ConciseExplain: Reducing Redundancy and Spuriousness in Persuasive Recommendation Explanation

Yixuan Cao*†‡
Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, CAS
Beijing, China
caoyixuan@ict.ac.cn

Juyao Liu†‡
Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, CAS
Beijing, China
juyaoliu@gmail.com

Haodong Wang†‡
Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, CAS
Beijing, China
wanghaodong23s@ict.ac.cn

Jian Wang*
China Securities Co., Ltd.
Beijing, China
wangjianzgs@csc.com.cn

Kun Wan*
China Securities Co., Ltd.
Beijing, China
wankun@csc.com.cn

Gang Xiao
China Securities Co., Ltd.
Beijing, China
xiaogang@csc.com.cn

Ping Luo†‡§
Institute of Computing Technology, CAS
University of Chinese Academy of Sciences, CAS
Beijing, China
luop@ict.ac.cn

Abstract

Recommendation systems are effective tools for information filtering and discovery. These systems are widely applied across various consumer sectors and hold significant potential for applications in professional domains to enhance work efficiency. However, supporting decision-making in professional contexts requires not only providing recommendation results but also offering explanations to persuade users to adopt the suggestions. Taking the task in the primary bond market as an example, where sales staff seek potential investors for bonds, this paper presents the development and deployment of a recommendation system designed for a professional setting. The system provides a set of key features as explanations for its recommendations. In this process, we observe that current explanation methods may select redundant and spurious features, which can undermine the persuasive impact of the explanations. To address this issue, we propose a method named ConciseExplain, which leverages a mask training strategy and gradient descent to directly identify a concise set of features. We conduct experiments on real-world and synthetic datasets. Our method achieves relative improvements of 6.1% and 12.4% over the best-performing baseline on redundant and spurious metrics, respectively. Our method also outperforms the baseline method in online manual evaluations. Moreover, during the one-year official deployment of our system at China Securities Co., Ltd. (a leading brokerage firm in China), we observed a continuous improvement in the accuracy of the recommendation system. This suggests that, with concise explanations, a positive feedback loop might be established between recommendation outcomes and investment decisions.

CCS Concepts

• **Information systems** → **Recommender systems**; *Electronic commerce*; *Enterprise information systems*; • **Computing methodologies** → **Artificial intelligence**; **Machine learning**.

Keywords

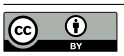
Recommendation; Explainable Recommendation; Trustworthy AI

ACM Reference Format:
Yixuan Cao, Juyao Liu, Haodong Wang, Jian Wang, Kun Wan, Gang Xiao, and Ping Luo. 2025. ConciseExplain: Reducing Redundancy and Spuriousness in Persuasive Recommendation Explanation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2 (KDD '25), August 3–7, 2025, Toronto, ON, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3711896.3737206

1 Introduction

Recommendation Systems (RS) filter items that may be of interest to users, addressing the problem of information overload when users are faced with an overwhelming number of items. While RSs are

*Corresponding authors.
†Key Lab of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing, China.
‡State Key Lab of AI Safety, Beijing, China.
§Peng Cheng Laboratory, Shenzhen, China



KDD '25, Toronto, ON, Canada

ACM ISBN 979-8-4007-1454-2/2025/08
https://doi.org/10.1145/3711896.3737206

predominantly deployed in consumer markets such as e-commerce platforms and short video services, applying RS to professional domains is a promising direction that could significantly enhance work efficiency. However, decision-making in professional domains is much more critical. It is necessary to explain the recommendations to convince users to adopt them.

We take the investor-seeking task in the primary bond market as an exemplary scenario to demonstrate the demand for RS in professional domains and the necessity of explanations of recommendation. In the primary bond market, brokerage firms assist companies (issuers) in issuing bonds to raise funds. A bond must receive sufficient bidding amounts from investors on its issuing day to be successfully issued; otherwise, the issuance fails. Therefore, ahead of the issuing day, the sales staff of brokerage firms endeavor to inform potential investors about the bond's information. Investors who do not receive information about a particular bond are unlikely to invest in it. This is because a brokerage firm releases dozens of new bonds daily, while the market sees hundreds, investors heavily rely on sales staff from various firms to obtain bond details to find interested bonds.

However, seeking potential investors is challenging for sales staff. First, predicting whether an investor will be interested in a bond is difficult: each bond has its unique characteristics, such as the issuer's industry and region, and the bond's interest rate. Each investor also has distinct features, such as historical investment preferences. Sales staff must analyze both the bond's and the investor's features to infer whether the investor might be interested in and willing to invest in a particular bond. Second, the number of potential investors is vast, often numbering in the thousands. Manually matching a bond with suitable investors is labor-intensive. Although challenging, sales staff have to comprehensively identify all potential investors while being selective (even conservative) to inquiries investors because recommending too many unsuitable bonds to investors can seriously harm their reputation and hinder future cooperation.

This problem is akin to selecting products from a large inventory that a customer might purchase, making it a particularly suitable application for an RS. However, in our task, adopting a recommendation is a serious decision because it implies spending significant time contacting investors and engaging in in-depth communication. In the long run, it may also undermine the investor's perception of the sales staff's professional competence. Thus, one unique aspect of using an RS in this task is the need to justify the recommendations by providing explanations to persuade the sales staff.

We developed and deployed an RS for this task. Figure 1 shows a screenshot of the system (anonymized), which provides details of investor recommendations along with explanations. The left panel displays bonds that are about to be issued. When a sales staff member clicks on a bond, the right panel shows a list of recommended investors accompanied by corresponding explanations. Each explanation is a set of tags displayed in colored tags. Blue, yellow, and green boxes represent three different predefined categories of recommendation reasons: namely investor preference for bonds, investor attributes, and investor ability, respectively. Users can hover over the tags to view their full descriptions. For instance, the first two tags of Explanation1 for Recommendation1, namely "Invest Recently" and "Daily Consumption," indicate that the investor has recently made frequent investments (suggesting they may have abundant funds in the near future) and that they prefer bonds issued by companies in Daily Consumption sector.

This paper introduces the model behind this system which generates both recommendation results and explanations. In this task, the input features for recommendation are interpretable, meaning that each feature has a clear, distinct meaning representing some aspect of either the investor or the bond. So, we focus on designing methods to select a subset of these features as explanations for the recommendations, which are the tags shown in Figure 1.

We identify two types of problems within the features selected by current feature-based explanation methods such as LIME [23], SHAP [19], and IG [25]. The first problem is *misleading* or *spurious* features. For example, when the RS recommends an investor for a bond issued by a *listed* company, the explanation provided by existing methods might mistakenly select a feature indicating that the investor prefers *unlisted* companies. Such explanations are not only unconvincing but could also confuse users, discouraging them from adopting the recommendations.

The second problem involves *redundant* features. For instance, in our RS, overlapping features like "Invest Frequency Last Year" and "Invest Frequency Last Month" are intentionally engineered to improve recommendation performance. If these are important features to recommend, existing methods tend to select both for the explanation. This is reasonable for the model interpretation task (aiming to explain *how the model makes predictions*), as these features contribute similarly to the model's decision. However, in our system, which aims to *persuade users* to adopt the recommendation, presenting both to users results in redundant information. Eliminating such redundancy and presenting only one of these features allows space for other features to provide more diverse information for persuasive explanations. These two problems are likely due to correlation among input features while existing methods like Shapley Value-based methods [19] evaluate the importance of each feature respectively. Details are discussed in Section 3.

To address these challenges, we propose **ConciseExplain**, a method to reduce redundant and spurious features in explanation feature sets. The proposed method derives from identifying a set of features that maximizes the expected probability that investors with this feature set will bid on a bond. We train a model to estimate the bidding probability given a subset of features, thereby considering selected features collectively. A gradient-based optimization method is then used on this model to identify a subset of size k that maximizes the probability.

The proposed system has been deployed at China Securities Co., Ltd. (CSC), a leading brokerage firm in China, since March 2023. Experimental results indicate that our method outperforms the best-performing baseline methods, achieving relative improvements of 6.1% in rationality and 12.4% in diversity on automatic evaluation, and is more preferred in 4.2% cases in manual evaluation. During the first nine months of deployment, we observed a steady improvement in the RS's recommendation accuracy. This suggests that with the support of explanations of recommendations, sales staff have become increasingly inclined to adopt the system's suggestions over time.

Figure 1: The screenshot of our explainable recommendation system for investor-seeking.

The proposed method has broad application value. In the bond market, for example, corporate credit bonds issued in China's bond market reached $2 trillion in 2024, and improving the efficiency of the bond market would yield significant benefits. It can also be applied to other scenarios, such as online recruitment platforms, to explain to HR why a candidate is recommended.

2 RELATED WORK

In this section, we first introduce general methods for feature-level explanations. Then, we introduce explanation methods specifically designed for RS, including feature-level explanations studied in this paper, as well as other forms of explanations, such as text-based and sequence-based approaches.

Generating feature-level explanations is widely studied in eXplainable AI (XAI). In this field, two approaches are particularly pertinent to our work. The first approach involves anchor-based methods [5, 24], which explain model decisions by identifying specific conditions that lead to a particular classification outcome, such as a feature equaling a specific value. While conceptually similar to our approach, integrating these methods into recommendation systems is non-trivial since recommendation models typically generate scores instead of discrete labels. Moreover, deriving such explanations has been proven to be a NP-hard task [22] posing significant challenges for practical implementations in recommendation systems, which generally involve numerous features.

The second approach entails feature attribution methods that quantify the contribution of each feature to the output. Among these methods, LIME [23] and SHAP [19] are the most well-known model-agnostic methods. These interpret a model's output for a given instance by perturbing its features. Notably, SHAP and our approach share a mathematical affinity through the use of value functions, even though their interpretative goals diverge.

Furthermore, given the differentiable nature of most deep neural networks, several gradient-based explanation methodologies [7, 25] have been developed. These methods leverage gradients with respect to input features to estimate feature significance. Although feature attribution-based explanation methods provide a pathway to selecting recommendation explanations by feature contributions, they treat each feature independently, often resulting in redundant and spurious features in explanations.

For explanations in RS, few methods are designed specifically for feature-level explanations. Lime-RS [21] introduces a variant of LIME [23] that modifies the perturbation strategy for the instance being considered. Meanwhile, RecXplainer [28] adopts an auxiliary model that predicts using user embeddings and one-hot encoded item features. This model conceptualizes a user's preference for an item attribute as the decrease in the auxiliary model's prediction score when the corresponding item feature is set to a *zero vector*. Both methods compute the importance of each feature independently, which may lead to redundant and spurious features when selecting a feature set for explanations.

There are also some other forms of explanations in RS. In scenarios where users write reviews on items, some studies build explainable RSs through personalized review generation using language models [16, 17, 32]. Another way to utilize user reviews is by extracting users' attention and item advantages from these reviews to generate counterfactual explanations [8, 26]. For sequential recommendation, some research proposes presenting the items a user previously interacted with as the recommendation reason [9, 29]. Different explanation forms are suitable for different scenarios. In this work, we focus on feature set-based explanations which is suitable for bond recommendation scenarios.

3 A Motivating Toy Example

We use a toy example to intuitively demonstrate why correlations among features can lead to spurious and redundant features, and what kind of approach might help mitigate this issue.

Considering an e-commercial system. Suppose that we have a set of products, each with four features, namely *price1*, *price2* $\in \{low, high\}$, *quality* $\in \{low, high\}$, and *color* $\in \{blue, other\}$. In our data, we intentionally set *price1*=*price2* for each instance to design a pair of redundant features. Also, *price* and *quality* are positively correlated since low-quality products usually have low prices. Specifically, each sample is generated as follows:

(1) Sample $price1 \sim Bernoulli(0.5)$, where 0 and 1 represent low and high prices.
(2) Set $price2 = price1$.
(3) if $price1 = 0$, $quality \sim Bernoulli(q)$, where $q = 0.1$, otherwise $q = 0.9$, and 0 and 1 represent low and high quality.
(4) Sample $color \sim Bernoulli(0.5)$, where 1 and 0 represent blue and other colors, respectively.
(5) Sample $y \sim Bernoulli(0.8\times(1-price1)+0.1\times quality+0.1\times color)$, where $y = 1$ means the user purchases the item.

Table 1: Feature scores produced by different methods.

	price1 =low	*price2 =low*	*quality =low*	*color =blue*	ranking
LIME	0.44	0.27	-0.01	0.10	*price1>price2>color>quality*
Kernel-SHAP	0.22	0.14	-0.01	0.04	*price1>price2>color>quality*
IG	0.14	0.21	0.01	0.05	*price2>price1>color>quality*
EG	0.21	0.13	-0.02	0.05	*price1>price2>color>quality*
Kernel-SHAP-M	0.11	0.12	0.07	0.03	*price2≈price1>quality>color*
weightedSHAP	0.04	0.02	0.01	0.03	*price1>price2>color>quality*

According to (5), the probability for this user to buy a product is:

$$P(Y = 1|price, quality, color) = 0.8 \times (1 - price1) + 0.1 \times quality + 0.1 \times color.$$

Suppose an item with features {*price1 = low*, *price2 = low*, *quality = low*, *color = blue*} is recommended to the user. Now we hope to tell the user why our model recommends this item to her. Clearly, the expected explanation with different numbers of features are

(1) |S|=1: {price1} or {price2}
(2) |S|=2: {price1, color} or {price2, color}
(3) |S|=3: {price1, price2, color}.

We do not want *quality=low* in explanation since it reduces the probability compared with when it takes the value *high*.

Table 1 shows the feature scores produced by baselines. These baselines score each feature individually and output the top k features as explanations. Take Kernel-SHAP-M as an example. When required $|S| = 2$, it will generate redundant features *{price1=low, price2=low}* as the explanation because they get the highest scores of 0.11 and 0.12. When $|S| = 3$, it adds *quality=low* (with the third highest score) to the explanation. Here, *quality=low* gets a high feature score due to its correlation with the low-price feature preferred by the user. However, it is unlikely to persuade the user to accept this recommendation as the user actually slightly prefers high-quality products. More detailed analyses are provided in Appendix C.

Thus, considering the correlations among features (considering a feature set jointly) is important to eliminate such problems.

4 METHOD

4.1 Problem Formulation

This work focuses on producing persuasive explanations for recommendation results in a professional context. In the bond recommendation scenario studied in this paper, the system recommends potential investors to bonds. This task intrinsically aligns with recommendation tasks in other domains, such as online shopping where the system recommends products to users. Here, bonds correspond to users, and investors correspond to products.

We first briefly introduce the recommendation task as preliminary knowledge. Denote investors in the primary bond market as $I = \{i_1, i_2, \ldots, i_{N_i}\}$ and bonds we underwrite as $B = \{b_1, b_2, ..., b_{N_b}\}$. For a bond $b \in B$ with features $\boldsymbol{x}^b$ and an investor $i \in I$ with features $\boldsymbol{x}^i$, the recommendation model f predicts the probability that i will invest b, denoted as $f(\boldsymbol{x}^b, \boldsymbol{x}^i; \theta)$. The top-m investors with the highest predicted probabilities, $R(b) = \{c_1, c_2, \ldots, c_m\}$, are recommended for b.

Next, we introduce the **Persuasive Explanation Task** studied in this paper. Suppose we recommend i to b. The explanation task is to select k features of $\boldsymbol{x}^i$, i.e., $\boldsymbol{x}^i_S = \{x^i_j | j \in S\}$ where S is the index set of selected features, that best explains why to recommend i for bond b and persuade users to adopt it.

In this work, we only select features belonging to the investor to provide explanations, rather than features of the bond. This choice is motivated by the fact that when identifying potential investors for a given bond, sales staff are usually more familiar with the bond's characteristics but may have limited knowledge about many investors. Explaining investor features helps them make more informed decisions. However, our method can be extended to use bond features or a combination of both.

To address the limitation of considering each feature independently (discussed in Section 3), our method jointly considers multiple features. We first illustrate our main idea through a hypothetical ideal scenario. Assuming we have access to infinite investing outcomes for bonds and investors (even the bond we are issuing). For a particular investor i and bond b, the probability of investing can be computed as $P(Y = 1|\boldsymbol{x}^b, \boldsymbol{x}^i) = \frac{|\{(i', b', y)|\boldsymbol{x}^{b'} = \boldsymbol{x}^b, \boldsymbol{x}^{i'} = \boldsymbol{x}^i, y=1\}|}{|\{(i', b', y)|\boldsymbol{x}^{b'} = \boldsymbol{x}^b, \boldsymbol{x}^{i'} = \boldsymbol{x}^i\}|}$, where (i', b', y) and $y = 1$ means i' bid on b'. Similarly, for a given subset of investor features $\boldsymbol{x}^i_S$, we can estimate the investing probability $P(Y = 1|\boldsymbol{x}^b, \boldsymbol{x}^i_S)$ based on subsets of features, which serves as a metric for the importance of $\boldsymbol{x}^i_S$:

$$P(Y = 1|\boldsymbol{x}^b, \boldsymbol{x}^i_S) = \frac{|\{(i', b', y)|\boldsymbol{x}^{b'} = \boldsymbol{x}^b, x^{i'}_j = x^i_j \text{ for } j \in S, y = 1\}|}{|\{(i', b', y)|\boldsymbol{x}^{b'} = \boldsymbol{x}^b, x^{i'}_j = x^i_j \text{ for } j \in S\}|}.$$

We call it the **sufficient value** of $\boldsymbol{x}^i_S$, as it measures how sufficient $\boldsymbol{x}^i_S$ is in outputting a high recommendation score.

Based on sufficient value, we formulate the explanation task as selecting the feature index set $S = \{S_1, ..., S_k\} \subseteq \{1, 2, \ldots, N\}$ that maximizes the *sufficient value*:

$$\begin{cases} \max\limits_{S \subseteq \{1,2,...,N\}} & P(Y = 1|\boldsymbol{x}_S) \\ \text{s.t.} & |S| = k \end{cases}, \tag{1}$$

where $\boldsymbol{x}_S = (\boldsymbol{x}^b, \boldsymbol{x}^i_S)$. Subsequently, without affecting understanding, we will combine $\boldsymbol{x}^b$ and $\boldsymbol{x}^i$ and write them as $\boldsymbol{x}$.

This formulation helps detect and reduce redundant features in explanations because adding a redundant feature to S likely will not improve its sufficient value. For example, in the toy example, if $\boldsymbol{x}_S = \{$*price1=low*$\}$ yields a high $P(Y = 1 \mid \boldsymbol{x}_S)$, adding *price2=low* to $\boldsymbol{x}_S$ will not increase this value as it is implied by *price1*. Moreover, it also helps minimize spurious features in explanations since spurious features combined with the main features in S tend to reduce its sufficient value. A detailed analysis is provided in Appendix C.

Clearly, the sufficient value cannot be directly computed statistically. So we propose a Mask Training Strategy to train the recommendation model f to estimate sufficient value (Section 4.3). Then, during inference, we recommend investors for each bond using f, and explain each recommendation via gradient descent on f using gates of features (Section 4.4).

This paper distinguishes between "**explanations**" and "**reasons**". An "explanation" refers to the set of selected features, while a "reason" refers to one feature within that set.

4.2 Model Structure

We first briefly introduce the structure of the recommendation model. In this paper, we consider recommendation models with two properties. 1. **Embedding-based**: The input features are transformed into embeddings before being fed into the neural network. 2. **Differentiable**: The model output needs to be differentiable w.r.t. the embedding. Most deep learning-based ranking models for industrial recommendation systems satisfy these two properties.

Suppose all features are in the categorical form [1] and an instance can be represented as $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, where $x_j \in V_j$ is the j_{th} feature of $\boldsymbol{x}$ and V_j is the set of unique IDs of the j_{th} feature. The sparse one-hot representation of x_j is then mapped to a dense vector via the embedding layer:

$$e_j = W^j_{emb}\text{one-hot}(x_j), \tag{2}$$

where $e_j \in \mathbb{R}^{d_j}$ is the embedding of x_j, $W^j_{emb} \in \mathbb{R}^{d_j \times |V_j|}$ is the embedding matrix of the j_{th} feature, d_j is the embedding size, and $\text{one-hot}(x_j) \in \{0,1\}^{|V_j|}$ is the one-hot representation of x_j. Then, these features are fed into deep models, such as DCN [30], to compute the probability of recommendation.

To train a model f to predict the probability of user-item interaction, we use cross-entropy:

$$L(\theta) = -\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\text{CE}(f(\boldsymbol{x};\theta), y), \tag{3}$$

where $\mathcal{D}$ is the distribution of training data, $y \in \{0,1\}$ is the label indicating whether the investor bids on the bond, and $\text{CE}(f(\cdot), y) = y\log f(\cdot) + (1-y)\log(1-f(\cdot))$ is the binary cross entropy.

4.3 Mask Training Strategy

The recommendation model f cannot take a subset of input $\boldsymbol{x}_S$ to compute the probability, so it cannot estimate sufficient values. Inspired by the masking method of BERT [6], we fill the non-selected features with special masks to synthesize a full example and use $f(\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}} = \texttt{mask}; \theta)$ to estimate $P(Y=1|\boldsymbol{x}_S)$, where $\boldsymbol{x}_{\bar{S}} = \texttt{mask}$ means $x_j = \texttt{mask}_j$ for $j \notin S$. Specifically, we add an embedding vector for each embedding matrix W^j_{emb} to represent $\texttt{mask}_j$.

However, the model trained with Equation 3 does not encounter samples with masks ($\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}} = \texttt{mask}$) during training, and thus does not guarantee an approximation to $P(Y=1|\boldsymbol{x}_S)$. So, we propose a Mask Training Strategy (MTS) that simulates masked samples by randomly masking features before feeding them to the model during training. The process is shown in Algo 1. The training loss of MTS is:

$$L_{MTS}(\theta) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\mathbb{E}_S\left[CE\left(f(\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}} = \texttt{mask}; \theta), y\right)\right], \tag{4}$$

where S follows the distribution in Algo 1. This loss covers the ordinary loss defined in Equation 3 because there are sampled examples whose $S = \emptyset$.

By minimizing L_{MTS}, we theoretically prove that when given a full example, $f(\boldsymbol{x};\theta)$ estimates the bidding probability $P(Y=1|\boldsymbol{x})$, and when given a selected feature set, $f(\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}} = \texttt{mask}; \theta)$ estimates the sufficient value $P(Y=1|\boldsymbol{x}_S)$. In other words, the model trained with MTS can recommend items and explain itself simultaneously.

[1] We convert continuous features into categorical forms using adaptive binning, and we find this leads to a better recommendation performance.

Algorithm 1 Masked Training Strategy

```
for each (x, y) in dataset do
    p_m ← random.uniform(0, 1)
    S ← ∅
    for j ∈ {1, 2, ..., N} do
        if random.uniform(0, 1) < p_m then
            S ← S ∪ {j}
        end if
    end for
    Compute loss using (x_S, y) according to Equation (4).
end for
```

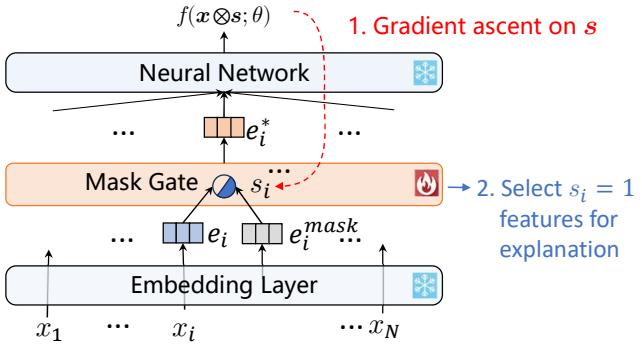


Figure 2: Selecting features for explanations through optimization of $\boldsymbol{s}$ to maximize $f(\boldsymbol{x}\otimes\boldsymbol{s};\theta)$.

The proof is provided in Appendix B. After training with MTS, given a $\boldsymbol{x}_S$, we can estimate the probability $P(Y=1|\boldsymbol{x}_S)$ in Equation 1.

4.4 Explaining via Gradient Descent

Now, we introduce a gradient descent method that approximately solves the optimization problem defined in Equation 1 to find the best explanation S. Given a recommended i on b, it conducts gradient ascent on $f(\boldsymbol{x}^b, \boldsymbol{x}^i)$ by altering which features should be masked.

We re-organize the computing process of f by rewriting the masking process ($\boldsymbol{x}_S, \boldsymbol{x}_{\bar{S}} = \texttt{mask}$) as gating on the embeddings to switch between ordinary feature and mask feature embeddings. Specifically, we design a gating binary vector $\boldsymbol{s} \in \{0,1\}^N$ where $s_j = 1$ if $j \in S$, otherwise $s_j = 0$. The embedding fed into the neural network is $e^*_j = s_j e_j + (1-s_j)e^{mask}_j$, denoted as $\boldsymbol{x}\otimes\boldsymbol{s}$. This is illustrated as the "Mask Gate" block in Figure 2. So, the optimization problem of Equation 1 is converted to:

$$\begin{cases} \max\limits_{\boldsymbol{s}\in\{0,1\}^N} & f(\boldsymbol{x}\otimes\boldsymbol{s};\theta) \\ \text{s.t.} & \sum_{j=1}^N s_j = k \end{cases}. \tag{5}$$

Equation 5 is a combinatorial optimization task with $\binom{N}{k}$ potential configurations for $\boldsymbol{s}$. We approximate it as a minimization problem with the following objective:

$$L_{explain}(\boldsymbol{s}) = -f(\boldsymbol{x}\otimes\boldsymbol{s};\theta) + \alpha(\|\boldsymbol{s}\|_1 - k)^2, \tag{6}$$

Table 2: Typical features used in our system.

Feature Group	Feature Name	Range
Investor Features	organization type (e.g. bank/fund/...)	1-12
	invest frequency	1-5
	yearly preference level for AAA bonds (preference feature → bond feature - rating)	1-5
	monthly preference level for AAA bonds (preference feature → bond feature - rating)	1-5
	yearly preference level for AA bonds (preference feature → bond feature - rating)	1-5
Bond Features	rating (e.g. AAA/AA/...)	1-5
	issuance period (e.g. 1 year/3 years/...)	1-5

where α is the weight of regularization that restricts the number of generated reasons. Then, we solve it via gradient descent by altering $\boldsymbol{s}$. However, since $\boldsymbol{s}$ is categorical and thus non-differentiable, we cannot directly apply gradient descent on $\boldsymbol{s}$. So, we adopt the Gumbel-Softmax trick for optimization, which learns a continuous parameter as the probability for sampling the gate. You may refer to [13, 20] for details about Gumbel-Softmax.

5 Experiments

The main experiments are conducted on a **real-world bond dataset**. We first introduce an automatic evaluation experiment, which allows us to evaluate and improve our algorithm during development without the help of sales staff. Next, we introduce the manual evaluation with sales staff. Finally, we introduce the deployment results achieved at China Securities Co., Ltd (CSC). An experiment on a **synthetic dataset** is also conducted to demonstrate the generalization capacity of our method. The results shown in Appendix C imply a similar conclusion like the real-world dataset.

5.1 Dataset

We collected a dataset of in-house bidding history for bonds underwritten by CSC over the past 3 years. It contains millions of investor-bond interactions (bid/not bid), along with 32 bond features and 259 investor features. Some of these features are described in Table 2. The dataset is split into training, validation, and test sets based on time, with proportions of 6:2:2.

During testing and deployment, we recommend 50 investors for each bond. For each recommended investor, we select $k = 6$ investor features as the explanation. The numbers of recommendations and features were collaboratively determined with sales staff.

5.2 Settings

Baselines: We compare our method with the following explanation baselines: LIME [23]; Kernel SHAP [19], which applies the Shapley kernel to LIME; Weighted SHAP [15], which generalizes the Shapley value and learns which marginal contributions to focus on directly from the data; IG [25] (we use zero vectors as the baseline value for feature embeddings); EG [7]; and RecXPlainer-L/RecExplainer-MLP, which are RecXPlainer [28] with a linear model or MLP as the auxiliary model. All baseline methods mentioned above are attribution-based explanation techniques that assign attribution scores to features. We select the features with the top-k scores as explanations. Additionally, we extend Kernel SHAP and IG by using our MTS-trained mask embedding as the baseline value, denoted as Kernel-SHAP-M and IG-M.

Implementation Details: We use MTS to train DCN-V2 [30] as the recommendation model with zero dropout rates. All explanation methods are applied to this model in the experiment for a fair comparison. (The explanation quality on the model trained **without** MTS can be found in Appendix A.5 which shows similar conclusions). We use Adam [14], with lr=0.1, to minimize Equation 6, where α=0.04, and perform 100 optimization steps.

5.3 Automatic Evaluation Metrics

Evaluating the quality of an explanation is generally challenging due to the difficulty in defining and annotating ground-truth explanations. Most existing methods rely on manual evaluation [1, 19, 23, 24]. In our study, the task is highly specialized, making it difficult for non-expert annotators to evaluate explanations, whereas asking professional sales staff to evaluate explanations is expensive. This makes it challenging for us to evaluate and improve our method during development.

Fortunately, we can leverage investor preference features to automate the evaluation. Specifically, we propose two quantitative metrics—rationality and diversity—for evaluation.

Rationality: A rational reason should encourage users to adopt a recommendation, while an irrational reason may confuse or discourage users. We first define when a reason is rational in our scenario. Here, we only consider investor preference features. Preference features (P-feature or P-reason) reflect the investor's preference towards a specific feature of bonds in a period. If a preference feature in the explanation is consistent with the corresponding bond feature, it is rational. For example, in Table 2, *yearly preference level for AAA bonds* is a P-feature reflecting the proportion of AAA-rated bonds over the bonds it has bid in the recent year. Then, for a bond with feature *rating=AAA*, the reason "*yearly preference level for AAA bonds=high*" is rational, and "*yearly preference level for B bonds=high*" is irrational, as it will confuse the sales staff why an investor preferring B-rated bonds is recommended for this AAA-rated bond. Based on the definition, we adopt Cohen's kappa coefficient [4] over all P-reasons in generated explanations of all samples compared against their corresponding bond features, as the rationality score.

Diversity: A diverse explanation should contain features from different aspects to convince the user to adopt the recommendation. So, we define how to measure diversity in an explanation. We adopt the classic topic modeling method, Probabilistic Latent Semantic Analysis (PLSA) [11], to generate a topic vector s for each feature (details can be found in Appendix A.3) and calculate the average cosine similarity among the topic vectors of features in an explanation. Formally, the diversity of an explanation $\boldsymbol{x}_S^i$ is $d_{\boldsymbol{x}_S^i} = 1 - 1/\binom{k}{2} \sum_{1 \le i < j \le k} \cos(s_{r_i}, s_{r_j})$, where s_{r_i} is the topic vector of the reason $r_i \in \boldsymbol{x}_S^i$. Then, the overall diversity is the average of

Table 3: Explanation quality results.

Method	Rationality	Diversity	HRD	AESV	Time (s) [2]
Baselines:					
LIME	0.324	0.520	0.399	0.339	0.61
Kernel SHAP	0.503	0.624	0.557	0.391	101.2
Weighted SHAP	0.457	0.662	0.540	0.393	11.4
IG	0.655	0.622	0.638	0.414	0.050
EG	0.652	0.566	0.605	0.390	0.052
RecXPlainer-L	0.027	0.577	0.051	0.255	**0.002**
RecXPlainer-MLP	0.178	0.404	0.247	0.318	0.025
Baselines + our MTS-generated mask vector:					
Kernel SHAP-M	0.650	0.603	0.625	0.421	10.1
IG-M	0.659	0.632	0.645	0.417	0.051
Ours	**0.695**	**0.701**	**0.697**	**0.439**	0.165

all explanations' diversities: $diversity = \frac{1}{|B|m} \sum_{b \in B} \sum_{i \in R(b)} d_{\boldsymbol{x}_S^i}$. Note that $0 \leq diversity \leq 1$, because the entries of topic vectors in PLSA are non-negative.

HRD: To comprehensively evaluate the explanation quality, we adopt the harmonic mean of Rationality and Diversity (HRD):

$$HRD = \frac{2 \cdot Rationality \cdot Diversity}{Rationality + Diversity}. \tag{7}$$

Note that the metrics defined above are tailored for our dataset and may not fully reflect the explanation quality. However, they can be automatically evaluated and reflect the quality of models. Although they can be easily maximized by designing tailored rule-based methods to trim the explanation, our methods and baseline methods do not incorporate these metrics in their design, thus using these metrics for comparison is fair. We expect that methods that perform better on these two metrics will also perform better on aspects of rationale and diversity that cannot be measured by these metrics.

AESV (Average Empirical Sufficiency Value): The Empirical Sufficiency Value (ESV) of an explanation $\boldsymbol{x}_S^i$ is the empirical value of sufficient value:

$$ESV(\boldsymbol{x}^b, \boldsymbol{x}_S^i) = \frac{|\{(i', b, y) \mid i' \in I; \boldsymbol{x}_S^i \subseteq \boldsymbol{x}^{i'}; i' \text{ bid } b)\}|}{|\{(i', b, y) \mid i' \in I; \boldsymbol{x}_S^i \subseteq \boldsymbol{x}^{i'}\}|}. \tag{8}$$

And AESV is the average of ESV for all explanations: $AESV = \frac{1}{N_b m} \sum_{b \in B} \sum_{i \in R(b)} ESV(\boldsymbol{x}^b, \boldsymbol{x}_S^i)$. This metric requires data labels and thus can only be used for evaluation. This metric is similar to the faithfulness metric [2, 12], but as we have set a fixed number of reasons, we directly adopted the AESV metric.

5.4 Result and Analysis

The results are shown in Table 3. Our explanation method has obvious advantages over baselines in both rationality and diversity. It outperforms the best baseline (i.e., IG) by an absolute margin of 0.040 in rationality and 0.079 in diversity, which translates to a 9.2% relative improvement in HRD. Integrating the MTS-trained masking vector into baseline methods can improve their performance, but our method consistently outperforms the IG-M and Kernel SHAP-M methods.

Next, we will use some **cases** to illustrate the details of the result. Figure 3 shows the explanations for two recommended investors on one bond. The features of the target bond are shown on the left panel, and the explanations for the two recommended investors are shown on the right. If a reason starts with an index number in the figure, it is a P-reason (refer to the definition of Rationality). That number corresponds to the index of the bond feature in the left panel. For example, the first reason selected by Kernel SHAP-M method on Investor 1 starts with 4), corresponding to bond feature 4) "debt type: quasi-public offering". Green-numbered reasons are rational, while red-numbered ones are irrational. For example, the second reason from Kernel SHAP-M is irrational as it conflicts with the value of bond feature 5). If a reason starts with a black square in the figure, it is not a P-reason, and we do not judge whether it is rational. The reasons enclosed by the curly brace are redundant.

Regarding rationality, the greater number of red reasons in baseline explanations indicates that the baselines are less rational. For example, Kernel-SHAP-M explains that investor 1 prefers the *central state-owned issuer*, while the bond is issued by the *local state-owned issuer*. By analyzing the dataset, we find that this is because a preference for *central state-owned issuers* is highly correlated with a preference for *quasi-public offerings* (feature 4) of this bond. So, baselines are misled by spurious correlations among features.

Research Question Q1: How does our method discard the spurious feature "prefer central state-owned issuer last year" in this example? We examine the dynamics of gate parameters (the probability parameter in Gumbel-Softmax) of two investor features during gradient descent. The two features are "prefer **C**entral State-owned Issuer last year" and "prefer **Q**uasi-Public Offering last month", and we denote their gates as s_C and s_Q respectively. As shown in Figure 4a, when s_Q is low, s_C increases with the optimization steps because s_C can enhance the network's output due to their positive correlation. However, once s_Q surpasses a certain threshold (after 80 steps), further increasing s_C decreases the network's output, causing s_C to decrease. Therefore, our method will dismiss the spurious reason "prefer central state-owned issuer last year". To further illustrate that s_C is suppressed by s_Q, Figure 4b shows that s_C will increase to 1 during gradient descent if we force $s_Q = 0$.

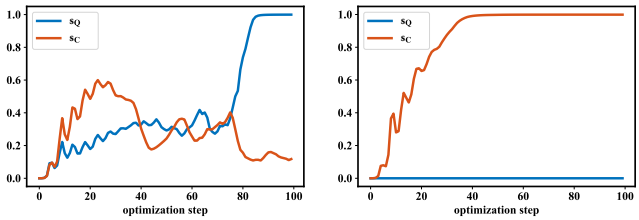
Similarly, Kernel SHAP-M and IG-M explain that investor 1 prefers renewable bonds, which is inconsistent with the current bond, due to spurious correlations with which the investor is active (investing frequently and in large amounts). Our method avoids generating such spurious reasons by jointly considering selected reasons.

Regarding diversity, for investor 2 in Figure 3, our method generates fewer redundant features. Although our method and baselines all provide rational explanations, our method can provide a more convincing explanation by providing more information within the same number of reasons. For example, we explain that investor 1 prefers issuers in Beijing and investor 2 prefers 5-year bonds, which are not revealed by baseline methods.

[2]We measure the average time for generating an explanation for one sample, using 8 cores of two Intel Xeon Gold 6248R processors.

Target Bond Features:		Kernel SHAP-M	IG-M	OURS
1) period: 5 year 2) subject rating: AAA 3) debt rating: AAA 4) debt type: quasi public offering 5) renewable: False 6) issuer industry: real estate 7) issuer province: Beijing 8) issuer property: local state-owned 9) issuer type: non-financial 10) issuer is listed: False	**Investor 1**	4) prefer quasi public offering last month 5) prefer renewable bond last year 8) prefer central state-owned issuer last year 9) prefer issuer under SASAC last month ■ invest frequently last year ■ invest large amount last year	3) prefer AAA bond last month 4) prefer quasi public offering last month 5) prefer renewable bond last year ■ invest frequently last year ■ invest large amount last year ■ is insurance company	3) prefer AAA bond last month 4) prefer quasi public offering last year 7) prefer issuer in BeiJing last year ■ invest frequently last year ■ invest large amount last year ■ invest at high interest rate last year
	Investor 2	2) prefer AAA subject last month 4) prefer quasi public offering last month 4) prefer quasi public offering last year ■ invest frequently last month ■ invest frequently last year ■ invest large amount last year	2) prefer AAA subject last month 2) prefer AAA subject last year 4) prefer quasi public offering last month 4) prefer quasi public offering last year ■ invest frequently last month ■ invest large amount last year	1) prefer 5-year bond last year 2) prefer AAA subject last month 4) prefer quasi public offering last month 5) prefer unrenewable bond last year ■ invest frequently last month ■ invest frequently last year

Figure 3: Case study. Explanations for two investors recommended for one bond, produced by three methods. Our method produces fewer spurious (red-numbered) and less redundant (in brace) reasons.



(a) Optimize on both parameters. **(b) Freeze $s_Q = 0$**

Figure 4: Changes of gates s_Q and s_C during gradient descent indicate that our method can suppress spurious features.

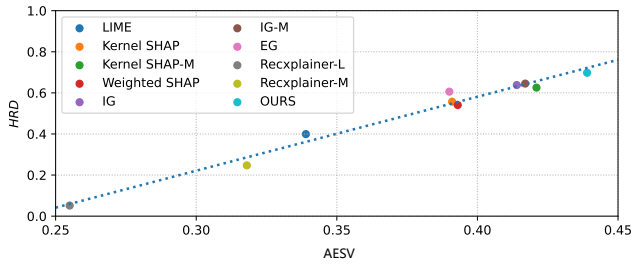


Figure 5: Explanation quality is highly correlated with the average sufficiency value.

5.5 Discussion and Ablation Study

Next, we conduct several detailed analyses of our proposed method by answering 3 research questions.

Q2: Is the objective defined in Equation 1 suitable to achieve a concise explanation, and how well do we optimize it? Figure 5 shows that AESV has a strong correlation with explanation quality (i.e., HRD) for all methods. This indicates that the objective is suitable and can lead to rational and diverse explanations. Moreover, as shown in Table 3, our method achieves a higher AESV than the best baseline (i.e., IG) by a margin of 6.0% relatively, which means that our method does optimize this objective.

Table 4: Ablation on mask embedding for explanation.

Method	Rationality	Diversity	AESV
Ours	**0.695**	**0.701**	**0.439**
Zero embedding	0.659	0.631	0.424
Avg embedding	0.343	0.644	0.333

Q3: What is the effect of MTS and the learned mask embeddings? First, we conduct an ablation study on the learned mask embeddings. We use a zero vector or average embedding vector of each feature to replace the mask embedding when explaining instances. Specifically, after training the recommendation model with MTS, we replace the mask embedding with a zero/average embedding when we optimize $\boldsymbol{s}$. Table 4 shows that mask embedding is crucial for the algorithm and that replacing mask embeddings with zero or average embedding will severely harm the explanation quality. This reflects that the embedding vectors generated by MTS are important for explanation.

Second, using the learned mask embeddings will improve the explanation performance of other methods. The *Baselines+ our MTS generated mask vector* part of Table 3 shows that IG and Kernel SHAP can be promoted by applying our trained mask embeddings as the baseline value (i.e., IG-M, Kernel SHAP-M). For IG, compared with the zero vector, the mask embedding represents an unknown feature and is more suitable for the baseline embedding. For Kernel SHAP, using mask embedding changes the SHAP value from off-manifold to on-manifold, leading to less spuriousness caused by algebraic model dependence.

Moreover, MTS has a negligible influence on recommendation performance. We compare training with or without MTS on three datasets and four models. Their recommendation performances are similar, as detailed in Appendix A.4.

Table 5: Similar features improve recomm. performance.

	AUC	LogLoss↓	Rec@50	Pre@50
Basic Model	0.9038	0.0791	0.552	0.221
- monthly preference	0.9027	0.0804	0.518	0.211
- yearly preference	0.8997	0.0795	0.522	0.212

Q4: Can we remove all similar features throughout the RS to avoid redundancy? The answer is no. First, the correlation among features is complex, defining what are similar features is difficult. For example, the cases discussed above show some subtle correlations between features that are intuitively irrelevant. Second, removing similar features by rules will decrease the recommendation performance. Table 5 shows that removing even very similar features, such as removing monthly preference features while retaining yearly preference features, will still decrease the recommendation performance. So, in real-world applications, we have to incorporate all features because they may contain different information that can be captured by the model. This justifies that our study on reducing similar features in explanation is necessary.

5.6 Online Manual Evaluation

We conducted an online test with the sales staff at CSC as follows. For multiple recommended investors for a bond recommendation, two sets of explanations were presented side by side under each investor, each containing 6 features. These two sets were the results of our method and the IG-M method (the best baseline method combined with our MTS-trained vectors). We filtered out cases where the two methods produced the same explanation. The two results were placed randomly. For each recommendation, the sales staff were asked to choose which set of explanations was more convincing in persuading them to contact the investor. Eleven sales staff participated in the experiment, with each person being presented with 60 options across 6 bonds currently to be issued, each with 10 recommended investors. The results showed that our method was selected 344 times (accounting for 52.1%), and the IG-M method was selected 316 times (accounting for 47.9%), indicating that our method is superior to IG-M from the perspective of sales staff.

5.7 Real-World Application and Evaluation

Our system has been deployed at China Securities Co., Ltd. for over a year and is integrated into the platform that covers the whole process of bond issuance and duration management in the company. Our model is scheduled to update each day to incorporate the up-to-date bidding data. Moreover, we collect the bidding data for each bond after the issuing day and evaluate the recommendation performance of our model using this online data. Although our explanation requires 100 steps of gradient descent for explaining, since the bonds and investors are fixed, we can pre-compute overnight.

We show the *Recall*@50 and *Precision*@50 of our bond RS after deployment in Figure 6, using the average of these metrics on bonds issued within each month. *Recall*@50 reflects the extent to which our RS can discover potential investors, while *Precision*@50 reflects the extent to which the recommended investors will bid

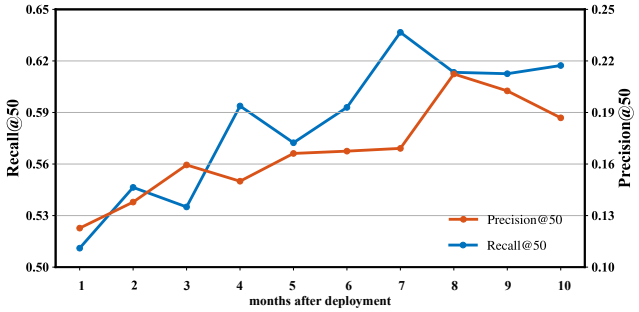


Figure 6: Recommendation performance evolution after deployed in China Securities Co., Ltd.

on the target bond. We can see that, throughout the nearly year-long deployment period, the recommendation performance shows a continuous improvement trend. This could be attributed to the growing tendency for sales staff to adopt the recommendation and inform suggested investors, thereby raising the chances of recommended bonds being known by interested investors. This trend indicates that concise explanations for recommendations might help to establish a beneficial feedback loop within the professional domain, where the results of recommendations positively influence investment actions (the business performance), and by following the recommendation, investment actions lift the recommendation performance in return.

5.8 Limitations

Our explanation method still has some limitations. First, there are still redundant and spurious features selected by our proposed method. This may be attributed to the following three reasons: 1. Features are not identical but are just correlated, thus distinguishing them is difficult. 2. The model trained with MTS still has deviation in estimating the sufficient value. 3. Gradient descent does not guarantee a globally optimal solution of Equation 5. Second, our method may be unsuitable for real-time recommendation applications, as gradient descent could be slow. Addressing this limitation will be a focus of our future research efforts.

6 Conclusion

This paper presents a recommendation system deployed at China Securities Co., Ltd., designed to assist sales staff in the primary bond market with finding investors for bonds. We focus on the feature-based persuasive explanation algorithm. To address spurious and redundancy issues, we propose sufficiency value as the optimization objective, and introduce a mask training strategy combined with a gradient descent method.

7 Acknowledgment

The work was supported by the National Natural Science Foundation of China (No.62206265, No.62076231) and the National Key Research and Development Program of China (No.2022YFB2702502).

References

[1] Krisztian Balog and Filip Radlinski. 2020. Measuring Recommendation Explanation Quality: The Conflicting Goals of Explanations. In *Proc. of SIGIR.*

[2] Umang Bhatt, Adrian Weller, and José MF Moura. 2021. Evaluating and aggregating feature-based model explanations. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence.*

[3] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, Rohan Anil, Zakaria Haque, Lichan Hong, Vihan Jain, Xiaobing Liu, and Hemal Shah. 2016. Wide & Deep Learning for Recommender Systems. In *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems.*

[4] J. Cohen. 1968. Weighted kappa: nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological bulletin* (1968).

[5] Julien Delaunay, Luis Galárraga, and Christine Largouët. 2020. Improving Anchor-based Explanations. In *Proc. of CIKM.*

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proc. of NAACL.*

[7] Gabriel G. Erion, Joseph D. Janizek, Pascal Sturmfels, Scott M. Lundberg, and Su-In Lee. 2021. Improving performance of deep learning models with axiomatic attribution priors and expected gradients. *Nature Machine Intelligence* (2021).

[8] Yingqiang Ge, Juntao Tan, Yan Zhu, Yinglong Xia, Jiebo Luo, Shuchang Liu, Zuohui Fu, Shijie Geng, Zelong Li, and Yongfeng Zhang. 2022. Explainable Fairness in Recommendation. In *Proc. of SIGIR.*

[9] Ehsan Gholami, Mohammad Motamedi, and Ashwin Aravindakshan. 2022. PARSRec: Explainable Personalized Attention-fused Recurrent Sequential Recommendation Using Session Partial Actions. In *Proc. of KDD.*

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In *Proc. of IJCAI.*

[11] Thomas Hofmann. 1999. Probabilistic Latent Semantic Analysis. In *Proc. of UAI.*

[12] Sara Hooker, Dumitru Erhan, Pieter-Jan Kindermans, and Been Kim. 2019. A benchmark for interpretability methods in deep neural networks. In *Proc. of NeurIPS.*

[13] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *Proc. of ICLR.*

[14] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proc. of ICLR.*

[15] Yongchan Kwon and James Y. Zou. 2022. WeightedSHAP: analyzing and improving Shapley based feature attributions. In *Proc. of NeurIPS.*

[16] Jiacheng Li, Zhankui He, Jingbo Shang, and Julian J. McAuley. 2023. UCEpic: Unifying Aspect Planning and Lexical Constraints for Generating Explanations in Recommendation. In *Proc. of KDD.*

[17] Lei Li, Yongfeng Zhang, and Li Chen. 2021. Personalized Transformer for Explainable Recommendation. In *Proc. of ACL.*

[18] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining Explicit and Implicit Feature Interactions for Recommender Systems. In *Proc. of KDD.*

[19] Scott M. Lundberg and Su-In Lee. 2017. A Unified Approach to Interpreting Model Predictions. In *Proc. of NeurIPS.*

[20] Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. 2017. The Concrete Distribution: A Continuous Relaxation of Discrete Random Variables. In *Proc. of ICLR.*

[21] Caio Nóbrega and Leandro Marinho. 2019. Towards explaining recommendations through local surrogate models. In *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing.*

[22] Sebastian Ordyniak, Giacomo Paesani, and Stefan Szeider. 2023. The Parameterized Complexity of Finding Concise Local Explanations. In *Proc. of IJCAI.*

[23] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proc. of KDD.*

[24] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Anchors: High-Precision Model-Agnostic Explanations. In *Proc. of AAAI.*

[25] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proc. of ICML.*

[26] Juntao Tan, Shuyuan Xu, Yingqiang Ge, Yunqi Li, Xu Chen, and Yongfeng Zhang. 2021. Counterfactual Explainable Recommendation. In *Proc. of CIKM.*

[27] Jean-Baptiste Tien, joycenv, and Olivier Chapelle. 2014. Display Advertising Challenge. https://kaggle.com/competitions/criteo-display-ad-challenge. Kaggle.

[28] Sahil Verma, Anurag Beniwal, Narayanan Sadagopan, and Arjun Seshadri. 2022. RecXplainer: Post-Hoc Attribute-Based Explanations for Recommender Systems. In *Progress and Challenges in Building Trustworthy Embodied AI.*

[29] Lei Wang, Ee-Peng Lim, Zhiwei Liu, and Tianxiang Zhao. 2022. Explanation Guided Contrastive Learning for Sequential Recommendation. In *Proc. of CIKM.*

[30] Ruoxi Wang, Rakesh Shivanna, Derek Zhiyuan Cheng, Sagar Jain, Dong Lin, Lichan Hong, and Ed H. Chi. 2021. DCN V2: Improved Deep & Cross Network and Practical Lessons for Web-scale Learning to Rank Systems. In *Proc. of WWW.*

[31] Steve Wang and Will Cukierski. 2014. Click-Through Rate Prediction. https://kaggle.com/competitions/avazu-ctr-prediction. Kaggle.

[32] Jingsen Zhang, Xu Chen, Jiakai Tang, Weiqi Shao, Quanyu Dai, Zhenhua Dong, and Rui Zhang. 2023. Recommendation with Causality enhanced Natural Language Explanations. In *Proceedings of the ACM Web Conference.*

[33] Jieming Zhu, Quanyu Dai, Liangcai Su, Rong Ma, Jinyang Liu, Guohao Cai, Xi Xiao, and Rui Zhang. 2022. BARS: Towards Open Benchmarking for Recommender Systems. In *Proc. of SIGIR.*

A Experiment Detail

A.1 Settings

During the explanation by gradient descent, for Gumbel-Softmax, $z \in [0,1]^N$ is produced by sigmoid(z'), and all entries of z' are initialized with -10. We set $\tau = 5$ as the initial value in Gumbel Softmax and anneal it down to 0 during optimization.

A.2 Understanding P-reason and Its Rationality

Table 6: All investor preference features and their corresponding bond features.

Investor Preference Feature	Bond Feature
monthly preference level for 1 year bond yearly preference level for 2-3 year bond ...	period
monthly preference level for AAA subject yearly preference level for AA+ subject ...	subject rating
monthly preference level for AAA bond yearly preference level for AA+ bond ...	debt rating
monthly preference level for commercial paper yearly preference level for corporate bonds ...	type
monthly preference level for renewable bond yearly preference level for unrenewable bond	renewable
monthly preference level for finance yearly preference level for material industry ...	issuer industry
monthly preference level for issuer in BeiJing yearly preference level for issuer in GuangDong ...	issuer province
monthly preference level for central state-owned co. yearly preference level for local state-owned co. ...	issuer property
monthly preference level for non-financial enterprise yearly preference level for commercial bank ...	issuer type
monthly preference level for unlisted company yearly preference level for listed company	issuer is listed

We group investor preference features into 10 types, each of which corresponds to a bond feature. The details can be seen in Table 6. For each investor preference feature r in generated explanations, we say it reflects the investor's preference towards a specific type of bond (i.e. it is a P-reason) if the preference level of r is above

Table 8: Explanation quality of Model Trained without MTS.

Method	Rationality	Diversity	HRD
LIME	0.323	0.581	0.415
Kernel SHAP	0.532	0.625	0.574
WSHAP	0.491	0.651	0.559
IG	0.519	0.707	0.598
EG	0.557	0.611	0.593
RecXPlainer-L	0.024	0.755	0.046
RecXPlainer-M	0.230	0.435	0.300
Ours (with MTS)	0.695	0.701	0.697

Table 7: Influence of Mask Training Strategy (MTS) on recommendation performance.

Model	Avazu		Criteo		CSC	
	AUC	Loss	AUC	Loss	AUC	Loss
DNN	0.7630	0.3682	0.8137	0.4381	0.8958	0.0837
+ MTS	0.7633	0.3675	0.8135	0.4385	0.8983	0.0833
W&D	0.7649	0.3665	0.8139	0.4380	0.8980	0.0823
+ MTS	0.7636	0.3675	0.8134	0.4385	0.9037	0.0793
DeepFM	0.7648	0.3667	0.8138	0.4381	0.9008	0.0857
+ MTS	0.7633	0.3676	0.8132	0.4386	0.9019	0.0794
DCN-v2	0.7656	0.3664	0.8142	0.4378	0.9009	0.0797
+ MTS	0.7644	0.3667	0.8140	0.4380	0.9038	0.0791

the average preference level of all investors. We decide whether a P-reason is rational by the target bond's corresponding feature.

A.3 Generating Topic Vectors for Investor Features via PLSA

Analogous to the scenario of using PLSA in document modeling, each sample is treated as a document where the investor features are the words in our experiment. The features of investors constitute the vocabulary. We set the number of topics to 50. The topic distribution for a word (i.e., investor feature) is used as the representation of the word in the topic space.

A.4 Influence of Mask Training Strategy on Recommendation Performance

To show the influence of MTS on recommendation performance, we conduct an ablation study in our dataset (CSC) and two public datasets (Avazu [31] and Criteo [27]). We test four well-known models: DNN, Wide&Deep [3], DeepFM [10], and DCN-V2 [30].

For experiments on the Avazu and Criteo datasets, we follow the implementation of BARS [33], which is an open benchmark for the recommendation system. Specifically, we use Avazu_x1 and Criteo_x1, where x1 means a specific way for data splitting and preprocessing in BARS. Since the features of these two datasets are anonymous, we apply MTS to all features of these datasets. Following previous studies [3, 10, 18, 30], we use AUC and LogLoss for evaluation. Table 7 shows the results. We can see that MTS has a slight impact on the performance of deep CTR models.

A.5 Explanation Quality on Model Trained without MTS

Table 8 shows the quality of explanations generated by baselines on the model trained without MTS. Generally, we observe that the model trained without MTS can generate more diverse but much less rational reasons compared with the model trained with MTS (refer to Table 3). All of these methods achieve HRD scores lower than our method.

B Minimizing L_{MTS} Gives Both Recommendation and Explanation Models

Here, we proof that minimizing L_{MTS} gives a model that can be used to approximate $P(Y=1 \mid \boldsymbol{x})$ for recommendation and $P(Y=1 \mid \boldsymbol{x}_S)$ for explanation. Before presenting the proof, we introduce essential notations. Let $X = \{X_1, X_2, \ldots, X_N\}$ denote the set of random variables corresponding to investor features, and let Y represent the random variable that signifies whether an investor bids on the bond. The specific instances of X, Y are denoted by lowercase $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$, y respectively. Additionally, we define a mask vector $\mathbf{m} = \{m_1, m_2, \ldots, m_N\}$, where $m_i \in \{0, 1\}$ means whether the i-th investor feature is masked. We denote $X_{\mathbf{m}} \triangleq \{X_i | 1 \le i \le N \wedge m_i = 0\}$ and $\mathbf{x}_{\mathbf{m}} \triangleq \{x_i | 1 \le i \le N \wedge m_i = 0\}$

Proposition B.1. *L_{MTS} is minimal w.r.t. $f_m(\mathbf{x}, \mathbf{m}; \theta)$ only if $f_m(\mathbf{x}, \mathbf{m}; \theta) = p(Y = 1 | X_{\mathbf{m}} = \mathbf{x}_{\mathbf{m}})$*

Proof.

$$
\begin{aligned}
&\frac{\partial L_{MTS}}{\partial f_m(\mathbf{x}, \mathbf{m}; \theta)} \\
&= \frac{\partial \mathbb{E}_{X,Y\sim D}\mathbb{E}_{\mathbf{m}'} CE(f_m(X, \mathbf{m}'; \theta), Y)}{\partial f_m(\mathbf{x}, \mathbf{m}; \theta)} \\
&= \mathbb{E}_{X,Y\sim D}\left[\mathbb{I}(X_{\mathbf{m}'} = \mathbf{x}_{\mathbf{m}}) p(\mathbf{m}' = \mathbf{m}) \frac{\partial CE(f_m(\mathbf{x}, \mathbf{m}; \theta), Y)}{\partial f_m(\mathbf{x}, \mathbf{m}; \theta)}\right] \\
&= p(\mathbf{m}' = \mathbf{m}) \mathbb{E}_{X,Y\sim D}\left[\mathbb{I}(X_{\mathbf{m}'} = \mathbf{x}_{\mathbf{m}}) \frac{\partial CE(f_m(\mathbf{x}, \mathbf{m}; \theta), Y)}{\partial f_m(\mathbf{x}, \mathbf{m}; \theta)}\right] \\
&= p(\mathbf{m}' = \mathbf{m}) \mathbb{E}_{X,Y\sim D}\left[\mathbb{I}(X_{\mathbf{m}'} = \mathbf{x}_{\mathbf{m}}) \frac{f_m(\mathbf{x}, \mathbf{m}; \theta) - Y}{f_m(\mathbf{x}, \mathbf{m}; \theta)(1 - f_m(\mathbf{x}, \mathbf{m}; \theta))}\right] \\
&= \frac{p(\mathbf{m}' = \mathbf{m})}{f_m(\mathbf{x}, \mathbf{m}; \theta)(1 - f_m(\mathbf{x}, \mathbf{m}; \theta))} \times \\
&\quad \mathbb{E}_{X,Y\sim D}\left[\mathbb{I}(X_{\mathbf{m}'} = \mathbf{x}_{\mathbf{m}})(f_m(\mathbf{x}, \mathbf{m}; \theta) - Y)\right] \\
&= \frac{p(\mathbf{m}' = \mathbf{m})}{f_m(\mathbf{x}, \mathbf{m}; \theta)(1 - f_m(\mathbf{x}, \mathbf{m}; \theta))} \times \\
&\quad [f_m(\mathbf{x}, \mathbf{m}; \theta) p(X_{\mathbf{m}} = \mathbf{x}_{\mathbf{m}}) - p(X_{\mathbf{m}} = \mathbf{x}_{\mathbf{m}}, Y = 1)]
\end{aligned} \tag{9}
$$

If L_{MTS} is minimal w.r.t. $f_m(\mathbf{x}, \mathbf{m}; \theta)$ then $\frac{\partial L_{MTS}}{\partial f_m(\mathbf{x}, \mathbf{m}; \theta)} = 0$. Thus,

$$
\begin{aligned}
&\frac{\partial L_{MTS}}{\partial f_m(\mathbf{x}, \mathbf{m}; \theta)} = 0 \\
\Leftrightarrow &f_m(\mathbf{x}, \mathbf{m}; \theta) p(X_{\mathbf{m}} = \mathbf{x}_{\mathbf{m}}) - p(X_{\mathbf{m}} = \mathbf{x}_{\mathbf{m}}, Y = 1) \\
\Leftrightarrow &f_m(\mathbf{x}, \mathbf{m}; \theta) = p(Y = 1 | X_{\mathbf{m}} = \mathbf{x}_{\mathbf{m}})
\end{aligned} \tag{10}
$$

□

Table 9: Sufficient value of different feature subsets.

$\|S\|$	x_S	$P(Y=1\|\boldsymbol{x}_S)$	$f(\boldsymbol{x}_S)$
1	***{price1 = low}***	**0.86**	0.860
	{price2 = low}	**0.86**	0.860*
	{*quality = low*}	0.77	0.768
	{*color = blue*}	0.55	0.612
2	***{price1 = low, color = blue}***	**0.91**	0.888
	{price2 = low, color = blue}	**0.91**	0.880*
	{*price1 = low, price2 = low*}	0.86	0.873
	{*price1 = low, quality = low*}	0.85	0.867
	{*price2 = low, quality = low*}	0.85	0.881
	{*quality = low, color = blue*}	0.82	0.797
3	***{price1 = low, price2 = low, color = blue}***	**0.91**	0.904*
	{*price1 = low, quality = low, color = blue*}	0.90	0.897
	{*price2 = low, quality = low, color = blue*}	0.90	0.902
	{*price1 = low, price2 = low, quality = low*}	0.85	0.871

Remark 1. *Based on proposition B.1, if L_{MTS} is minimal w.r.t. $f_m(\mathbf{x}, \mathbf{m}; \theta)$, then we have*

$$\begin{aligned} f_m(\mathbf{x}, \mathbf{0}; \theta) &= p(Y=1|X=\mathbf{x}) = p(y=1|\mathbf{x}) \\ f_m(\mathbf{x}, \mathbf{m}_S; \theta) &= p(Y=1|X_{\mathbf{m}_S} = \mathbf{x}_{\mathbf{m}_S}) = p(y=1|\mathbf{x}_S), \end{aligned} \tag{11}$$

where $\mathbf{m}_S = [\mathbb{I}(i \in S)|1 \le i \le N]$ is the corresponding mask vector for the chosen set S, which means $f_m(\mathbf{x}, \mathbf{0}; \theta)$ can be used for recommendation, and $f_m(\mathbf{x}, \mathbf{m}_S; \theta)$ can be used to compute the sufficiency score for explanation.

C Details and Discussion on the Toy Example

Table 9 shows the proposed sufficient value $P(Y=1|\boldsymbol{x}_S)$ and model estimation $f(\boldsymbol{x}_S)$ for each feature subset S. The subsets are shown in 3 groups according to $|S|$. In each group, the feature subsets that achieve the highest scores are highlighted in bold.

We first look at sufficient values. When $|S| = 1$, *price1=low* and *price2=low* obtain the highest score of 0.86. When $|S| = 2$, *{price1=low,color=blue}* and *{price2=low,color=blue}* obtain the highest score of 0.91. In this case, we find $P(Y=1|price_1 = low, color = blue) > P(Y=1|price_1 = low, price_2 = low) = P(Y=1|price_1 = low)$. This reflects that selecting a redundant feature $price_2 = low$ into $\{price_1 = low\}$ cannot increase the sufficiency value while selecting another feature $color = blue$ does. More generally, suppose two features a and a' are redundant to each other. For a recommended item, if a feature set S contains a but not a', then $P(Y=1|\boldsymbol{x}_S)$ would be the same as $P(Y=1|\boldsymbol{x}_{S\cup\{a'\}})$. And, it is possible to substitute a' with another feature to increase $P(Y=1|\boldsymbol{x}_S)$ further. Thus, explaining based on sufficient value can avoid redundant features.

And for 3-feature subsets, *{price1=low, price2=low, color=blue}* obtain the highest score of 0.91. In this case, $P(Y=1|price_1 = low) > P(Y=1|price_1 = low, quality = low)$, reflecting that adding *quality=low* into *{price=low}* will decrease the sufficiency value. In other words, our method effectively filters out the spurious feature if the "real" reason behind the spurious feature has already been in the current candidate feature set.

Then, we look at the estimated value $f(\boldsymbol{x}_S)$ in the column. They are similar to $P(Y=1|\boldsymbol{x}_S)$ in values as well as in rankings, demonstrating our method can correctly estimate the sufficient value. For each $|S|$ group, we mark the subset selected by gradient descent with a star beside the corresponding $f(\boldsymbol{x}_S)$ value. The selected explanations are as expected, demonstrating our method can find the best subset via gradient descent.