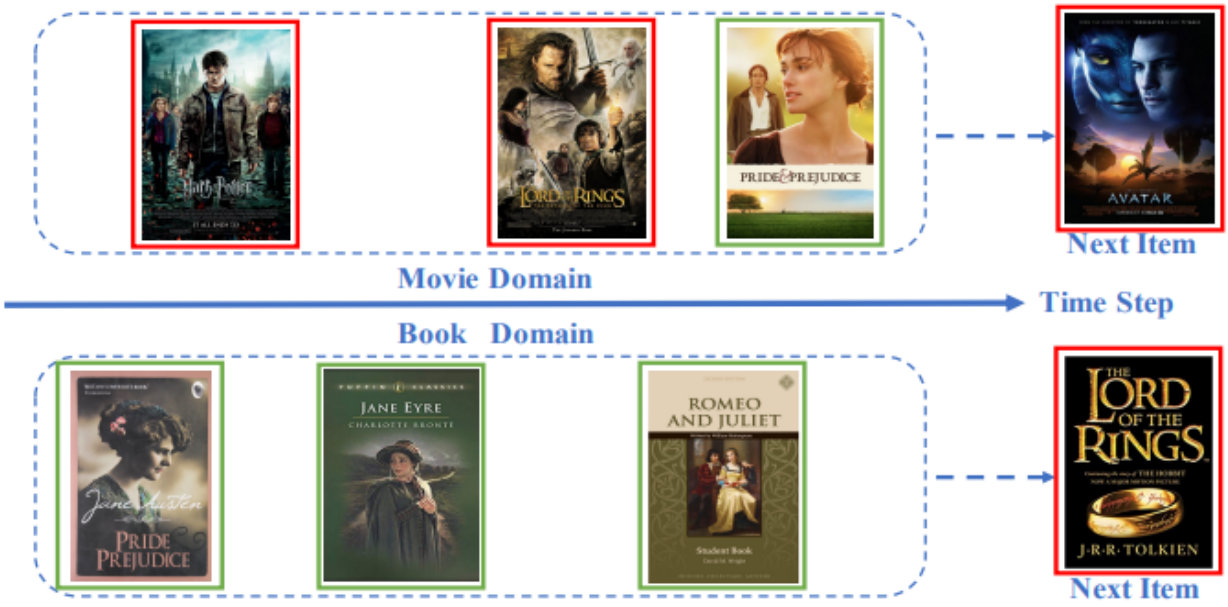
C2DSR

Sequential recommendation, is a type of recommendation system that takes into account the order in which items are interacted with by users. Unlike traditional recommendation systems that might treat user interactions with items as independent events, sequential recommendation systems recognize that the sequence of interactions can provide additional context and thus can lead to more accurate predictions. For example, if a user watches a horror movie followed by a thriller, a sequential recommendation system might infer a preference for suspenseful content and recommend a mystery novel next.



Missing：

Bottleneck of the Transferring Module: In the context of cross-domain recommendations, a transferring module is typically used to transfer knowledge from one domain (e.g., books) to another (e.g., movies). The bottleneck here refers to the limitations of this module to effectively transfer relevant information that can aid in making accurate recommendations in the target domain. This bottleneck can result from several factors, such as the complexity of mapping preferences across domains with different characteristics or the lack of sufficient data to learn this mapping accurately.

Not Considering Inter-Sequence Item Relationships: Traditional sequential recommendation systems often focus on the relationships between items within a single sequence of interactions (intra-sequence relationships) but may ignore the relationships between items across different sequences (inter-sequence relationships). Inter-sequence relationships can provide valuable collaborative signals, as they reflect broader patterns in user behavior across multiple sequences, which can be leveraged to improve recommendation quality.

Proposed Solution:

The authors propose the C2DSR model to capture precise user preferences by leveraging both intra- and inter-sequence item relationships and jointly learning single- and cross-domain user preferences. The model uses a graph neural network to mine inter-sequence item collaborative relationships and a sequential attentive encoder to capture intra-sequence item sequential relationships.

Contrastive Infomax Objective: A novel aspect of the C2DSR model is the contrastive infomax objective, which enhances the correlation between single- and cross-domain user representations by maximizing their mutual information. This is inspired by the infomax principle, which is commonly used in contrastive learning.

Related work：

The pioneering work is π-net [28], which first generates singledomain representation by modeling item interaction sequence in each single domain, and then transfers the learned single-domain representations into other domains with a gated transferring module. To enhance the transferring module in π-net, MIFN [27] further introduces an external knowledge graph transferring module to guide the connection between different domain items。

Despite the promising improvements, the pipeline-style paradigm learns single-domain user preference separately, which usually generates domain-biased user representations. Simply transferring the biased single-domain preference can be intractable to describe precise cross-domain user preference, which would easily lead to unstable and sub-optimal recommendation results. Therefore, we argue that it is necessary to learn the **single- and crossdomain user preference in a joint way** for unbiased information transferring. More importantly, typical user interactions in different domains usually exhibit related preferences, thus **we further consider the correlation between the single- and cross- domain user preference.** Besides, previous CDSR works [28, 37] only focus on modeling the intra-sequence item relationship to capture

the sequential pattern signal to obtain sequence representation (i.e, user representation), but ignore the inter-sequence relationship of items (as shown in Figure 2), which provides valuable collaborative signal to generate better user representation. Therefore, **we propose to capture the intra- and inter- sequence item relationships at same time** for representation learning

3 METHODOLOGY

In this section, we propose our model C2DSR, which captures and transfers valuable information across domains by modeling singledomain and cross-domain representations. There are three major components of C2DSR: (1) Graphical and attentional encoder, which includes an embedding layer, a graph neural network module, and a self-attention module to generate a series of sequential representations (i.e., user representations) for each interaction sequence. (2) Sequential training objective, which includes two training objectives for single-domain and cross-domain interaction sequences to obtain single-domain and cross-domain user representations. (3) Contrastive infomax objective, which leverages the mutual information maximization principle to enhance the correlation between single-domain and cross-domain representations.

3.1 Graphical and Attentional Encoder

3.1.1 Embedding Initialization Layer.

stage, to obtain initialized item representations for three single- and cross- item interaction sequences, we introduce three parameter matrices $\mathbf{E}^X \in \mathbb{R}^{|\mathcal{X}|\times d}$, $\mathbf{E}^Y \in \mathbb{R}^{|\mathcal{Y}|\times d}$ and $\mathbf{E} \in \mathbb{R}^{(|\mathcal{X}|+|\mathcal{Y}|)\times d}$,

Besides, to recognize the ordered information of sequence, we define a learnable parameter position embedding matrix

$$\mathbf{T} \in \mathrm{R}^{M\times d}$$

to enhance the input item embeddings for the self-attention module。

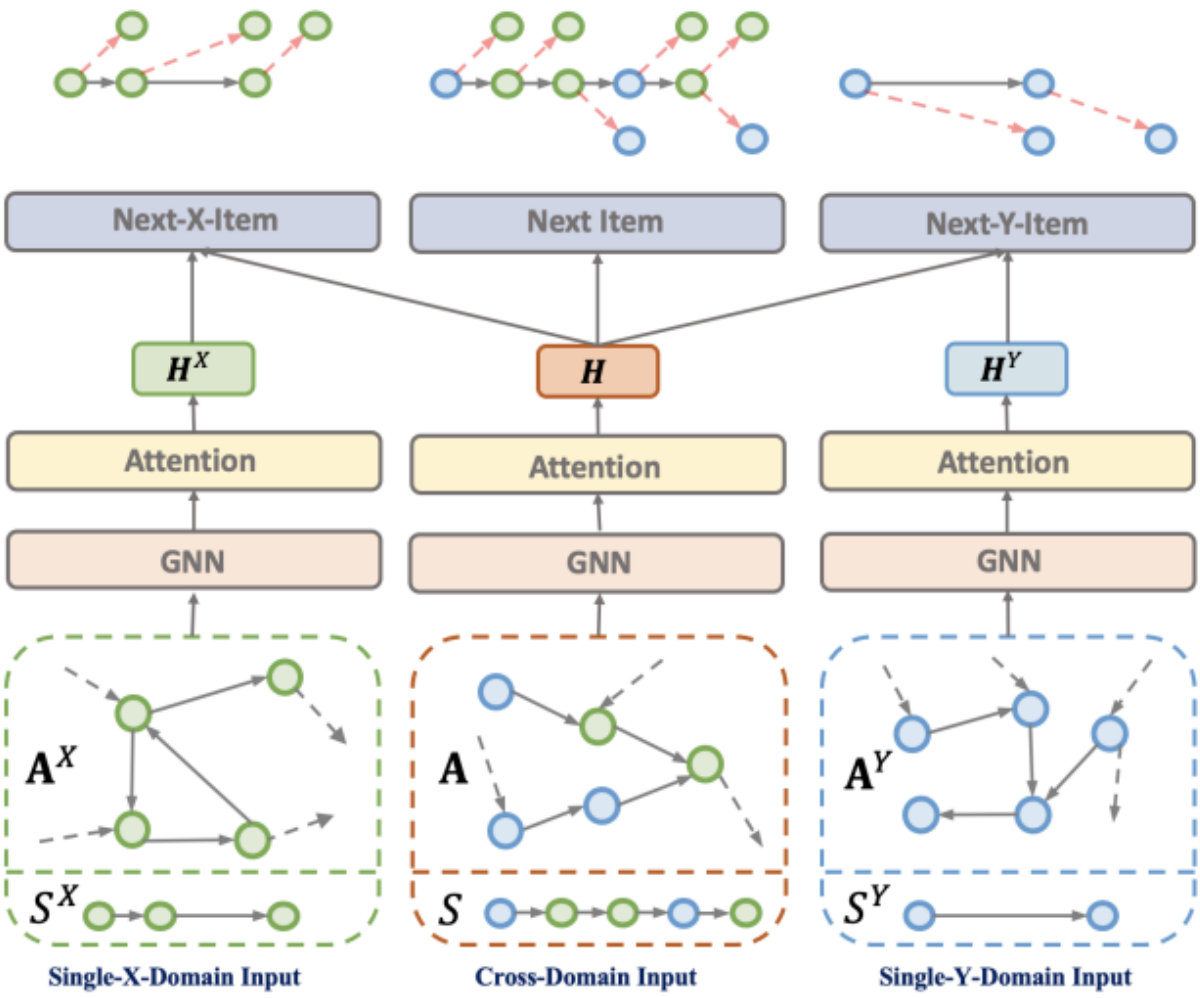


Figure 3: A toy example of sequential training objective for CDSR. The red dotted lines indicate the next prediction item.

3.1.2 Graph Neural Network Module.

As a promising way to model the inter-sequence item relationship, we consider employing the graph neural network。

we also remove the convolution matrix multiplication and non-linear activation function to capture the cooccurrence collaborative filtering signal better. Specifically, given the binary directed item-item matrices

$$\mathbf{A}^X, \mathbf{A}^Y, \mathbf{A}$$

and initialized embedding

$$\mathbf{G}_0^X = \mathbf{E}^X, \mathbf{G}_0^Y = \mathbf{E}^Y, \mathbf{G}_0 = \mathbf{E},$$

we have:

$$\mathbf{G}_1^X = \mathrm{Norm}(\mathbf{A}^X)\mathbf{G}_0^X, \quad \mathbf{G}_1^Y = \mathrm{Norm}(\mathbf{A}^Y)\mathbf{G}_0^Y, \quad \mathbf{G}_1 = \mathrm{Norm}(\mathbf{A})\mathbf{G}_0, \quad (2)$$

where Norm(·) denote the row-normalized function,

$$\mathbf{G}_1^X, \mathbf{G}_1^Y, \mathbf{G}_1$$

To fully capture graphical information across layers, we use a Mean(·) function to average them to fulfill **item representations** as:

$$\mathbf{G}^X = \mathrm{Mean}(\mathbf{G}_l^X) + \mathbf{E}^X, \mathbf{G}^Y = \mathrm{Mean}(\mathbf{G}_l^Y) + \mathbf{E}^Y, \mathbf{G} = \mathrm{Mean}(\mathbf{G}_l) + \mathbf{E}. \quad (3)$$

3.1.3 Self-Attention Module

To capture the intra-sequence item relationship, we utilize the self-attention module to encode the interaction sequences。

, there are two types of sub-layers: (1) the multi-head self-attention layer captures the complex intra-sequence item dependency in an **interaction sequence**, (2) the point-wise feed-forward layer endows a non-linearity projection to output the final sequential representations.

[2]We add a <pad> item to the corresponding positions to separate single-domain sequences S^X and S^Y from S. For example, S^X = [<pad>, x_1, x_2, <pad>, x_3], S^Y = [y_1, <pad>, <pad>, y_2, <pad>], S = [y_1, x_1, x_2, y_2, x_3], where a constant zero vector $\mathbf{0}$ is used as the embedding for the <pad> item.

For brevity, we employ the padding technique on the input sequences , and then formulate the overall encoding process (containing several feed-forward and self-attention layers) as

$$\boldsymbol{H}^X = \texttt{AttEncoder}^X(S^X, \mathbf{G}^X), \quad \boldsymbol{H}^Y = \texttt{AttEncoder}^Y(S^Y, \mathbf{G}^Y), \\ \boldsymbol{H} = \texttt{AttEncoder}(S, \mathbf{G}), \tag{4}$$

where

$$\boldsymbol{H}^X \in \mathbb{R}^{|S| \times d}, \boldsymbol{H}^Y \in \mathbb{R}^{|S| \times d}, \boldsymbol{H} \in \mathbb{R}^{|S| \times d}$$

are sequential outputs, and we adopt several AttEncoder parameters to encode different interaction sequence

$$S^X, S^Y$$

and S for better adaptation.the most common strategy is to train the model to recommend the next item based on the observed sequence directly. Taking the domain X as an example, given single-domain padding interaction sequence

$$S^X = [\text{<pad>}, x_1, x_2, \text{<pad>}, \ldots, x_t]$$

, and its expected next item

$$x_{t+1}$$

. We adopt the commonly used training strategy to optimize our encoder as follows:

$$\mathcal{L}^X_{\text{single}} = \sum_{S^X \in \mathcal{S}} \sum_t \mathcal{L}^X_{\text{single}}(S^X, t) \\ \mathcal{L}^X_{\text{single}}(S^X, t) = -\log \mathrm{P}^X_{\text{single}}(x_{t+1} | [\text{<pad>}, x_1, x_2, \text{<pad>}, \ldots, x_t]), \tag{5}$$

where the probability $\mathrm{P}^{X}_{\text{single}}(x_{t+1}|[\text{<pad>}, x_1, x_2, \text{<pad>}, \ldots, x_t])$ is designed to be proportional to the similarity between all the items $x \in \mathcal{X}$ and the given sequence in the vector space. Based on the learned representations $\boldsymbol{H}^X$ and $\boldsymbol{H}$, we calculate single-X-domain prediction probability $\mathrm{P}^{X}_{\text{single}}(\cdot)$ as follows (similarly for domain Y):

$$\mathrm{P}^{X}_{\text{single}}(x_{t+1}|[\ldots, x_t]) = \texttt{Softmax}\left(\boldsymbol{h}^X_t \mathbf{W}^X + \boldsymbol{h}_t \mathbf{W}^X\right)_{x_{t+1}} \tag{6}$$

每個item都會有representation h。

Cross-Domain Item Prediction.

$$\begin{aligned} \mathcal{L}_{\text{cross}} &= \sum_{S \in \mathcal{S}} \sum_{t} \mathcal{L}_{\text{cross}}(S, t), \\ \mathcal{L}_{\text{cross}}(S, t) &= \begin{cases} -\log \mathrm{P}^{X}_{\text{cross}}(x_{t+1}|[y_1, x_1, x_2, \ldots, x_t]), \\ -\log \mathrm{P}^{Y}_{\text{cross}}(y_{t+1}|[y_1, x_1, x_2, \ldots, x_t]), \end{cases} \end{aligned} \tag{7}$$

where we implement the prediction probability $\mathrm{P}^{X}_{\text{cross}}(\cdot), \mathrm{P}^{Y}_{\text{cross}}(\cdot)$ by utilizing the learned representation $\boldsymbol{H}$ as follows:

$$\begin{aligned} \mathrm{P}^{X}_{\text{cross}}(x_{t+1}|[y_1, x_1, x_2, \ldots, x_t]) &= \texttt{Softmax}\left(\boldsymbol{h}_t \mathbf{W}^X\right)_{x_{t+1}}, \\ \mathrm{P}^{Y}_{\text{cross}}(y_{t+1}|[y_1, x_1, x_2, \ldots, x_t]) &= \texttt{Softmax}\left(\boldsymbol{h}_t \mathbf{W}^Y\right)_{y_{t+1}}, \end{aligned} \tag{8}$$

the learned representations. In detail, $\boldsymbol{H}^X$, $\boldsymbol{H}^Y$ are only used to predict the next single-domain item, and $\boldsymbol{H}$ aims to predict both domains items. Therefore, by optimizing Eq.(5) and Eq.(7), $\boldsymbol{H}^X$, $\boldsymbol{H}^Y$ are tended to encode single-domain user preference and $\boldsymbol{H}$ is encouraged to act as the cross-domain user preference.

3.3 Contrastive Infomax Objective

3.3.1 Single- and Cross- Domain Prototype Representations

to obtain single-domain prototype representations, the concrete procedures can be formulated as:

$$\boldsymbol{o}_{\text{single}}^{X} = \text{Mean}(\boldsymbol{H}^{X}), \qquad \boldsymbol{o}_{\text{single}}^{Y} = \text{Mean}(\boldsymbol{H}^{Y}), \tag{9}$$

where the $\boldsymbol{o}_{\text{single}}^{X} \in \mathbb{R}^{1\times d}$ and $\boldsymbol{o}_{\text{single}}^{Y} \in \mathbb{R}^{1\times d}$ are the X-domain and Y-domain prototype representations of S^X and S^Y, respectively.

For the cross-domain interaction sequence S, we generate two cross-domain prototype representations by average corresponding domain item representations:

$$\boldsymbol{o}_{\text{cross}}^{X} = \text{Mean}(\{\boldsymbol{h}_t : S_t \in \mathcal{X}\}), \quad \boldsymbol{o}_{\text{cross}}^{Y} = \text{Mean}(\{\boldsymbol{h}_t : S_t \in \mathcal{Y}\}), \tag{10}$$

where the $\boldsymbol{o}_{\text{cross}}^{X}$ and $\boldsymbol{o}_{\text{cross}}^{Y}$ are the cross-domain prototype representations for domain X and Y, which encode the user's holistic preferences from both domains information.

3.3.2 Infomax Objective.

After obtaining the prototype representations, we follow the intuition from DIM and use a noisecontrastive objective between the samples from joint (positive examples) and the product of marginals (negative examples) to formulate our contrastive infomax objective. To generate negative samples from the positive cross-domain interaction sequence S, we devise two corruption functions

$$\text{Corrupt}^{X} \text{ and } \text{Corrupt}^{Y}$$

with the negative sampling trick for domain X and Y respectively. The detail of our corruption functions can be performed as:

$$\begin{aligned} \widehat{S}^{X} &= \text{Corrupt}^{X}(S) = [\widehat{y}_1, x_1, x_2, \widehat{y}_2, \dots], \\ \widehat{S}^{Y} &= \text{Corrupt}^{Y}(S) = [y_1, \widehat{x}_1, \widehat{x}_2, y_2, \dots], \end{aligned} \tag{11}$$

where $\widehat{x}$ and $\widehat{y}$ are randomly selected items in corresponding domain, and $\widehat{S}^X$ and $\widehat{S}^Y$ are the corrupted cross-domain interaction sequences. By Eq.(4) and Eq.(10), we produce the prototype representations $\widehat{\boldsymbol{o}}^X_{\text{cross}}$ and $\widehat{\boldsymbol{o}}^Y_{\text{cross}}$ for $\widehat{S}^X$ and $\widehat{S}^Y$ respectively.

we hope single-X-domain representation $\boldsymbol{o}^X_{\text{single}}$ is relevant to $\boldsymbol{o}^Y_{\text{cross}}$ with true X domain items, but is irrelevant to $\widehat{\boldsymbol{o}}^Y_{\text{cross}}$ with false X domain items. Such a design not only enforces correlations between two domains, but also makes cross-domain representations (e.g., $\boldsymbol{o}^Y_{\text{cross}}$) focusing on the real sequential interaction in the domain X. Thereby, our infomax objective $\mathcal{L}^X_{\text{disc}}$ is defined as (note that also holds for domain Y and leads to $\mathcal{L}^Y_{\text{disc}}$):

$$\mathcal{L}^X_{\text{disc}} = \sum_{(S^X, S^Y, S)_u \in \mathcal{S}} -\Big(\log \mathcal{D}^X(\boldsymbol{o}^X_{\text{single}}, \boldsymbol{o}^Y_{\text{cross}}) + \log\big(1 - \mathcal{D}^X(\boldsymbol{o}^X_{\text{single}}, \widehat{\boldsymbol{o}}^Y_{\text{cross}})\big)\Big) \tag{12}$$

where the $\mathcal{D}_X$ can be regarded as a binary discriminator to measure single- and cross- domain prototype representation pairs by a bilinear mapping function:

$$\mathcal{D}^X(\boldsymbol{o}^X_{\text{single}}, \boldsymbol{o}^Y_{\text{cross}}) = \sigma(\boldsymbol{o}^X_{\text{single}} \mathbf{W}^X_{\text{disc}} (\boldsymbol{o}^Y_{\text{cross}})^\top), \tag{13}$$