonstrating that the user is not limited to visit cafes at a specific time. Thus, recommendation systems should tend to recommend different kinds of cafes for the user during the daytime.

- *D2000 is a check-in decision of User8245 to go to Haiyou Express Hotel, which has a high relevance in* Check-in Time(0.4280)*, but exhibits high causal dependencies in* Category(0.3469) *and* Query Time Session(0.2245). This suggests that the real reason for the decision is the user's demand for accommodation and he actually cared more about the convenience attribute of the express hotel rather than when he visited. To further verify and analyze this explanation, we examined the user's query and check-in histories and discovered that the user queried and checked in many different hotels, mainly express hotels. Moreover, the user often queried express hotels in the morning, and often visited at noon and afternoon, suggesting that he may often travel here on business with a need for accommodation, which is consistent with the above conclusion. It motivates recommendation systems to recommend express hotels to the user in the morning.

6 CONCLUSION

In this article, we proposed FCE-UTD, a novel factor-level causal explanation generation framework based on counterfactual data augmentation for user travel decisions. To be specific, we first assume that a user decision is composed of a set of several different factors. Then, we learn the representation of factors and detect the relevant factors by preserving the user decision structure with a joint counterfactual contrastive learning paradigm. Furthermore, we identify causal factors from relevant factors by constructing counterfactual decisions with counterfactual representation generator, which can not only augment the dataset and mitigate the sparsity but also contribute to clarifying the causal factors from other false causal factors that may cause spurious explanations. In addition, a causal dependency learner is proposed to identify causal factors for each decision by learning causal dependency scores, which further leads to factor-level causal explanations. Extensive experiments conducted on three real-world datasets shows the effectiveness of FCE-UTD in terms of check-in rate, fidelity, and downstream tasks under different behavior scenarios. Case studies also validated the ability of FCE-UTD to generate causal explanations and to distinguish spurious explanations and true causality for user travel decisions.

REFERENCES

[1] Behnoush Abdollahi and Olfa Nasraoui. 2016. Explainable matrix factorization for collaborative filtering. In *Proceedings of the 25th International Conference Companion on World Wide Web*. 5–6.

[2] Behnoush Abdollahi and Olfa Nasraoui. 2017. Using explainability for constrained matrix factorization. In *Proceedings of the Eleventh ACM Conference on Recommender Systems*. 79–83.

[3] David Alvarez-Melis and Tommi Jaakkola. 2017. A causal framework for explaining the predictions of black-box sequence-to-sequence models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 412–421.

[4] Konstantin Bauman, Bing Liu, and Alexander Tuzhilin. 2017. Aspect based recommendations: Recommending items with the most valuable aspects based on user reviews. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 717–725.

[5] Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning, Association for Computational Linguistics*. 10.

[6] Shuo Chang, F. Maxwell Harper, and Loren Gilbert Terveen. 2016. Crowd-based personalized natural language explanations for recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*. 175–182.

[7] Xu Chen, Zheng Qin, Yongfeng Zhang, et al. 2016. Learning to rank features for recommendation over multiple categories. In *Proceedings of the 39th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 305–314.

[8] Yongjun Chen, Zhiwei Liu, Jia Li, et al. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.

[9] Jingyue Gao, Xiting Wang, Yasha Wang, et al. 2019. Explainable recommendation through attentive multi-view learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 3622–3629.

[10] Mengyue Hang, Ian Pytlarz, and Jennifer Neville. 2018. Exploring student check-in behavior for improved point-of-interest prediction. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 321–330.

[11] Jing He, Xin Li, Lejian Liao, et al. 2018. Personalized next point-of-interest recommendation via latent behavior patterns inference. arXiv:1805.06316 (2018).

[12] Irina Higgins, Loic Matthey, Arka Pal, et al. 2016. beta-vae: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*.

[13] Yunfeng Hou, Ning Yang, Yi Wu, and S. Yu Philip. 2019. Explainable recommendation with fusion of aspect information. *World Wide Web* 22, 1 (2019), 221–240.

[14] Renjun Hu, Xinjiang Lu, Chuanren Liu, et al. 2021. Why we go where we go: Profiling user decisions on choosing POIs. In *Proceedings of the 29th International Conference on International Joint Conferences on Artificial Intelligence*. 3459–3465.

[15] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining*. IEEE, 263–272.

[16] Jin Huang, Wayne Xin Zhao, Hongjian Dou, et al. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 505–514.

[17] Mengyuan Jing, Yanmin Zhu, Tianzi Zang, and Ke Wang. 2023. Contrastive self-supervised learning in recommender systems: A survey. *ACM Transactions on Information Systems* 42, 2 (2023), 1–39.

[18] Hyemi Kim, Seungjae Shin, JoonHo Jang, et al. 2021. Counterfactual fairness with disentangled causal effect variational autoencoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 8128–8136.

[19] Matt J. Kusner, Joshua Loftus, Chris Russell, et al. 2017. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, Vol. 30 (2017).

[20] Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized transformer for explainable recommendation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing* (Volume 1: Long Papers).

[21] Chang Liu, Chen Gao, et al. 2021. Improving location recommendation with urban knowledge graph. arXiv:2111.01013. Retrieved from https://arxiv.org/abs/2111.01013

[22] Zhiwei Liu, Yongjun Chen, Jia Li, et al. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. arXiv:2108.06479. Retrieved from https://arxiv.org/abs/2108.06479

[23] Zhuang Liu, Yunpu Ma, Yuanxin Ouyang, et al. 2021. Contrastive learning for recommender system. arXiv:2101.01317. Retrieved from https://arxiv.org/abs/2101.01317

[24] Weizhi Ma, Min Zhang, Yue Cao, et al. 2019. Jointly learning explainable rules for recommendation with knowledge graph. In *Proceedings of the World Wide Web Conference*. 1210–1221.

[25] Kelong Mao, Jieming Zhu, et al. 2021. UltraGCN: Ultra simplification of graph convolutional networks for recommendation. In *Proceedings of the Conference on Information and Knowledge Management (CIKM '21)*. 1253–1262.

[26] Andre Martins and Ramon Astudillo. 2016. From softmax to sparsemax: A sparse model of attention and multi-label classification. In *International Conference on Machine Learning*. PMLR, 1614–1623.

[27] Raha Moraffah, Mansooreh Karami, Ruocheng Guo, et al. 2020. Causal interpretability for machine learning-problems, methods and evaluation. *ACM SIGKDD Explor. Newslett.* 22, 1 (2020), 18–33.

[28] Ramaravind K. Mothilal, Amit Sharma, and Chenhao Tan. 2020. Explaining machine learning classifiers through diverse counterfactual explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*. 607–617.

[29] Shanlei Mu, Yaliang Li, Wayne Xin Zhao, et al. 2022. Alleviating spurious correlations in knowledge-aware recommendations through counterfactual generator. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1401–1411.

[30] Ingrid Nunes and Dietmar Jannach. 2017. A systematic review and taxonomy of explanations in decision support and recommender systems. *User Modeling and User-Adapted Interaction* 27, 3.5 (2017), 393–444.

[31] Aaron van den Oord, Yazhe Li, et al. 2018. Representation learning with contrastive predictive coding. arXiv:1807.03748. Retrieved from https://arxiv.org/abs/1807.03748

[32] Sung-Jun Park, Dong-Kyu Chae, Hong-Kyun Bae, et al. 2022. Reinforcement learning over sentiment-augmented knowledge graphs towards accurate and explainable recommendation. In *Proceedings of the 15th ACM International Conference on Web Search and Data Mining*. 784–793.

[33] Georgina Peake and Jun Wang. 2018. Explanation mining: Post hoc interpretability of latent factor models for recommendation systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2060–2069.

[34] Shaowen Peng, Kazunari Sugiyama, and Tsunenori Mine. 2022. SVD-GCN: A simplified graph convolution paradigm for recommendation. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 1625–1634.

[35] Yifang Qin, Yifan Wang, Fang Sun, et al. 2023. DisenPOI: Disentangling sequential and geographical influence for point-of-interest recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*. 508–516.

[36] Lin Qiu, Sheng Gao, Wenlong Cheng, and Jun Guo. 2016. Aspect-based latent factor model by integrating ratings and reviews for recommender system. *Knowledge-Based Systems* 100, 110 (2016), 233–243.

[37] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and et al. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv:1205.2618. Retrieved from https://arxiv.org/abs/1205.2618

[38] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 815–823.

[39] Sungyong Seo, Jing Huang, Hao Yang, et al. 2017. Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In *Proceedings of the 11th ACM Conference on Recommender Systems*. 297–305.

[40] Jie Shuai, Kun Zhang, Le Wu, et al. 2022. A review-aware graph contrastive learning framework for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1283–1293.

[41] Jaspreet Singh and Avishek Anand. 2019. Exs: Explainable search using local model agnostic interpretability. In *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. 770–773.

[42] Dylan Slack, Anna Hilgard, Himabindu Lakkaraju, et al. 2021. Counterfactual explanations can be manipulated. In *Advances in Neural Information Processing Systems*, Vol. 34 (2021), 62–75.

[43] Juntao Tan, Shuyuan Xu, Yingqiang Ge, et al. 2021. Counterfactual explainable recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1784–1793.

[44] Zhiqiang Tao, Sheng Li, Zhaowen Wang, et al. 2019. Log2Intent: Towards interpretable user modeling via recurrent semantics memory unit. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1055–1063.

[45] Daheng Wang, Meng Jiang, Qingkai Zeng, et al. 2018. Multi-type itemset embedding for learning behavior success. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2397–2406.

[46] Xiang Wang, Xiangnan He, Meng Wang, et al. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 165–174.

[47] Zhenlei Wang, Jingsen Zhang, Hongteng Xu, et al. 2021. Counterfactual data-augmented sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 347–356.

[48] Tianxin Wei, Fuli Feng, Jiawei Chen, et al. 2021. Model-agnostic counterfactual reasoning for eliminating popularity bias in recommender system. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1791–1800.

[49] Jiancan Wu, Xiang Wang, Fuli Feng, et al. 2021. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 726–735.

[50] Libing Wu, Cong Quan, Chenliang Li, et al. 2019. A context-aware user-item representation learning for item recommendation. *ACM Trans. Inf. Syst.* 37, 2 (2019), 1–29.

[51] Yao Wu and Martin Ester. 2015. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *Proceedings of the 8th ACM International Conference on Web Search and Data Mining*. 199–208.

[52] Lianghao Xia, Chao Huang, Yong Xu, et al. 2022. Hypergraph contrastive collaborative filtering. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 70–79.

[53] Yikun Xian, Zuohui Fu, Shan Muthukrishnan, et al. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 285–294.

[54] Min Xie, Hongzhi Yin, Hao Wang, et al. 2016. Learning graph-based poi embedding for location-based recommendation. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. 15–24.

[55] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*. IEEE, 1259–1273.

[56] Kun Xiong, Wenwen Ye, Xu Chen, et al. 2021. Counterfactual review-based recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2231–2240.

[57] Shuyuan Xu, Yunqi Li, Shuchang Liu, et al. 2021. Learning causal explanations for recommendation. In *Proceedings of the 1st International Workshop on Causality in Search and Recommendation*.

[58] Mengyue Yang, Quanyu Dai, Zhenhua Dong, et al. 2021. Top-n recommendation with counterfactual user preference simulation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management.* 2342–2351.

[59] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 1434–1443.

[60] Shengyu Zhang, Dong Yao, Zhou Zhao, et al. 2021. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 367–377.

[61] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable recommendation: A survey and new perspectives. *Found. Trends Inf. Retriev.* 14, 1 (2020), 1–101.

[62] Yongfeng Zhang, Guokun Lai, Min Zhang, et al. 2014. Explicit factor models for explainable recommendation based on phrase-level sentiment analysis. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval.* 83–92.

[63] Shenglin Zhao, Tong Zhao, Haiqin Yang, et al. 2016. STELLAR: Spatial-temporal latent ranking for successive point-of-interest recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 30.

[64] Yu Zheng, Chen Gao, Xiang Li, et al. 2021. Disentangling user interest and conformity for recommendation with causal embedding. In *Proceedings of the Web Conference 2021.* 2980–2991.

[65] Kun Zhou, Hui Wang, Wayne Xin Zhao, et al. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management.* 1893–1902.

[66] Ran Zmigrod, Sabrina J Mielke, Hanna Wallach, and Ryan Cotterell. 2019. Counterfactual data augmentation for mitigating gender stereotypes in languages with rich morphology. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.* 1651–1661.

Received 12 August 2023; revised 19 January 2024; accepted 6 March 2024