G-STO: Sequential Main Shopping Intention Detection via Graph-Regularized Stochastic Transformer

Yuchen Zhuang*
yczhuang@gatech.edu
Georgia Tech
Atlanta, GA, USA

Xin Shen
xinshen@amazon.com
Amazon
Seattle, WA, USA

Yan Zhao
yzhaoai@amazon.com
Amazon
Seattle, WA, USA

Chaosheng Dong
chaosd@amazon.com
Amazon
Seattle, WA, USA

Ming Wang
mingww@amazon.com
Amazon
New York, NY, USA

Jin Li
jincli@amazon.com
Amazon
Seattle, WA, USA

Chao Zhang
chaozhang@gatech.edu
Georgia Tech
Atlanta, GA, USA

ABSTRACT

Sequential recommendation requires understanding the dynamic patterns of users' behaviors, contexts, and preferences from their historical interactions. While most research emphasizes item-level user-item interactions, they often overlook underlying shopping intentions, such as preferences for ballpoint pens or miniatures. Identifying these latent intentions is vital for enhancing shopping experiences on platforms like Amazon. Despite its significance, the area of main shopping intention detection remains under-investigated in the academic literature. To fill this gap, we introduce a graph-regularized stochastic Transformer approach, G-STO. It considers intentions as product sets and user preferences as intention composites, both modeled as stochastic Gaussian embeddings in latent space. We also employ a global intention relational graph as prior knowledge for regularization, ensuring related intentions are distributionally close. These regularized embeddings are then input into Transformer-based models to capture sequential intention transitions. On testing our model with three real-world datasets, it outperformed the baselines by 18.08% in Hit@1, 7.01% in Hit@10, and 6.11% in NDCG@10.

CCS CONCEPTS

• **Information systems** → **Personalization**.

KEYWORDS

Personalization; Main Shopping Intention Detection

ACM Reference Format:
Yuchen Zhuang, Xin Shen, Yan Zhao, Chaosheng Dong, Ming Wang, Jin Li, and Chao Zhang. 2023. G-STO: Sequential Main Shopping Intention Detection via Graph-Regularized Stochastic Transformer. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3583780.3614890

*Work done during the author's internship at Amazon.

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10.
https://doi.org/10.1145/3583780.3614890

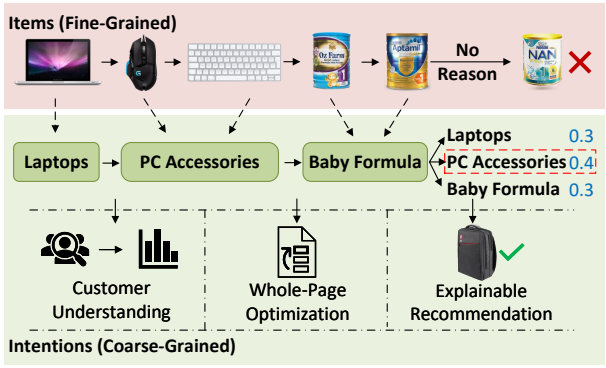


Figure 1: Illustration of sequential item recommendation and main shopping intention identification. For the majority of sequential recommendation algorithms, recommendations are provided without specific reasons, which can be explained by the main shopping intention identification.

1 INTRODUCTION

Sequential recommendation, which aims at understanding evolving customer behaviors and dynamically recommending new items, has garnered considerable interests. E-commerce stores, *e.g.*, Amazon, have adopted customers' *shopping intentions* signals into their recommendation systems [11]. Although such initial works were built with a heuristic shopping intention detection approach, they already achieved significant improvement in terms of the *MOI* metric (metric of interest), which considers both the short- and long-term effects on the customers' shopping experience. Besides product-level recommendations, whole-page optimization, another downstream application, can also benefit from such customer intention signals. Recent work [17] has also achieved significant lift of MOI on retail website homepage and checkout page through online A/B experimentation. Thus, a robust, scalable, and explainable shopping intention detection approach plays crucial roles in multiple stages of the recommendation pipeline.

However, the majority of existing sequential recommendation algorithms [5, 18, 37, 38] focus solely on product-level data to predict subsequent recommendation items, regardless of the underlying shopping intentions. This leaves these approaches inadequate in

capturing whole-sequence patterns, as they tend to place more emphasis on the item transitions learned from training data than on comprehending the customers' underlying objectives. For example, when a user's interaction sequence is provided as in Figure 1, most sequential recommendation systems will emphasize the commonly-seen transition pair, (`baby formula`→`baby formula`), to recommend subsequent products without explanation. In contrast, if we can identify the main shopping intention, we will find out that `PC Accessories` should be the most important intention for this customer and recommend products accordingly.

To this end, some approaches [2, 23] start to take both product-level and intention-level data into account, but merely use the concealed main shopping intentions as implicit guidance for the following item recommendation. However, these implicit guidances can be severely influenced by the product-level interactions, making popular item transitions dominate the main intention detection. Thus, explicitly identifying the customers' main shopping intentions to infer the user preferences becomes a prerequisite for explainable recommendations and user understanding.

To accomplish the main shopping intention identification task, the most typical and direct method is to map product-level interactions to intention-level sequences and then apply sequential recommendation algorithms. Among the existing methods, recent advancements in Transformer [18, 24, 37] introduce the self-attention mechanism to reveal the position-wise item-item relations, which have achieved state-of-the-art performances. Despite their success in product-level sequential recommendations, we argue that simply adapting such embedding-based Transformer architectures to intention-level sequences fail to incorporate: (1) *the shopping intention characteristics*: Shopping intentions are higher-level taxonomy, which can be considered as sets of products. Using only deterministic embeddings to represent shopping intentions is insufficient to capture this high-level characteristic; (2) *the user preferences composed of multiple intentions*: Users can have multiple intentions and preferences in mind during a shopping journey. Using the Transformer architecture to characterize user preferences as deterministic points is also insufficient for estimating the relevance between user preferences and a composition of multiple shopping intentions; (3) *the collaborative transitivity*: Collaborative transitivity indicates the ability of introducing additional collaborative relevance beyond constrained intention transition pairs. Transformer architectures employ dot-product-based attention mechanism, which is difficult to infer the relevance across pairs, (*e.g.*, using a and b, b and c pairs to infer a and c are relevant as well); (4) *dynamic uncertainty*: In customer interaction sequences, it is common to witness a significant random shift in preferences without obvious correlations across intention transitions. Customers with more interests in dynamic variability are intuitively more uncertain. Therefore, when modeling user preferences, dynamic uncertainty is a vital component.

To this end, we present a new graph-regularized stochastic Transformer framework, G-STO, for main shopping intention identification. Our approach overcomes the aforementioned challenges with the following key designs: (1) To better incorporate collaborative transitivity and uncertainty into representations, we describe each intention as an elliptical Gaussian distribution. Specifically, G-STO applies stochastic embedding layers to assign each intention a mean and covariance embedding, composing the stochastic representations; (2) To transfer knowledge from popular intentions to unpopular ones for cold-start issue resolution, we introduce global intention relation information as prior knowledge for improved intention modeling. Specifically, we design an intention relation graph for regularization, with diverse intentions as nodes and complementary/relevant relations between intentions as edges. By propagating intention representations on the graph, relevant shopping intentions are dragged towards each other on latent representation space to share close embedding distributions. It can also alleviate the data scarcity issue for unpopular intentions, whose embeddings can be inferred from their neighbors on the graph. (3) Once we obtain the regularized stochastic representations for the users' interactions with intentions, we send them to mean and covariance Transformers to model the sequential information from intention transitions. Instead of using dot product to compute relevance score between user preferences and recommendations in deterministic models, we apply Wasserstein distance to measure the distances between distributions. Considering the distances as dissimilarity between intentions with uncertainty information, we combine it with Bayesian Personalize Ranking (BPR) loss [30] as the training objective.

Our contributions can be summarized as follows:

- To the best of our knowledge, this is the first work focusing on the main shopping intention identification task with solely intention-level data. This will help with user understanding and improve the performances of downstream tasks, including product-level recommendation, ranking stage of whole page optimization;
- We describe the intentions as Gaussian distributions using stochastic representations to reflect the high-level properties of intentions, collaborative transitivity across intentions, and user uncertainty;
- We introduce the shopping intention relation graph as prior knowledge and propose a novel graph regularizer to restrict the stochastic representations in distribution-based methods;
- We develop three different Amazon real-world datasets, covering long-term, short-term, and purchase-related user cases. Our proposed G-STO outperforms state-of-the-art baselines significantly by 18.08% in Hit@1, 6.11% in NDCG@10, and 7.01% in Hit@10 on average of three datasets.

2 RELATED WORKS

2.1 Sequential Recommendation (SR)

Sequential Recommendation (SR) aims to predict the next item based on the user historical interactions with products. Earlier SR works, like FPMC [31] and Fossil [13], apply Markov Chains with matrix factorization to model the first-order and higher-order item-to-item transition matrices. However, when encountering with long interaction sequences or considering long-term influence from previous items, the computations of modeling the transition matrices increase exponentially. More recent sequential recommendations can better learn item-to-item transition patterns automatically via deep learning architectures, including Convolution Neural Networks (CNN) [38], Recurrent Neural Networks (RNN) [7, 21, 25, 27, 29] and self-attentive models [18, 37, 42]. Among them, the Transformer-based models reach the state-of-the-art performances with the capability of extracting context information from all past actions and learn short-term dynamics. However,

they still struggle to solve the cold-start issue [45], which means for unpopular items, the representations are under-trained and it is difficult for models to make precise predictions over them without sufficient data. In addition, despite the fact that these models have been successful in sequential recommendation systems and can be directly transferred to the main shopping intention identification task, they are incapable of capturing the unique characteristics of intentions, as intentions are higher-level concepts than items.

2.2 Intention Identification in SR

With the development of recommendation systems, people start to seek for other side information to help improve the recommendation qualities. Shopping intentions usually serve as a coarse-grained side information that can better describe users' preferences. Most of the existing methods [23, 24, 36, 39, 46, 49] focus on treating the intentions as an implicit guidance or an additional feature for downstream product-level recommendation. However, the intention guidance learned from the sequences may not be in line with expectation. As they are using the same model architecture on both intention and item sequences, the main shopping intention can be misled by some commonly seen yet totally irrelevant item pairs. To resolve this issue, some other methods leverage clustering algorithms [2] or graph neural networks [4, 6, 41] to better understand the intentions via linking them with items and users. However, these approaches lose the explainability of shopping intentions for better user understanding. Another line of works identify the shopping intentions from search queries input by the users [10, 12, 22, 44], which is orthogonal to our work.

2.3 Stochastic Representations in SR

Representing concepts (*e.g.*, natural language sequences, images, graphs) as distributions has attracted interests from the research community [1, 14, 20, 28, 33–35]. Most stochastic models represent the concepts with Gaussian distributions, composed of mean and covariance. The distribution representations introduce uncertainties and provide more flexibility compared with deterministic embeddings. In the recommendation systems area, a few studies propose to leverage the advantages of Gaussian distributions to flexibly represent users and items. For example, GeRec [16] models each user and item as a Gaussian distribution and applies CNN on the sampled matrix from their distributions for inference. To dynamically monitor the sequential changes in user interactions, a series of work [8, 9, 48] also combine Gaussian distributions with sequential recommendation algorithms. DT4SR [8] is one of these attempts, proposing the mean and covariance embeddings to model items as distributions. STOSA [9] extends DT4SR architecture via proposing a new stochastic attention mechanism based on Wasserstein distance. However, all the previous stochastic methods, learning the distributions only from the transitions within the sequence, can still fail in cold-start situations in § 2.1, which can be mitigated by G-STO. In addition, none of the aforementioned methods naturally combine stochastic representations with intentions to learn the intention distributions, which is accomplished by G-STO as well. Another methodology line is variational autoencoder (VAE), approximating posterior distributions of latent variables via variational inference. Combining SR and VAE, SVAE [32] and VSAN [47] learn the dynamic hidden representations of shopping sequences. However, these efforts still perform worse than existing deterministic models on many tasks [18, 37]. As an update, ACVAE [43] incorporates adversarial variational Bayes and maximization of mutual information between user embeddings and input sequences to obtain more distinctive and personalized representations of individual users. However, all these VAE-based SR methods are easy to suffer from posterior collapse problems, generating poor-quality latent representations, which G-STO can avoid.

3 METHOD

In this section, we introduce our proposed graph regularized stochastic model, G-STO, for main shopping intention identification (§ 3.1). Figure 2 displays the working flow of G-STO. It consists of several key components: (1) stochastic representations (§ 3.2), which model the shopping intentions as stochastic distributions, consisting of mean and covariance embeddings; (2) intention relation regularizer (§ 3.3), which creates an intention relation graph and regularize more relevant intentions on the graph to have closer stochastic representations; (3) mean and covariance Transformers (§ 3.4), encoding sequential information from the transition patterns in user historical interactions to generate stochastic representations of user preferences; (4) Wasserstein distance (§ 3.5), which measures the dissimilarity between intentions and user preferences and can be combined with Bayesian Personalized Ranking (BPR) [30] loss on the positive and negative sequences for model training.

3.1 Problem Definition

A sequential recommendation system collects the interactions between a set of users $\mathcal{U}$ and items $\mathcal{V}$, (*e.g.*, clicks, purchases, *etc.*) and sorts them chronologically into sequences. Similarly, to identify users' main shopping intentions, models need to map all the items $\mathcal{V}$ to their belonging shopping intentions $\mathcal{M}$ and reorganize them chronically as $\mathcal{S}^u = [m_1^u, m_2^u, \cdots, m_{|\mathcal{S}^u|}^u]$. The goal of the main shopping intention identification can be formulated as:

$$p(m^u = m|\mathcal{S}^u), \tag{1}$$

which measures the probability of an intention m being the main shopping intention m^u given user u's sequence.

3.2 Stochastic Embedding Layers

Different from the deterministic embedding layers that only map the items/intentions to unique high-dimensional vectors, stochastic embedding layers formulate the intentions as high-dimensional elliptical Gaussian distributions. These Gaussian distributions are constructed of mean and covariance embeddings, spanning a broader space to include more high-dimensional points, which naturally captures high-level semantics of shopping intentions. Specifically, we define a mean embedding table $\mathbf{T}^\mu \in \mathbb{R}^{|\mathcal{M}|\times d}$ and a covariance embedding table $\mathbf{T}^\Sigma \in \mathbb{R}^{|\mathcal{M}|\times d}$, where $|\mathcal{M}|$ denotes the total number of shopping intentions and d denotes the hidden dimension size. As the mean and covariance of a Gaussian distribution identify different signals of expectation and uncertainty, we introduce different position embeddings $\mathbf{P}^\mu \in \mathbb{R}^{L\times d}$ and $\mathbf{P}^\Sigma \in \mathbb{R}^{L\times d}$, where L indicates the sequence length. Thus, we can obtain the stochastic representations via the summation of the previously defined

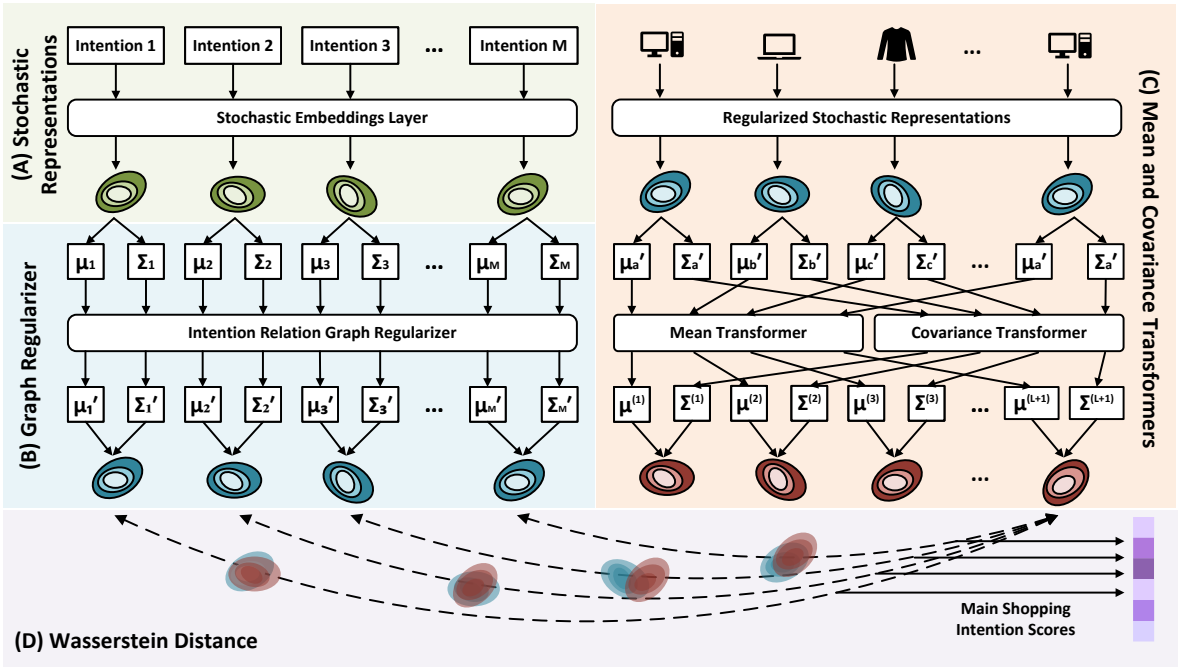


Figure 2: Illustration of G-STO model, containing key components: (A) stochastic representations to map each intention as a Gaussian distribution; (B) graph regularizer to restrict the more relevant intentions to have closer representations; (C) mean and covariance transformers to encode the sequential information from user historical interactions; (D) Wasserstein distance for training and inference.

embeddings and position embeddings:

$$\begin{aligned}\mathbf{E}^{\mu}_{\mathcal{S}^u} &= [\mathbf{E}^{\mu}_{s_1}, \cdots, \mathbf{E}^{\mu}_{s_L}] = [\mathbf{T}^{\mu}_{s_1} + \mathbf{P}^{\mu}_{s_1}, \cdots, \mathbf{T}^{\mu}_{s_L} + \mathbf{P}^{\mu}_{s_L}],\\ \mathbf{E}^{\Sigma}_{\mathcal{S}^u} &= [\mathbf{E}^{\Sigma}_{s_1}, \cdots, \mathbf{E}^{\Sigma}_{s_L}] = [\mathbf{T}^{\Sigma}_{s_1} + \mathbf{P}^{\Sigma}_{s_1}, \cdots, \mathbf{T}^{\Sigma}_{s_L} + \mathbf{P}^{\Sigma}_{s_L}].\end{aligned} \tag{2}$$

For example, the first intention s_1 in the sequence $\mathcal{S}^u$ is formulated as a Gaussian distribution $\mathcal{N}(\mu_{s_1}, \Sigma_{s_1})$, where $\mu_{s_1} = \mathbf{E}^{\mu}_{s_1}$ and $\Sigma_{s_1} = \text{diag}(\text{elu}(\mathbf{E}^{\Sigma}_{s_1}) + \mathbf{1}) \in \mathbb{R}^{d\times d}$. $\text{elu}(\cdot)$ is the ELU activation function and $\mathbf{1} \in \mathbb{R}^d$ is an all-ones vector. [1]

3.3 Intention Relation Graph Regularizer

To effectively model infrequent and under-trained shopping intentions, our goal is to utilize the most related intentions to help the model comprehend them. To this end, we propose a novel graph-based regularizer, allowing more pertinent shopping intentions to share closer stochastic representations. Thus, we introduce the global intention relationship as the prior knowledge and create a graph accordingly, which can be introduced as follows:

Intention Relation Graph. We create the intention relation graph with the aid of P-Companion [11]. Given a pair of shopping intentions (m_i, m_j) as inputs, we treat the co-purchase relations between them as distant supervised labels $y_{i,j} = \{+1, -1\}$. To extract relevant information from co-purchase relations, each shopping intention and its complementary side are represented by two separate embeddings, ϕ_w, ϕ^c_w. Thus, to infer the relation between shopping intentions (m_i, m_j), we first apply a 2-layer feed-forward network (FFN) to transform m_i:

$$\gamma_{m_i} = \text{ReLU}(\phi_{m_i}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2, \tag{3}$$

where $\{\mathbf{W}_1, \mathbf{W}_2, \mathbf{b}_1, \mathbf{b}_2\}$ are trainable parameters. Then, we optimize the relation between the transformed embedding of m_i, γ_{m_i}, and the complementary embedding of m_j, $\phi^c_{m_j}$, via applying hinge loss function on labels $y_{i,j}$:

$$\ell = \min \sum_{m_i, m_j \in \mathcal{M}} (\max\{0, \epsilon - y_{i,j}(\lambda - \|\gamma_{m_i} - \phi^c_{m_j}\|_2^2\}), \tag{4}$$

where λ is the base distance to distinguish γ_{m_i} and $\phi^c_{m_j}$, and ϵ is the margin distance. When $y_{i,j} = 1$, the two shopping intentions have the co-purchase/complementary relation and the model will force the distance between γ_{m_i} and $\phi^c_{m_j}$ to be smaller than $\lambda - \epsilon$. Otherwise, when $y_{i,j} = -1$, the two shopping intentions do not possess the complementary relation, pushing the γ_{m_i} and $\phi^c_{m_j}$ far away from each other with distance more than $\lambda + \epsilon$. With the trained ϕ_{m_i} and $\phi^c_{m_i}$, we can create the shopping intention relation graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where the nodes $\mathcal{V}$ are different shopping intentions, and the edges $\mathcal{E}$ indicate the relevant/complementary scores between the intentions. To compute edge weights between intentions m_i and m_j, we apply cosine similarities between m_i's complementary embedding, $\phi^c_{m_i}$, and m_j's embedding, ϕ_{m_j}:

$$e(m_i, m_j) = \frac{\phi^c_{m_i} \cdot \phi_{m_j}}{\|\phi^c_{m_i}\| \|\phi_{m_j}\|}. \tag{5}$$

[1] The operations on covariance embeddings are designed to guarantee the covariance matrix to be positive definite.

Graph Neural Regularizer. Considering the previously constructed relational graph of intentions as prior knowledge, we aim to regularize the more relevant intentions on the graph so that they share closer distributions. When determining the main shopping intentions of a user, we can rank all relevant intentions higher, including those even unpopular ones, if they are represented similarly. Thus, we employ the graph convolutional network (GCN) to induce stochastic representations of nodes based on the neighboring features, transferring knowledge to under-trained nodes from their frequently-seen neighbors. To learn a unified set of parameters regularizing both the mean embeddings $\mathbf{T}^{\mu} \in \mathbb{R}^{|\mathcal{M}|\times d}$ and the covariance embeddings $\mathbf{T}^{\Sigma} \in \mathbb{R}^{|\mathcal{M}|\times d}$ simultaneously, we concatenate them together as the initial node representations:

$$\mathbf{X}^{(0)} = \mathbf{T}^{\mu} \oplus \mathbf{T}^{\Sigma} \in \mathbb{R}^{|\mathcal{M}|\times 2d}. \tag{6}$$

Then, the GCN propagation can be represented as follows:

$$\mathbf{X}^{(l)} = \sigma(\tilde{\mathbf{A}}\mathbf{X}^{(l-1)}\mathbf{W}^{(l-1)}), \tag{7}$$

where $\mathbf{W}^{(l-1)} \in \mathbb{R}^{2d\times 2d}$ is the trainable weight matrix after the l-th layer, and $\mathbf{X}^{(l)} \in \mathbb{R}^{|\mathcal{M}|\times 2d}$ indicates the obtained new regularized intention representations at the l-th layer. $\tilde{\mathbf{A}} \in \mathbb{R}^{|\mathcal{M}|\times|\mathcal{M}|}$ is the normalized format of the adjacent matrix $\mathbf{A} \in \mathbb{R}^{|\mathcal{M}|\times|\mathcal{M}|}$ derived from graph $\mathcal{G}$, which can be computed via $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{D}$ is the degree matrix [2] of $\mathbf{A}$. Once obtain the regularized intention representations $\mathbf{X}^{(l)} \in \mathbb{R}^{|\mathcal{M}|\times 2d}$ at the l-th layer, we will separate the new mean embeddings $\hat{\mathbf{T}}^{\mu} \in \mathbb{R}^{|\mathcal{M}|\times d}$ and the new covariance embeddings $\hat{\mathbf{T}}^{\Sigma} \in \mathbb{R}^{|\mathcal{M}|\times d}$ from them to form the regularized Gaussian distributions with intention relations:

$$[\hat{\mathbf{T}}^{\mu} \oplus \hat{\mathbf{T}}^{\Sigma}] = \mathbf{X}^{(l)}. \tag{8}$$

Thus, we can finally rewrite the embedding layer in Eq.(2) to encode the user shopping sequences with regularized distributions:

$$\begin{aligned}\hat{\mathbf{E}}^{\mu}_{\mathcal{S}^u} &= [\hat{\mathbf{E}}^{\mu}_{s_1}, \cdots, \hat{\mathbf{E}}^{\mu}_{s_L}] = [\hat{\mathbf{T}}^{\mu}_{s_1} + \hat{\mathbf{P}}^{\mu}_{s_1}, \cdots, \hat{\mathbf{T}}^{\mu}_{s_L} + \hat{\mathbf{P}}^{\mu}_{s_L}],\\ \hat{\mathbf{E}}^{\Sigma}_{\mathcal{S}^u} &= [\hat{\mathbf{E}}^{\Sigma}_{s_1}, \cdots, \hat{\mathbf{E}}^{\Sigma}_{s_L}] = [\hat{\mathbf{T}}^{\Sigma}_{s_1} + \hat{\mathbf{P}}^{\Sigma}_{s_1}, \cdots, \hat{\mathbf{T}}^{\Sigma}_{s_L} + \hat{\mathbf{P}}^{\Sigma}_{s_L}].\end{aligned} \tag{9}$$

3.4 Mean and Covariance Transformers

Apart from the prior knowledge we obtain from the global intention relation information, we still need to encode the sequential information from the user historical interaction sequences. Therefore, we propose mean and covariance Transformers to automatically learn the hidden patterns from the intention transitions in the sequences. The deterministic Transformer-based models build up the self-attention mechanisms with the dot products between query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$. In sequential recommendation, the query, key, and value are obtained from the linear transformations of the same sequence embedding $\hat{\mathbf{E}}_{\mathcal{S}^u}$. However, in distribution-based stochastic models, we use mean and covariance embeddings to form a Gaussian distribution as intention and sequence representations. Thus, we need two separate sets of $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ for both mean and covariance embeddings of the sequence:

$$\begin{aligned}&\mathbf{Q}_{\mu}(\mathcal{S}^u) = \hat{\mathbf{E}}^{\mu}_{\mathcal{S}^u}\mathbf{W}^{Q}_{\mu},\ \mathbf{K}_{\mu}(\mathcal{S}^u) = \hat{\mathbf{E}}^{\mu}_{\mathcal{S}^u}\mathbf{W}^{K}_{\mu},\ \mathbf{V}_{\mu}(\mathcal{S}^u) = \hat{\mathbf{E}}^{\mu}_{\mathcal{S}^u}\mathbf{W}^{V}_{\mu};\\ &\mathbf{Q}_{\Sigma}(\mathcal{S}^u) = \hat{\mathbf{E}}^{\Sigma}_{\mathcal{S}^u}\mathbf{W}^{Q}_{\Sigma},\ \mathbf{K}_{\Sigma}(\mathcal{S}^u) = \hat{\mathbf{E}}^{\Sigma}_{\mathcal{S}^u}\mathbf{W}^{K}_{\Sigma},\ \mathbf{V}_{\Sigma}(\mathcal{S}^u) = \hat{\mathbf{E}}^{\Sigma}_{\mathcal{S}^u}\mathbf{W}^{V}_{\Sigma};\end{aligned} \tag{10}$$

where $\mathbf{W}^{*}_{*} \in \mathbb{R}^{d\times d}$ represent the learnable weight matrices in the linear transformation. Combining the computed query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ with scaled dot-product attention. we can use the mean self-attention (MSA) and covariance self attention (CSA) to obtain newly generated sequence stochastic representations $\{\mathbf{z}^{\mu}_{\mathcal{S}^u}, \mathbf{z}^{\Sigma}_{\mathcal{S}^u}\}$:

$$\begin{aligned}\mathbf{z}^{\mu}_{\mathcal{S}^u} &= \mathbf{MSA}(\mathcal{S}^u) = \sigma\Big(\frac{\mathbf{Q}_{\mu}(\mathcal{S}^u)\mathbf{K}_{\mu}(\mathcal{S}^u)^{\mathrm{T}}}{\sqrt{d}}\Big)\mathbf{V}_{\mu}(\mathcal{S}^u);\\ \mathbf{z}^{\Sigma}_{\mathcal{S}^u} &= \mathbf{CSA}(\mathcal{S}^u) = \sigma\Big(\frac{\mathbf{Q}_{\Sigma}(\mathcal{S}^u)\mathbf{K}_{\Sigma}(\mathcal{S}^u)^{\mathrm{T}}}{\sqrt{d}}\Big)\mathbf{V}_{\Sigma}(\mathcal{S}^u).\end{aligned} \tag{11}$$

In addition to uncovering sequential patterns from linear transformations, we leverage the feed-forward network (FFN) to endow the model with non-linearity. The FFN with respect to both **MSA** and **CSA** at position t are defined as:

$$\begin{aligned}\mathbf{F}_{\mu}(\mathcal{S}^u_t) &= \mathrm{FFN}^{\mu}(\mathbf{MSA}(\mathcal{S}^u_t)) = \mathrm{elu}(\mathbf{z}^{\mu}_{\mathcal{S}^u_t}\mathbf{W}^{\mu}_1 + \mathbf{b}^{\mu}_1)\mathbf{W}^{\mu}_2 + \mathbf{b}^{\mu}_2;\\ \mathbf{F}_{\Sigma}(\mathcal{S}^u_t) &= \mathrm{FFN}^{\Sigma}(\mathbf{CSA}(\mathcal{S}^u_t)) = \mathrm{elu}(\mathbf{z}^{\Sigma}_{\mathcal{S}^u_t}\mathbf{W}^{\Sigma}_1 + \mathbf{b}^{\Sigma}_1)\mathbf{W}^{\Sigma}_2 + \mathbf{b}^{\Sigma}_2;\end{aligned} \tag{12}$$

where all the $\mathbf{W}^{*}_{*} \in \mathbb{R}^{d\times d}$ and $\mathbf{b}^{*}_{*} \in \mathbb{R}^{d}$ are trainable parameters in feed-forward networks.

3.5 Training and Evaluation

Wasserstein Distance. For embedding-based models, when measuring how accurately the model detects the main intention, we need to apply dot products between the user preference embedding and the intention embeddings. Similarly, for stochastic models, we need to identify the distances between distributions of the ground-truth labels and the inferred distributions. Many existing works formulating concepts as distributions use Kullback-Leibler (KL) divergence to compute the distribution distances. However, when two intentions are excessively unrelated, their stochastic representations will be expressed as two almost non-overlapping distributions, and the KL divergence will describe the distance as nearly infinity, resulting in value instability. Thus, we use Wasserstein distance to measure the distance between Gaussian distributions. Given two Gaussian distributions $q_i = \mathcal{N}(\mu_i, \Sigma_i)$ and $q_j = \mathcal{N}(\mu_j, \Sigma_j)$, the Wasserstein distance can be computed as:

$$d_W(i,j) = \|\mu_i - \mu_j\|_2^2 + \mathrm{trace}(\Sigma_i + \Sigma_j - 2(\Sigma_j^{1/2}\Sigma_i\Sigma_j^{1/2})^{1/2}). \tag{13}$$

For time efficiency, the second term in the above equation can be simplified as a calculation of Euclidean norm:

$$\mathrm{trace}(\Sigma_i + \Sigma_j - 2(\Sigma_j^{1/2}\Sigma_i\Sigma_j^{1/2})^{1/2}) = \|\Sigma_i^{1/2} - \Sigma_j^{1/2}\|_F^2, \tag{14}$$

where $\|\cdot\|_F^2$ is Frobenius norm that can be calculated by matrix multiplications.

Training Objective. Many deterministic sequential recommendation models apply Bayesian Personalize Ranking (BPR) loss [30] on the dot-product scores, making the ground-truth label nearest to the customer preference. In G-STO, we apply BPR loss on the

[2] $D_{ii} = \sum_j A_{ij}$.

previous defined Wasserstein distances between distributions to measure the correctness of the main intention identification:

$$\ell = -\sum_{\mathcal{S}^u \in \mathcal{S}} \sum_{t \in \{1,2,\cdots,|\mathcal{S}^u|\}} \log(\sigma(d_W(m_t^+, \hat{p}_t) - d_W(m_t^-, \hat{p}_t))), \tag{15}$$

where $\hat{p}_t$ dentoes the inferred distribution of user preference at position t, m_t^+ is the ground-truth shopping intention and m_t^- denotes the negative samples from the intentions that the user never interact with. During the inference stage, for user u, we calculate the distances between the customer preference distribution and the candidates set containing the ground-truth intention and 100 negative sampled intentions.

4 EXPERIMENTS

In this section, we evaluate the empirical effectiveness G-STO by studying the following research questions (RQs):

•**RQ1:** Does G-STO provide better shopping intention identification results than baselines?

•**RQ2:** Why do we need to design different kinds of scenarios? How does G-STO perform on different circumstances?

•**RQ3:** What is the influence of the intention relation graph regularizer and stochastic representations?

•**RQ4:** How sensitivity is G-STO to the hyper-parameters?

•**RQ5:** Why can G-STO alleviate intention cold start issue?

4.1 Data Curation

We create three benchmarks for main shopping intention identification using anonymized data from *amazon.com*. To provide a comprehensive evaluation, the ground-truth main shopping intention labels of the three datasets are created based on different real-life scenarios: (1) *original sequences*, using the raw sequences composed of user historical interactions to model the long-term shopping scenario; (2) *24-hours sequences*, leveraging more frequent user-intention interactions sampled from raw data to model the short-term shopping scenario; (3) *purchase sequences*, considering purchase actions of the raw sequences as strong signal of the main shopping intentions to model the purchase-related shopping scenario. Table 1 shows examples and the statistics of three datasets, original, 24-hours, purchase sequences.

Original sequences: We follow the "leave-one-out" strategy [18] to split the sequences into training, validation, and test datasets. To create the intention labels for each user and each split, we partition the curated historical sequence $\mathcal{S}^u$ for each user u into three parts: (1) the most recent intention action $\mathcal{S}^u_{|\mathcal{S}^u|}$ as the intention label for test set; (2) the second most recent action $\mathcal{S}^u_{|\mathcal{S}^u|-1}$ as the intention label for validation set; and (3) all remaining actions as training data. Note that during testing, the input sequences contain training actions and validation action.

24-hours sequences: Although the original sequences can more accurately describe the long-term preferences of customers and be more useful for the downstream task of next item recommendation, the users' final interactions cannot always convey the main shopping intention labels due to the random shift of interests. Assuming that, for more dense and frequent interactions in a short period, the last intentions from "leave-one-out" mechanism can better reflect the main shopping intentions, we assess the time intervals between the successive activities to breakdown the raw sequences. If there is a temporal gap longer than a pre-determined threshold (*e.g.*, 24 hours), we will insert a break-point and divide the entire sequence into two sub-sequences.

Purchase sequences: Aside from using the last intentions as the labels, the purchase actions can also serve as a very strong positive signal for main shopping intention identification. For each customer's historical data, the purchase actions can be viewed as the main shopping intentions for its preceding sub-sequences. As purchasing activities are dispersed over the sequences, we can only apply user-based split for train/val/test separation in this scenario.

4.2 Experiment Setup

4.2.1 Evaluation Protocol. We evaluate all models with the following metrics: (1) **NDCG@10:** A position-aware metric which assigns larger weights for higher positions; (2) **Hit@K, (K=1,2,5,10):** Metrics counting the fraction of times that the ground-truth intention is among top K predictions. We report the averaged metrics over all users. We select the model to report the test set performances based on the best validation NDCG@10 score.

4.2.2 Baselines. We compare our proposed model with baselines from three different groups: (1) *static recommendation methods*: **Count-based Bayesian (CB)** is a non-learning approach that solely considers the appearance frequency of shopping intentions. We consider the intention frequencies across the entire market as prior, and the intention frequencies in each user's shopping sequence as likelihood. The final shopping intention rankings are derived using posterior, which is the multiplication of prior and likelihood; **LightGCN [15]** is a state-of-the-art graph-based static recommendation method, which considers high-order collaborative signals in user-item graph; (2) *deterministic sequential recommendation methods*: **SASRec [18]** is a self-attention based sequential recommendation system model that captures long-term semantics and short-term dynamics; (3) *stochastic sequential recommendation methods*: **DT4SR [8]** is a distribution-based method, mapping the intentions to elliptical Gaussian distributions and then send them into two separate Transformer-based model to infer the users' preferences; **STOSA [9]** is a state-of-the-art distribution-based recommendation system model. STOSA extends DT4SR by proposing a new stochastic self-attention mechanism to further improve the combination of Transformer and stochastic representations; (4) *VAE-based sequential recommendation methods:* **SVAE [32]** is a recurrent version of VAE, combining recurrent neural network (RNN) and VAE. The model outputs the probability distribution of the most likely future preferences at each time step; **ACVAE [43]** is a state-of-the-art VAE-based model, first introducing the adversarial training for sequence generation, enabling the model to generate high-quality latent variables.

4.2.3 Implementation Details. For the hyper-parameters, the learning rate is set as $1e-4$, the maximum number of epochs is 500, the batch size is 128. For the stochastic representations, the hidden dimensionalities of mean and covariance embeddings are set as 64. For the intention relation graph regularizer, we apply 1-layer graph convolution network (GCN). We train and test our code on the system Ubuntu 18.04.4 LTS with CPU: Intel(R) Xeon(R) Silver 4214 CPU@

Table 1: An example of generated sequences under different scenarios. Different numbers represent different shopping intentions. The blue intentions indicate the validation set labels and the red intentions indicate the test set labels.

Scenarios	Examples	Explanation	Avg. Len	#Users
Original	(0, 4293, 234, 2173, 232, 183, 913, 4298, 582, 98, 4299)	Raw Sequences obtained from user historical data;	49.3	140k
24-Hours	(0, 4293, 234, 2173, 232, 183), (913, 4298, 582, 98, 4299)	Time interval between 183 and 913 is longer than 24 hours;	15.7	130k
Purchase	(377, 19, 76, 6, 87, 112)~~, 90, 4, 346, 85~~ (0, 4293, 234, 2173, 232, 183)~~, 913, 4298, 582, 98, 4299~~	112 is the PURCHASE action. 183 is the PURCHASE action;	38.7	90k

2.20GHz and GPU: NVIDIA V100. We implement our method using Python 3.8 and PyTorch 1.6 [26]. For the hyper-parameters, the learning rate is set as $1e-4$, the maximum number of epochs is 500, the batch size is 128. For the stochastic representations, the hidden dimensionalities of mean and covariance embeddings are set as 64. For the intention relation graph regularizer, we apply 1-layer graph convolution network (GCN). During training, we use the Adam [19] optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ in our experiments for all the models. We select the best set of hyper-parameters of the models based on the NDCG@10 on the corresponding validation sets.

4.3 Performance Comparison

Cross-Method Comparison (RQ1). Table 2 reports the performances of G-STO and the baselines on all three benchmarks. The results demonstrate that G-STO consistently outperforms all baselines in terms of all metrics on all the three datasets. Compared with the strongest baseline, STOSA, G-STO shows significant improvements of 18.08% of Hit@1, 6.11% of NDCG@10, 16.87% of Hit@2, 11.87% of Hit@5, and 7.01% of Hit@10 on average of three datasets. From the results, we have the following observations:
(1) The performance gaps between our method and static methods, CB and LightGCN, show the importance of temporal order sequential information. Different from product-level recommendations, the intentions may appear multiple times in single sequence, making counts-based method, CB, achieve comparable results to state-of-the-art static method of LightGCN, and even SASRec.
(2) Comparing our method with the backbone model, SASRec, G-STO shows significant improvement of 43.44% in Hit@1 and 15.50% in NDCG@10. This indicates that the stochastic representations expand the latent space for user-intention interactions and equip the model with collaborative transitivity. Besides, the graph regularizer captures global intention information, enabling model to better understand the intentions. Both modules help enhance the intention identification capabilities, particularly in small-data situations;
(3) The comparison between G-STO and other distribution-based recommendation methods, DT4SR and STOSA, shows the efficacy of leveraging the intention relation graph as prior knowledge to regularize the stochastic representations and reveals the potential to further incorporate G-STO with distribution-based attention;
(4) Comparing G-STO with VAE-based methods, we find that VAE-based methods are easy to suffer posterior collapse problems: if the decoder is too expressive, the KL divergence term in the loss will converge to 0, generating similar latent representations for all inputs. This situation becomes worse for less-interacted under-trained intentions. On the contrary, our model leverages graph regularization to transfer knowledge from more-interacted intentions to less-interacted ones to mitigate this issue.

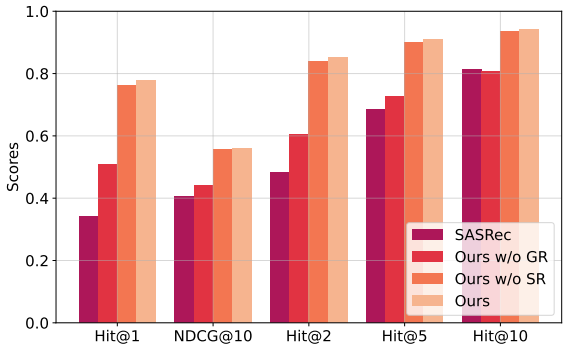


Figure 3: Ablation study on model components. The performances are averaged on three datasets.

Cross-Dataset Comparison (RQ2). From Table 2, we can also compare the performances horizontally to gain insights into the models' performances in different scenarios. We notice that the performance improvements made by G-STO, compared with baselines, varies by different scenarios. The performance gap is larger on 24-hours and purchase sequences than on original sequences. We summarize the reasons as follows: (1) On 24-hour sequences, G-STO and all the baselines achieve higher absolute performances than on the other two categories. As the 24-hour sequences consist of more dense and frequent activities, where users' severe random interest shifts are less likely to appear, and the hidden sequential dynamic patterns are easier for models to learn and capture. (2) On purchase sequences, G-STO performs slightly better, while most baselines perform poorly. This is because of the difference of data split for train/validation/test set: we apply user-based split on purchase sequences, and "leave-one-out" on the other two categories. Thus, it is easier for model to face new users/intentions during validation or testing, making the models more likely to suffer from the cold-start issue. With the graph-regularized stochastic Transformer, G-STO can better resolve this issue than the other baselines, enlarging the performance gap on this dataset.

4.4 A/B Test

We have an existing products recommendation feature using a static mapping created from the work [11]. However, the procedure only provides static product to product recommendation, it doesn't take customer's shopping history and provide personalized shopping experience. Our goal was to improve the recommendation's relevance by taking customers' shopping intentions into consideration while optimizing sales and revenue. Our work, G-STO, has been recently deployed for online A/B testing as the treatment group,

Table 2: Performance Comparison in Hit@1, NDCG@10, Hit@2, Hit@5, and Hit@10 on three different datasets. The best results are boldfaced.

Scenarios (→)	Original					24-Hours					Purchase				
Methods(↓)	Hit@1	NDCG@10	Hit@2	Hit@5	Hit@10	Hit@1	NDCG@10	Hit@2	Hit@5	Hit@10	Hit@1	NDCG@10	Hit@2	Hit@5	Hit@10
CB	0.3242	0.4007	0.5224	0.7268	0.7708	0.3327	0.3949	0.5235	0.7233	0.7542	0.2767	0.3738	0.4376	0.6753	0.7502
LightGCN	0.2810	0.3528	0.4743	0.6461	0.7436	0.3545	0.4423	0.5411	0.7265	0.8783	0.2281	0.3635	0.4054	0.6470	0.7038
SASRec	0.3192	0.3886	0.4509	0.6529	0.7865	0.4341	0.4537	0.5824	0.7610	0.8708	0.2761	0.3763	0.4160	0.6372	0.7804
DT4SR	0.4879	0.4417	0.5908	0.7231	0.8197	0.5950	0.5003	0.7027	0.8245	0.8968	0.4410	0.3863	0.5202	0.6308	0.7097
STOSA	0.5646	0.4736	0.6598	0.7714	0.8530	0.6240	0.5484	0.7106	0.8298	0.9183	0.6017	0.4781	0.6779	0.7763	0.8449
SVAE	0.2722	0.2666	0.3014	0.3573	0.4041	0.3142	0.3718	0.4863	0.7163	0.8257	0.2897	0.3706	0.4221	0.6419	0.7871
ACVAE	0.3866	0.4582	0.4843	0.6712	0.7107	0.3264	0.4573	0.5286	0.7261	0.9125	0.2858	0.3743	0.4834	0.6682	0.7644
Ours	**0.7061**	**0.5297**	**0.7890**	**0.8623**	**0.9050**	**0.8985**	**0.6115**	**0.9592**	**0.9896**	**0.9955**	**0.7280**	**0.5423**	**0.8062**	**0.8818**	**0.9259**

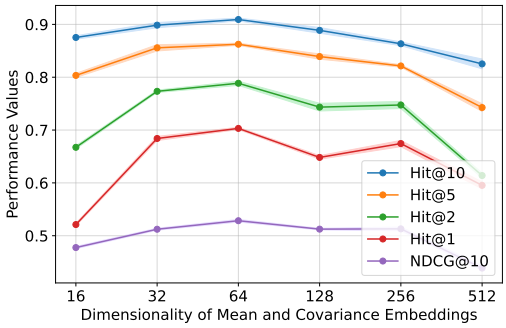


Figure 4: Parameter studies of our model about the hidden dimensionality of mean and covariance embeddings on original sequences.The red line indicates that the best performances are obtained when the hidden size is 64.

Table 3: Ablation studies on stochastic embeddings.

Methods	Recall@1	NDCG@10	Recall@2	Recall@5	Recall@10
G-STO w/o covar	0.5516	0.4418	0.6280	0.7030	0.7571
G-STO w/o G-mean	0.5685	0.4744	0.6626	0.7739	0.8512
G-STO w/o G-covar	0.6704	0.5163	0.7553	0.8359	0.8962
G-STO	**0.7061**	**0.5297**	**0.7890**	**0.8623**	**0.9050**

while the control group is the existing solution using static mapping. The model takes customer's past shopping activities as input, then predict the possible product categories for downstream recommendations. This online testing has been run for 2 weeks, and we have already found the achievements shown below:

We observe a World Wide (WW) commercial success:

- WW annual revenue was improved by 1%;
- WW annual sales were improved by 2%.

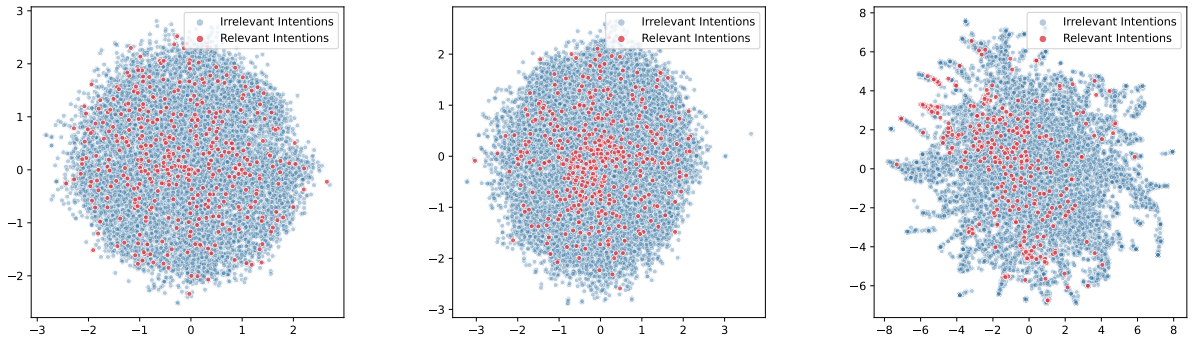
4.5 Ablation Study (RQ3)

Ablation on Model Structure. Figure 3 presents the ablation study to verify the effects of different components in G-STO. We compare G-STO with model variants that remove one of the key components: (1) **Ours w/o Graph Regularizer (GR):** We remove the intention graph regularizer and directly send the stochastic representations generated by the stochastic embedding layer into Transformers; (2) **Ours w/o Stochastic Representation (SR):** We degrade the stochastic representations into deterministic embeddings for each intention. Then, we have the following observations: (1) Compared with the backbone model, SASRec, stochastic representations can bring significant improvements of 16.48% in Hit@1 and 3.66% in NDCG@10, and graph regularizer can bring 41.85% of Hit@1 and 14.92% of NDCG@10 increase. Removing either the graph regularizer or the stochastic representation will cause performance drop in all metrics on all three datasets, which shows the efficacy of the components in our model design; (2) Comparing the performance drops caused by removing different components, we notice that removing the intention relation graph regularizer is more harmful than the stochastic representation. This is because some collaborative transitivity relations captured by stochastic representations may be already included in the global intention relation graph.

Table 4: Study on different graph neural networks (GNN) in intention graph regularizer on original sequences.

Methods	Recall@1	NDCG@10	Recall@2	Recall@5	Recall@10
Ours (1-layer GCN)	0.7061	0.5297	0.7890	0.8623	0.9050
Ours (2-layer GCN)	0.4970	0.4633	0.6302	0.7756	0.8564
Ours (1-layer GAT)	0.3390	0.4660	0.5698	0.7949	0.9202

Ablation on stochastic Representations. Table 3 shows the ablation studies on stochastic embeddings in G-STO. Our objective is to study the effect of various components of the stochastic embeddings and assess if the use of graph regularization enhances their performance. To achieve this, we conduct evaluations under three distinct experimental settings: (1) exclusion of graph regularization on the covariance embedding (w/o G-covar); (2) exclusion of graph regularization on the mean embedding (w/o G-mean); and (3) exclusion of the entire covariance embedding (w/o covar). The results of our evaluation indicate that the removal of graph regularization on either component of the stochastic embedding results in a significant decrease in performance. This is due to the fact that regularizing only one component of the embedding allows for more flexibility in the other component, thereby weakening the overall regularization power. Furthermore, the removal of the entire



(a) Intention embeddings trained by SASRec. (b) Intention embeddings trained by STOSA. (c) Intention embeddings trained by G-STO.

Figure 5: T-SNE visualization of mission embeddings trained by SASRec, STOSA, and G-STO. Blue points are irrelevant points to a certain intention. Red points are relevant shopping intentions.

covariance embedding results in a further decline in performance, as the model degenerates into a lower-dimensional deterministic sequential recommendation model.

4.6 Parameter Studies (RQ4)

Effect of Hidden Dimensionalities. We study the influence of hidden dimensionalities of mean and covariance embedding, d, towards our methods. Among the searching range of [16, 32, 64, 128, 256, 512], $d = 64$ works the best. If d is too small, the model cannot encode the shopping intentions well and cannot learn some high-dimensional relation between intentions. If d is too large, it will become a problem for model to learn the high-dimensional representations, also causing a performance drop.

Effect of GNN Types. Table 4 presents the results of using various GNNs in the intention relation graph regularizer. The results indicate that the graph convolution network (GCN) outperforms the graph attention network (GAT). This is because GAT requires additional edge weight learning for dense, noisy graphs. However, our intention relation graph is not dense, and the edge quality is high, rendering the training of additional attention weights unnecessary. We utilize the MAD metric [3] to evaluate the smoothness of node representations. The results show that as the number of GCN layers increases from one to two, the MAD drops significantly from 0.774 to 0.379, indicating that the 2-layer GCN produces smoother representations. On the other hand, the performance drop as shown in Table 4 suggests that the 2-layer GCN is over-smoothing.

Effect of Number of GCN Layers. Besides, we also investigate the effect of the number of GCN layers on G-STO. With a single layer of graph convolution, GCN can only gather data from its immediate neighbors. Information from larger neighborhoods can only be incorporated when numerous GCN layers are applied. The results in Table 4 indicate that 1-layer GCNs perform better than multi-layer GCNs. The reason is that the directly related intentions on the graph are more crucial when identifying the main shopping intentions from consumer historical data, and introducing more distant neighbours on the graph can introduce more noise.

4.7 Intention Embeddings Comparison (RQ5)

To better explain the efficacy of the proposed intention relation graph regularizer, we compare the shopping intention representations trained by SASRec, STOSA, and G-STO via t-SNE visualization [40] in Figure 5. The red points in the figure represent the relevant shopping intentions to the intention "miniature", while the blue points indicate the irrelevant ones. From Figure 5(a), we observe that although SASRec is trained on user historical data to learn some correlations between intentions, the embeddings of relevant shopping intentions are still quite scattered across the latent space. From Figure 5(b), the red points start to cluster with each other, because the state-of-the-art distribution-based model, STOSA, can capture the collaborative transitivity between intentions, which are ignored by SASRec. In contrary, from Figure 5(c), it is obvious that the related intention embeddings further cluster with each other. This proves that the intention relation graph truly regularizes the stochastic representations, constricting relevant intentions embeds closer to improve G-STO performances.

5 CONCLUSION

We presented G-STO, a graph-regularized stochastic Transformer-based model for main shopping intention identification. G-STO first models the shopping intentions as Gaussian distributions and then creates an intention relation graph as prior knowledge to regularize these distributions. The regularized stochastic representations will be fed to the Transformer architecture for main shopping intention identification. We perform experiments under three different scenarios in real-life applications. Extensive experimental results demonstrate the superiority of G-STO over the state-of-the-art baselines. In the future, we will change the Transformer architecture to accommodate distribution-based models more effectively.

ACKNOWLEDGMENTS

This work was supported in part by NSF (IIS2008334, IIS-2106961, CAREER IIS-2144338), ONR (MURI N00014-17-1-2656) and partly supported in part by Amazon.com LLC. We would like to thank Tong Zhao for his insightful advice on this work.

REFERENCES

[1] Aleksandar Bojchevski and Stephan Günnemann. 2018. Deep Gaussian Embedding of Graphs: Unsupervised Inductive Learning via Ranking. In *International Conference on Learning Representations*.

[2] Jianxin Chang, Chen Gao, Yu Zheng, Yiqun Hui, Yanan Niu, Yang Song, Depeng Jin, and Yong Li. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 378–387.

[3] Deli Chen, Yankai Lin, Wei Li, Peng Li, Jie Zhou, and Xu Sun. 2020. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 3438–3445.

[4] Weixin Chen, Mingkai He, Yongxin Ni, Weike Pan, Li Chen, and Zhong Ming. 2022. Global and Personalized Graphs for Heterogeneous Sequential Recommendation by Learning Behavior Transitions and User Intentions. In *Proceedings of the 16th ACM Conference on Recommender Systems*. 268–277.

[5] Xu Chen, Hongteng Xu, Yongfeng Zhang, Jiaxi Tang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2018. Sequential recommendation with user memory networks. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 108–116.

[6] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *Proceedings of the ACM Web Conference 2022*. 2172–2182.

[7] Robin Devooght and Hugues Bersini. 2017. Long and short-term recommendations with recurrent neural networks. In *Proceedings of the 25th Conference on User Modeling, Adaptation and Personalization*. 13–21.

[8] Ziwei Fan, Zhiwei Liu, Shen Wang, Lei Zheng, and Philip S Yu. 2021. Modeling Sequences as Distributions with Uncertainty for Sequential Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3019–3023.

[9] Ziwei Fan, Zhiwei Liu, Yu Wang, Alice Wang, Zahra Nazari, Lei Zheng, Hao Peng, and Philip S Yu. 2022. Sequential Recommendation via Stochastic Self-Attention. In *Proceedings of the ACM Web Conference 2022*. 2036–2047.

[10] Yantao Gong, Cao Liu, Jiazhen Yuan, Fan Yang, Xunliang Cai, Guanglu Wan, Jiansong Chen, Ruiyao Niu, and Houfeng Wang. 2021. Density-based dynamic curriculum learning for intent detection. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 3034–3037.

[11] Junheng Hao, Tong Zhao, Jin Li, Xin Luna Dong, Christos Faloutsos, Yizhou Sun, and Wei Wang. 2020. P-companion: A principled framework for diversified complementary product recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2517–2524.

[12] Helia Hashemi, Hamed Zamani, and W Bruce Croft. 2021. Learning Multiple Intent Representations for Search Queries. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 669–679.

[13] Ruining He and Julian McAuley. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 191–200.

[14] Shizhu He, Kang Liu, Guoliang Ji, and Jun Zhao. 2015. Learning to represent knowledge graphs with gaussian embedding. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. 623–632.

[15] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.

[16] Junyang Jiang, Deqing Yang, Yanghua Xiao, and Chenlu Shen. 2019. Convolutional Gaussian embeddings for personalized recommendation with uncertainty. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2642–2648.

[17] Sameer Kanase, Yan Zhao, Shenghe Xu, Mitchell Goodman, Manohar Mandalapu, Benjamyn Ward, Chan Jeon, Shreya Kamath, Ben Cohen, Vlad Suslikov, Yujia Liu, Hengjia Zhang, Yannick Kimmel, Saad Khan, Brent Payne, and Patricia Grao. 2022. An Application of Causal Bandit to Content Optimization. In *Proceedings of the 5th Workshop on Online Recommender Systems and User Modeling (ORSUM 2022), in conjunction with the 16th ACM Conference on Recommender Systems (RecSys 2022), Seattle, WA, USA*.

[18] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[19] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[20] Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

[21] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[22] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. arXiv:2305.13731 [cs.IR]

[23] Jiacheng Li, Tong Zhao, Jin Li, Jim Chan, Christos Faloutsos, George Karypis, Soo-Min Pantel, and Julian McAuley. 2022. Coarse-to-Fine Sparse Sequential Recommendation. *arXiv preprint arXiv:2204.01839* (2022).

[24] Chang Liu, Xiaoguang Li, Guohao Cai, Zhenhua Dong, Hong Zhu, and Lifeng Shang. 2021. Noninvasive self-attention for side information fusion in sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4249–4256.

[25] Chen Ma, Peng Kang, and Xue Liu. 2019. Hierarchical gating networks for sequential recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 825–833.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[27] Bo Peng, Zhiyun Ren, Srinivasan Parthasarathy, and Xia Ning. 2021. HAM: Hybrid Associations Models for Sequential Recommendation. *IEEE Transactions on Knowledge & Data Engineering* 01 (2021), 1–1.

[28] Chen Qian, Fuli Feng, Lijie Wen, and Tat-Seng Chua. 2021. Conceptualized and Contextualized Gaussian Embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 13683–13691.

[29] Massimo Quadrana, Alexandros Karatzoglou, Balázs Hidasi, and Paolo Cremonesi. 2017. Personalizing session-based recommendations with hierarchical recurrent neural networks. In *proceedings of the Eleventh ACM Conference on Recommender Systems*. 130–137.

[30] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[31] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*. 811–820.

[32] Noveen Sachdeva, Giuseppe Manco, Ettore Ritacco, and Vikram Pudi. 2019. Sequential variational autoencoders for collaborative filtering. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 600–608.

[33] Xin Shen et al. 2020. FishRecGAN: An End to End GAN Based Network for Fisheye Rectification and Calibration. Advances in Artificial Intelligence and Machine Learning. 2023; 3 (2): 69.

[34] Xin Shen, Kyungdon Joo, and Jean Oh. 2023. FishRecGAN: An End to End GAN Based Network for Fisheye Rectification and Calibration. arXiv:2305.05222 [cs.CV]

[35] Xin Shen, Xiaonan Zhao, and Rui Luo. 2023. Semantic Embedded Deep Neural Network: A Generic Approach to Boost Multi-Label Image Classification Performance. arXiv:2305.05228 [cs.CV]

[36] Xin Shen, Yan Zhao, Sujan Perera, Yujia Liu, Jinyun Yan, and Mitchell Goodman. 2023. Learning Personalized Page Content Ranking Using Customer Representation. arXiv:2305.05267 [cs.IR]

[37] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[38] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.

[39] Md Mehrab Tanjim, Congzhe Su, Ethan Benjamin, Diane Hu, Liangjie Hong, and Julian McAuley. 2020. Attentive sequential models of latent intent for next item recommendation. In *Proceedings of The Web Conference 2020*. 2528–2534.

[40] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[41] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *Proceedings of the Web Conference 2021*. 878–887.

[42] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. 2020. SSE-PT: Sequential recommendation via personalized transformer. In *Fourteenth ACM Conference on Recommender Systems*. 328–337.

[43] Zhe Xie, Chengxuan Liu, Yichi Zhang, Hongtao Lu, Dong Wang, and Yue Ding. 2021. Adversarial and contrastive variational autoencoder for sequential recommendation. In *Proceedings of the Web Conference 2021*. 449–459.

[44] Yatao Yang, Biyu Ma, Jun Tan, Hongbo Deng, Haikuan Huang, and Zibin Zheng. 2021. FINN: Feedback Interactive Neural Network for Intent Recommendation. In *Proceedings of the Web Conference 2021*. 1949–1958.

[45] Yue Yu, Rongzhi Zhang, Ran Xu, Jieyu Zhang, Jiaming Shen, and Chao Zhang. 2023. Cold-Start Data Selection for Few-shot Language Model Fine-tuning: A Prompt-Based Uncertainty Propagation Approach. arXiv:2209.06995 [cs.CL]

[46] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation.. In *IJCAI*. 4320–4326.

[47] Jing Zhao, Pengpeng Zhao, Lei Zhao, Yanchi Liu, Victor S Sheng, and Xiaofang Zhou. 2021. Variational self-attention network for sequential recommendation.

In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 1559–1570.

[48] Lei Zheng, Chaozhuo Li, Chun-Ta Lu, Jiawei Zhang, and Philip S Yu. 2019. Deep Distribution Network: Addressing the Data Sparsity Issue for Top-N Recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1081–1084.

[49] Nengjun Zhu, Jian Cao, Yanchi Liu, Yang Yang, Haochao Ying, and Hui Xiong. 2020. Sequential modeling of hierarchical user intention and preference for next-item recommendation. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 807–815.