Self-Supervised Learning for Recommender Systems: A Survey

Junliang Yu, Hongzhi Yin*, Xin Xia, Tong Chen, Jundong Li, and Zi Huang

Abstract—Neural architecture-based recommender systems have achieved tremendous success in recent years. However, when dealing with highly sparse data, they still fall short of expectation. Self-supervised learning (SSL), as an emerging technique to learn with unlabeled data, recently has drawn considerable attention in many fields. There is also a growing body of research proceeding towards applying SSL to recommendation for mitigating the data sparsity issue. In this survey, a timely and systematical review of the research efforts on self-supervised recommendation (SSR) is presented. Specifically, we propose an exclusive definition of SSR, on top of which we build a comprehensive taxonomy to divide existing SSR methods into four categories: contrastive, generative, predictive, and hybrid. For each category, the narrative unfolds along its concept and formulation, the involved methods, and its pros and cons. Meanwhile, to facilitate the development and evaluation of SSR models, we release an open-source library SELFRec, which incorporates multiple benchmark datasets and evaluation metrics, and has implemented a number of state-of-the-art SSR models for empirical comparison. Finally, we shed light on the limitations in the current research and outline the future research directions.

Index Terms—Recommendation, Self-Supervised Learning, Contrastive Learning, Pre-Training, Data Augmentation.

1 Introduction

As the tool that can discover users' latent interests and ease the way of decision-making, recommender systems [1] have been widely deployed on a variety of online E-commerce platforms, for creating a delightful user experience whilst driving incremental revenue. In recent years, powered by highly expressive deep neural architectures, modern recommender systems [2], [3], [4] have achieved tremendous success and yielded unparalleled performance. However, deep recommendation models are inherently data-hungry. To take advantage of the deep architecture, an enormous amount of training data is required. Unlike the image annotation that can be undertaken by the crowdsourcing, data acquisition in recommender systems is costly since personalized recommendations rely on the data generated by users themselves, while most users normally can only consume/click a tiny fraction of countless items [5]. Consequently, the data sparsity issue bottlenecks deep recommendation models to reach their full potential [6].

Self-supervised learning (SSL) [7], emerging as a learning paradigm that can reduce the dependency on manual labels and enables training on massive unlabeled data, recently has received considerable attention. The essential idea of SSL is to extract informative and transferrable knowledge from abundant unlabeled data through well-designed pretext tasks (i.e. self-supervised tasks), in which the supervision signals are semi-automatically generated. Due to the ability to overcome the pervasive label insufficiency problem, SSL has been applied to a wide range of domains including visual representation learning [8], [9], [10], language model pre-training [11], [12], audio representation learning [13], node/graph classification [14], [15], etc, and it has turned out to be a powerful technique. As the principle of SSL is well-aligned with recommender systems' needs for more annotated data, motivated by the immense success of SSL in the aforementioned domains, a large and growing body of research is now proceeding towards applying SSL to recommendation.

The early prototypes of self-supervised recommendation (SSR) can be traced back to unsupervised methods like autoencoder-based recommendation models [16], [17], which rely on different corrupted data to reconstruct the original input to avoid overfitting. Afterwards, SSR appeared as the network embedding-based recommendation models [18], [19], in which the random-walk proximity is used as the self-supervision signals to capture the similarities between users and items. Over the same period, some generative adversarial networks [20] (GANs)-based recommendation models [21], [22], which augment the user-item interactions, can be seen as another embodiment of SSR. After the pre-trained language model BERT [12] made a huge breakthrough in 2018, SSL as an independent concept, came into the spotlight. Then the recommendation community started to embrace SSL and the subsequent research [23], [24], [25] shifted attention to pre-training recommendation models with the Cloze-like tasks on sequential data. Since 2020, SSL has enjoyed a period of prosperity and the latest SSL-based methods have almost performed on par with their supervised counterparts in many CV and NLP tasks [9], [26]. Particularly, the resurgence of contrastive learning (CL) [27] significantly pushes forward the frontier of SSL. In the meantime, a flurry of enthusiasm on SSR has also been witnessed [28], [29], [30], [31], [32], [33]. The paradigms of SSR become diverse and the scenarios are no longer limited

- *J. Yu, H. Yin, X. Xia, T. Chen, and Zi Huang are with the School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Queensland, Australia. E-mail: {jl.yu, h.yin1, x.xia, tong.chen, helen.huang}@uq.edu.au*
- *J. Li is with Department of Electrical and Computer Engineering, Department of Computer Science, and School of Data Science, University of Virginia, Charlottesville, Virginia, US. E-mail: jundong@virginia.edu*

*Corresponding author.

to sequential recommendation.

Although there have been several surveys on SSL in the fields of CV, NLP [34], [7] and graph learning [35], [36], [37], the research efforts of SSR have not been systematically investigated despite the fast-growing number of papers. Unlike the aforementioned fields, recommendation involves a plethora of scenarios that differ in optimization objectives and deal with multiple types of data, making it difficult to perfectly generalize the ready-made SSL methods designed for CV, NLP and graph tasks to recommendation. As such, it provides soil for new-type SSL. Meanwhile, the problems like the highly-skewed data distribution [38], widely observed biases [39], and large-vocabulary categorical features [40], which are unique to recommender systems, also spur a series of distinctive SSR methods that can enrich the SSL family. With SSR getting increasingly prevailing, a timely and systematic survey is urgently needed, to summarize the attainments and discuss the advantages and shortcomings of existing research efforts on SSR for promoting future research. To this end, we present an up-to-date and comprehensive retrospective on the frontier of SSR. In summary, our contributions are fourfold:

- We survey a broad spectrum of SSR methods to cover as many related papers as possible. To the best of our knowledge, this is the first survey focusing on this new topic.
- We provide an exclusive definition of SSR and clarify its connections to related concepts. On top of that, we propose a comprehensive taxonomy that divides existing SSR methods into four categories: contrastive, generative, predictive, and hybrid. For each category, the narrative unfolds along its concept and formulation, the involved methods, and its pros and cons. We believe the definition and the taxonomy provide a clear design space for developing and customizing new SSR methods.
- We introduce an open-source library to facilitate the implementation and evaluation of SSR models. It incorporates multiple benchmark datasets and evaluation metrics, and has implemented 10+ state-of-the-art SSR methods for empirical comparison.
- We shed light on the limitations in the existing research, and identify the remaining challenges and future directions to advance SSR.

Paper collection. In this survey, we review over 60 papers which purely concentrate on SSR and were released after 2018. As for the early implementations of SSR such as autoencoder-based and GAN-based models, they have been included and thoroughly discussed in previous surveys on deep learning [6], [41] and adversarial training [42], [43]. Therefore, we will not revisit them in the ensuing chapters. When retrieving related papers, DBLP and Google Scholar were used as the main search engines with the key words: self-supervised + recommendation, contrastive + recommendation, augmentation + recommendation, and pre-training + recommendation. We then traversed the citation graph of the identified papers and retained the related papers. Particularly, we kept a close watch on the top-tier conferences/journals such as ICDE, CIKM, ICDM, KDD, WWW, SIGIR, WSDM, AAAI, IJCAI, TKDE, TOIS etc, to avoid missing out high-quality work. Apart from these published papers, we also screened the preprints on arXiv, and sorted out those with novel and interesting ideas for a more inclusive panorama.

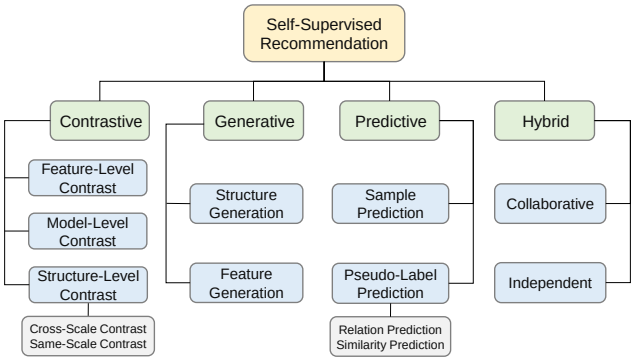


Fig. 1: The taxonomy of self-supervised recommendation.

Connections to existing surveys. Although there are some surveys on graph SSL [35], [36], [44] that include a few papers on recommendation, they just take those works as the supplementary applications of graph SSL. Another relevant survey [45] pays attention to the pre-training of recommendation models. However, its focus is transferring knowledge between different domains by exploiting knowledge graphs, and it only covers a small number of BERT-like works that are pre-trained in a self-supervised manner. Compared with them, our survey purely centers on recommendation-specific SSL and is the first one to provide a systematic review of a large number of up-to-date papers in this line of research.

Readers. This survey will benefit the following researchers and practitioners in the recommendation community: 1) who are new to SSR and want to fast step into this area; 2) who lost in a bewildering array of self-supervised approaches, and need a bird's-eye view of this realm; 3) who want to keep up with the latest progress of SSR; and 4) who are developing self-supervised recommender systems and seeking guidances.

Survey Structure. The remainder of this survey is structured as follows. In section 2 we begin with a definition and a formulation of SSR, and then present the taxonomy distilled from surveying a large number of research papers. Section 3 introduces the commonly used data augmentation approaches. Section 4-7 respectively review four categories of SSR models and demonstrate their pros and cons. Section 8 introduces the open-source framework **SELFRec** which facilitates the implementation and comparison of SSR methods. Section 9 discusses the limitations in current research and identifies some promising directions for inspiring future research. Finally, section 10 concludes this paper.

2 Definition and Taxonomy

In this section, we first define and formalize SSR. Then we lay out a comprehensive taxonomy to categorize existing SSR methods into four paradigms based on the features of their pretext tasks (Fig. 1). At last, we introduce three typical training schemes of SSR.

2.1 Preliminaries

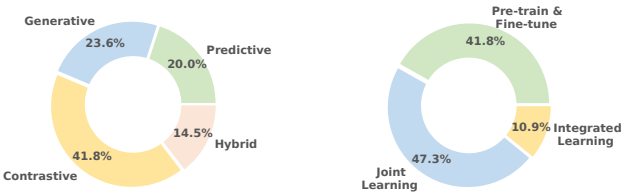
The current research of SSR mainly exploits the graph and sequential data, where the original user-item interactions are modeled as bipartite graphs and item sequences in chronological order, respectively. In the scenario of graph-based recommendation, we let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote the user-item bipartite graph where $\mathcal{V}$ is the node set (i.e. users $\mathcal{U}$ and items $\mathcal{I}$), and $\mathcal{E}$ is the edge set (i.e. interactions). The graph structure is represented with the adjacency matrix $\mathbf{A}$ where $\mathbf{A}_{ui} = 1$ denotes that node u and node i are connected. In the scenario of sequence-based next item recommendation, we let $\mathcal{I} = [i_1, i_2, ..., i_n]$ denote the item set. The behaviors of each user are often modeled as an ordered sequence $S_u = [i_1^u, i_2^u, ..., i_k^u], (1 \leq k \leq n)$, and $\mathcal{S} = \{S_1, S_2, ..., S_m\}$ refers to the whole dataset. In some cases, the users and items are associated with their attributes. We use $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_{m+n}]$ to denote the attribute matrix where $\mathbf{x}_i \in \mathbb{R}^t$ is the multi-hot vector representing object i's attributes. The general purpose of recommendation models is to learn quality representations $\mathbf{H} \in \mathbb{R}^{(m+n)\times d} = [\mathbf{U}, \mathbf{V}]$ for the users and items to generate satisfactory recommendation results, where d is the dimension of representations. To facilitate the reading, in this paper, matrices are denoted by capital letters, vectors appear in bold lowercase letters, sets are represented with italic capital letters.

2.2 Definition of Self-Supervised Recommendation

SSL provides a new way to conquer the data sparsity issue in recommendation. However, there has been no clear definition of SSR. Referring to the definition of SSL in other fields [7], [35] and how SSL functions in the collected literature on recommendation, we summarize the essential features of SSR as:

(i) Obtain more supervision signals by semi-automatically exploiting the raw data itself.
(ii) Have a pretext task(s) to (pre-)train the recommendation model with the augmented data.
(iii) Recommendation task is the unique primary task and the pretext task plays a secondary role to enhance recommendation.

Among these features, (i) is fundamental and implies the **scope** of SSR. (ii) states the **setup** of SSR which makes SSR distinct from other recommendation paradigms. (iii) indicates the **relation** between the recommendation task and the pretext task. Given the definition, we can clarify the difference between pre-training-based recommendation models [45] and SSR models. The former is often misunderstood as a branch of or being synonymous with the latter because in the fields of CV and NLP, pre-training has become a standard technique in SSL. However, some pre-training-based recommendation methods [46], [47] are purely supervised where there is no data augmentation and they request extra human annotated side information for pre-training. Therefore, the two paradigms are just partially overlapped. Analogously, contrastive learning (CL) [27] based recommendation is often considered a subset of SSR. However, CL can be applied to both supervised and unsupervised settings, and those CL-based recommendation methods which do not augment the raw data [48], [49], [50] and just optimize a marginal loss [51], [52], should not be roughly classified into SSR either.



(a) Split by pretext tasks. (b) Split by training schemes.

Fig. 2: Distributions of self-supervised recommendation.

Due to the diversity of data and optimization objectives in recommender systems, to formally define SSR, a model-agnostic framework is needed. Although the concrete structures and the number of used encoders and projection heads vary from situation to situation, most existing models in general can be sketched into an **Encoder** + **Projection-Head** architecture. To deal with different data such as graphs, sequences, and categorical features, a wide range of neural networks like GNNs [53], Transformers [54], and MLPs can be the candidate structure of the encoder f_θ, while the projection head g_ϕ (a.k.a. decoder in the generative models) usually has a lightweight structure. It can be a linear transformation, a nonlinear mapping (e.g., shallow MLP) or even a non-parametric identical mapping. The encoder f_θ generally aims to learn distributed representations $\mathbf{H}$ for users and items. The projection head g_ϕ refines $\mathbf{H}$ for the recommendation task or a specific pretext task. Based on this architecture, SSR can be formulated as:

$$f_{\theta^*}, g_{\phi^*}, \mathbf{H}^* = \arg\min_{f_\theta, g_\phi} \mathcal{L}\left(g_\phi(f_\theta(\mathcal{D}, \tilde{\mathcal{D}}))\right), \quad (1)$$

where $\mathcal{D}$ denotes the original data, $\tilde{\mathcal{D}}$ refers to the augmented data that satisfies $\tilde{\mathcal{D}} \sim \mathcal{T}(\mathcal{D})$, $\mathcal{T}(\cdot)$ denotes the augmentation module, and $\mathcal{L}$ is the merged loss function that can be divided into the loss of the recommendation task $\mathcal{L}_{rec}$ and the loss of the pretext task $\mathcal{L}_{ssl}$. By minimizing Eq. (1), the optimal encoder(s) f_{θ^*}, projection header(s) g_{ϕ^*}, and representations $\mathbf{H}^*$ can be learned for generating quality recommendation results.

2.3 Taxonomy of Self-Supervised Recommendation Methods

As the key ingredient of SSR, the pretext task makes SSR distinct from other recommendation paradigms. According to the characteristics of the pretext tasks, we divide the existing SSR models into four categories: contrastive, predictive, generative, and hybrid.

2.3.1 *Contrastive Methods*

Driven by CL [27], contrastive methods have become the dominant branch in SSR (shown in Fig. 2(a)). The fundamental idea behind contrastive methods is to treat every instance (e.g., user/item/sequence) as a class, and then pull views of the same instance closer in the embedding space, and push views of different instances apart, where the views are created by imposing different transformations on the

original data. Generally, two views of the same instance are considered a positive pair, and views of different instances are considered negative samples of each other. A positive view is supposed to introduce the non-essential variations into examples without modifying semantic meanings. By maximizing the consistency between positive pairs while minimizing the agreement between negative pairs, we can obtain discriminative and generalizable representations for recommendation. Formally, the pretext task of the contrastive SSR methods (Fig. 3(a)) can be defined as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\Big(g_{\phi_s}(f_\theta(\tilde{\mathcal{D}}_1), f_\theta(\tilde{\mathcal{D}}_2))\Big), \tag{2}$$

where $\tilde{\mathcal{D}}_{(1)} \sim \mathcal{T}_1(\mathcal{D})$ and $\tilde{\mathcal{D}}_{(2)} \sim \mathcal{T}_2(\mathcal{D})$ are two differently augmented views of $\mathcal{D}$, $\mathcal{T}_1(\cdot)$ and $\mathcal{T}_2(\cdot)$ are augmentation operators (e.g. node dropout and item reordering). The loss function $\mathcal{L}_{ssl}$ estimates the mutual information (MI) between views through the representations learned by a shared encoder f_θ, where the projection head g_{ϕ_s} is usually with a light structure such as a bilinear network.

2.3.2 Generative Methods

Generative methods are inspired by the masked language models (MLM) [12] whose pretext task is to reconstruct the original user/item profile with its corrupted versions (Fig. 3(b)). In other words, the model learns to predict a portion of the available data from the rest. The most common tasks are the structure reconstruction (e.g., masked item prediction) and the feature reconstruction. In these situations, the pretext task is formulated as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\Big(g_{\phi_s}(f_\theta(\tilde{\mathcal{D}})), \mathcal{D}\Big), \tag{3}$$

where $\tilde{\mathcal{D}} \sim \mathcal{T}(\mathcal{D})$ denotes the corrupted version of the original input. For most of the generative SSR methods, the objective function $\mathcal{L}_{ssl}$ is often instantiated as either the Cross Entropy (CE) loss or the Mean Squared Error (MSE) which estimates the probability distribution/values of the masked items/numerical features.

2.3.3 Predictive Methods

Predictive methods often appear to be like generative methods because they both have the action of predicting. However, in generative methods, the goal is to predict the missing part that originally exists, which can be seen as self-prediction. While in predictive methods, new samples or labels are generated from the original data in order to guide the pretext task. We further categorize existing predictive SSR methods into two types: sample-based and pseudo-label-based (Fig. 3(c)). The former focuses on predicting informative samples based on current parameters of the encoder, and then these samples are fed to the encoders again so as to generate new samples with higher confidence [55], [56]. In this way, self-training (a flavor of semi-supervised learning) and SSL are connected. The pseudo-label-based branch yields labels through a generator which can be another encoder or rule-based selectors. The generated labels are then used as the ground-truth to guide the encoder f_θ. The pseudo-label-based branch can be formulated as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\Big(g_{\phi_s}(f_\theta(\mathcal{D})), \tilde{\mathcal{D}}\Big), \tag{4}$$

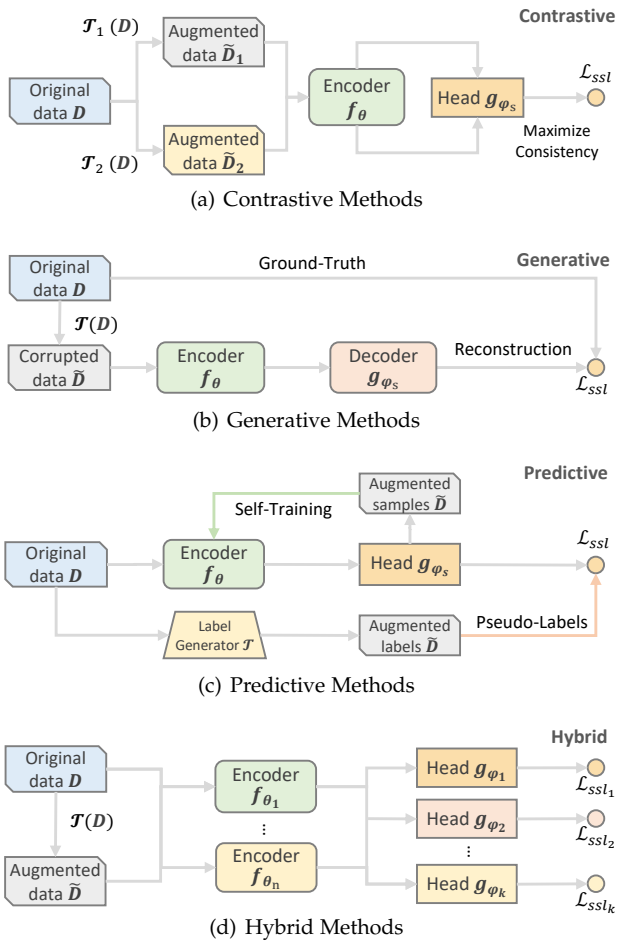


Fig. 3: Four common paradigms of self-supervised recommendation.

where $\tilde{\mathcal{D}} \sim \mathcal{T}(\mathcal{D})$ denotes the generated labels, and $\mathcal{L}_{ssl}$ often appears in forms of the CE/softmax or MSE. The former aligns the predicted probability with the labels and the latter measures the difference between the output of g_{ϕ_s} and the labels, which correspond to the classification problem and the regression problem, respectively.

2.3.4 Hybrid Methods

Every type of the aforementioned pretext tasks has its own advantage and can utilize different self-supervision signals. A natural way to obtain comprehensive self-supervision is combining different pretext tasks and integrating them into one recommendation model. In these hybrid methods, more than one encoder and projection head are probably needed (3(d)). Different pretext tasks either work independently or collaborate to intensify the self-supervision signals. The combination of different types of pretext tasks is usually formulated as a weighted sum of different self-supervised losses presented in the aforementioned categories.

2.4 Typical Training Schemes of Self-Supervised Recommendation

Despite the unified formulation of SSR (Eq. (1)), the recommendation task is coupled with the pretext task in more concrete fashions in different scenarios. In this section, we

introduce three typical training schemes of SSR: Joint Learning (JL), Pre-training and Fine-tuning (PF), and Integrated Learning (IL). Brief pipelines of them are given in Figure 4.

2.4.1 Joint Learning (JL)

According to Fig. 2(b), nearly half of the collected SSR methods prefer the JL training scheme. In this scheme, the pretext and the recommendation task are usually jointly optimized with a shared encoder (Fig. 4(a)). A trade-off between the two objectives $\mathcal{L}_{ssl}$ and $\mathcal{L}_{rec}$ is reached by tuning the hyper-parameter α which controls the magnitude of self-supervision. Although the JL scheme can be considered a kind of multi-task learning, the result of the pretext task is generally not concerned and it is regarded as an auxiliary task that helps to regularize the recommendation task. The formulation of the JL scheme is defined as follows:

$$f_{\theta^*}, g_{\phi_r^*}, \mathbf{H}^* = \arg\min_{f_\theta, g_\phi} [\mathcal{L}_{rec}(g_{\phi_r}(f_\theta(\mathcal{D}))) + \alpha \mathcal{L}_{ssl}(g_{\phi_s}(f_\theta(\tilde{\mathcal{D}})))] \tag{5}$$

The JL scheme is mostly used in the contrastive methods.

2.4.2 Pre-training and Fine-tuning (PF)

The PF scheme is the second most prevalent training scheme, which consists of two stages: pre-training and fine-tuning (Fig. 4(b)). In the first stage, the encoder f_θ is pre-trained with pretext tasks on the augmented data for a good initialization of the encoder's parameters. After that, the pre-trained encoder $f_{\theta_{init}}$ is fine-tuned on the original data followed by a projection head g_{ϕ_r} for the recommendation task. The previous surveys on graphs [35], [37] introduce another training scheme named unsupervised representation learning which first pre-trains the encoder with the pretext task and then freezes the encoder and only learns a linear head for the downstream tasks. We consider it a special case of the PF scheme and it is rarely used in the scenario of recommendation. The formulation of the PF scheme is defined as follows:

$$\begin{aligned} f_{\theta_{init}} &= \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}(g_{\phi_s}(f_\theta(\tilde{\mathcal{D}}), \mathcal{D})) \\ f_{\theta^*}, g_{\phi_r^*}, \mathbf{H}^* &= \arg\min_{f_{\theta_{init}}, g_{\phi_r}} \mathcal{L}_{rec}(g_{\phi_r}(f_\theta(\mathcal{D}))) \end{aligned} \tag{6}$$

The PF scheme is often applied to train the BERT-like generative SSR models which are pre-trained on the mask-based sequence augmentation and are then fine-tuned on the interaction data. Some contrastive methods also employ this training scheme where the contrastive pretext task is for pre-training.

2.4.3 Integrated Learning (IL)

Compared with the JL and PF schemes, the IL scheme receives less attention and is not extensively used. The pretext task and the recommendation task are well aligned under this setting and they are unified into an integrated objective. The loss $\mathcal{L}$ usually measures the difference or the mutual information between the two outputs. The IL scheme can be formalized as follows:

$$f_{\theta^*}, g_{\phi_r^*}, \mathbf{H}^* = \arg\min_{f_{\theta_r}, g_\phi} \mathcal{L}\left(g_{\phi_r}(f_\theta(\mathcal{D})), g_{\phi_s}(f_\theta(\tilde{\mathcal{D}}))\right) \tag{7}$$

The IL scheme is mainly used by the pseudo-labels-based predictive methods and a few contrastive methods.

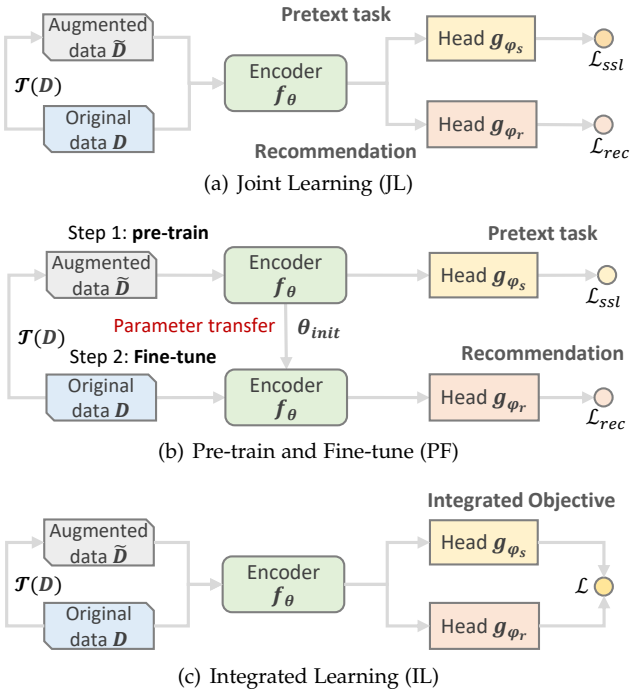


Fig. 4: Three typical training schemes of self-supervised pretext tasks.

3 Data Augmentation

Previous efforts in multiple fields on SSL [57], [58], [59] have demonstrated that data augmentation plays a pivotal role in learning quality and generalizable representations. Before detailing the SSR methods, we summarize the commonly used data augmentation approaches in SSR and divide them into three categories: sequence-based, graph-based, and feature-based. Most of these augmentation approaches are task-independent and model-agnostic which have been used across different paradigms of SSR. For those task- and model-dependent approaches, we will introduce them along with the concrete SSR methods in Section 4-7.

3.1 Sequence-Based Augmentation

Given a sequence of items $S = [i_1, i_2, ..., i_k]$ which is a user's historic behaviors, the common sequence-based augmentations $\tilde{S} = \mathcal{T}(S)$ (shown in Fig. 5(a)) are listed as follows.

Item Masking. Analogous to the word masking in BERT [12], the item masking strategy (a.k.a. item dropout) [60], [25], [61], [62], [63] randomly masks a portion γ of items and replaces them with special tokens [mask]. The idea behind is that a user's intention is relatively stable during a period of time. Therefore, though part of items are masked, the primary intent information is still retained in the rest. Let $\mathcal{M}$ denote the indices of the masked items. This augmentation strategy can be formulated as:

$$\tilde{S} = \mathcal{T}_{\text{masking}}(S) = [\tilde{i}_1, \tilde{i}_2, ..., \tilde{i}_k], \tilde{i}_t = \begin{cases} i_t, & t \notin \mathcal{M} \\ [\text{mask}], & t \in \mathcal{M} \end{cases} \tag{8}$$

Item Cropping. Inspired by the image cropping in CV, some works [61], [25], [62], [60] propose the item cropping augmentation method (a.k.a. sequence splitting). Given a

user's historic sequence S, a continuous sub-sequence with length $L_c = \lfloor \eta * |S| \rfloor$ is randomly chosen, where $\eta \in (0, 1)$ is a coefficient that adjusts the length. The item cropping augmentation can be formulated as:

$$\tilde{S} = \mathcal{T}_{\text{cropping}}(S) = [\tilde{i}_c, \tilde{i}_{c+1}, \ldots, \tilde{i}_{c+L_c-1}] \tag{9}$$

This method provides a local view of the user's historic sequence. Through the self-supervised task, the selected subsequence is expected to endow the model with the ability to learn generalized representations without the comprehensive user profile.

Item Reordering. Many sequential recommenders [64], [65], [68] assume that the item order in a sequence is strict so that the item transitions are sequentially dependent. However, this assumption is probably problematic because in the real world many unobserved external factors can impact the item order [66] and different item orders may actually correspond to the same user intent. Some works [61], [60] propose to shuffle a continuous sub-sequence $[i_r, i_{r+1}, ..., i_{r+L_r-1}]$ to $[\tilde{i}_r, \tilde{i}_{r+1}, ..., \tilde{i}_{r+L_r-1}]$ to create sequence augmentations, which can be formulated as:

$$\tilde{S} = \mathcal{T}_{\text{reordering}}(S) = [i_1, ..., \tilde{i}_r, \tilde{i}_{r+1}, ..., \tilde{i}_{r+L_r-1}, ..., i_k] \tag{10}$$

Item Substitution. Random item cropping and masking could exaggerate the data sparsity issue in short sequences. [67] proposes to substitute items in short sequences with highly correlated items, which injects less corruption to the original sequential information. Given $\mathcal{Z}$ denoting the indices of the randomly selected items which are going to be substituted, the item substitution augmentation [68], [67] method is formulated as:

$$\begin{aligned} \tilde{S} &= \mathcal{T}_{\text{substitution}}(S) = [\tilde{i}_1, \tilde{i}_2, ..., \tilde{i}_k], \\ \tilde{i}_t &= \begin{cases} i_t, & t \notin \mathcal{Z} \\ \text{item correlated to } i_t, & t \in \mathcal{Z} \end{cases} \end{aligned} \tag{11}$$

where the correlated item is obtained by calculating the correlation score which is based on the item co-occurrence or the similarity of the corresponding representations.

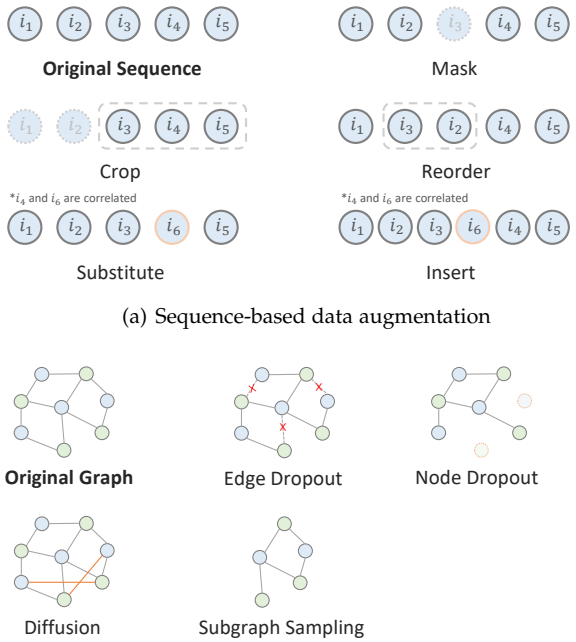
Item Insertion. In short sequences, recorded interactions fail to track comprehensive user dynamics and item correlations. Therefore, [67] also proposes to insert correlated items into short sequences to complete the sequence. l items $[id_1, id_2, ..., id_l]$ are firstly selected from the given sequence at random, and then their highly correlated items are inserted around them. After the insertion, the augmented sequence with length $l + k$ is:

$$\tilde{S} = \mathcal{T}_{\text{insert}}(S) = [i_1, ..., \tilde{i}_{id_1}, i_{id_1}, ..., \tilde{i}_{id_l}, i_{id_l} ..., i_k] \tag{12}$$

3.2 Graph-Based Augmentation

Given the user-item graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with the adjacency matrix $\mathbf{A}$ (or other graphs such as the user-user graph and item-attribute graph), the following graph augmentation approaches (illustrated in Fig. 5(b)) can be applied.

Edge/Node Dropout. With the probability ρ, each edge may be removed from the graph. The idea behind is that only partial connections contribute to the node representations, and discarding the redundant connections can endow



(a) Sequence-based data augmentation

(b) Graph-based data augmentation

Fig. 5: Data augmentations on sequences and graphs.

the representations more robustness against noisy interactions, which is analogous to the item masking and cropping methods. This edge dropout method is formulated as:

$$\tilde{\mathcal{G}}, \tilde{\mathbf{A}} = \mathcal{T}_{\text{E-dropout}}(\mathcal{G}) = (\mathcal{V}, \mathbf{m} \odot \mathcal{E}), \tag{13}$$

where $\mathbf{m} \in (0, 1)^{|\mathcal{E}|}$ is the masking vector on the edge set generated by a Bernoulli distribution. This method is widely used in many contrastive methods [31], [69], [30], [70], [56], [71]. Similarly, with the probability ρ, each node can also be dropped from the graph, together with its associated edges [72], [30], [70]. This augmentation method is expected to identify the influential nodes from differently augmented views, which is formulated as:

$$\tilde{\mathcal{G}}, \tilde{\mathbf{A}} = \mathcal{T}_{\text{N-dropout}}(\mathcal{G}) = (\mathcal{V} \odot \mathbf{m}, \mathcal{E} \odot \mathbf{m}'), \tag{14}$$

where $\mathbf{m} \in (0, 1)^{|\mathcal{V}|}$ is the masking vector on the node set and $\mathbf{m}'$ is the vector masking the associated edges.

Graph Diffusion. Opposite to the dropout-based method, the diffusion-based augmentation adds edges into the graph to create views. [73] considers that the missing user behaviors include unknown positive preferences which can be represented with weighted user-item edges. Therefore, they discover the possible edges by calculating the similarities of the user and item representations and retain the edges with top-K similarities. The diffusion method is formulated as:

$$\tilde{\mathcal{G}}, \tilde{\mathbf{A}} = \mathcal{T}_{\text{diffusion}}(\mathcal{G}) = (\mathcal{V}, \mathcal{E} + \tilde{\mathcal{E}}) \tag{15}$$

When the edges are randomly added, this method can also be used to generate negative samples.

Subgraph Sampling. This method samples a portion of nodes and edges which form subgraphs and reflect the local connectivity. A lot of approaches can be used to induce subgraphs like meta-path guided random walks [74], [70],

and the ego-network sampling [29], [75], [76], [77]. The underlying idea of subgraph sampling is analogous to the idea of edge dropout whereas the subgraph sampling usually operates on local structures. Give the sampled node set $\mathcal{Z}$, this augmentation method can be formulated as:

$$\tilde{\mathcal{G}}, \tilde{\mathbf{A}} = \mathcal{T}_{\text{sampling}}(\mathcal{G}) = (\mathcal{S} \in \mathcal{V}, \mathbf{A}[\mathcal{Z}, \mathcal{Z}]) \tag{16}$$

3.3 Feature-Based Augmentation

The feature-based augmentations focus on the user/item features, operating in the attribute/embedding space. In this part, the categorical attributes and the learned continuous embeddings are collectively called features and denoted by $\mathbf{X}$ for brevity.

Feature Dropout. The feature dropout (a.k.a. feature masking) [25], [71], [78], [79], [33], [80] is similar to the item masking and edge dropout, which randomly masks/drops a small portion of features and is formulated as:

$$\tilde{\mathbf{X}} = \mathcal{T}_{\text{F-dropout}}(\mathbf{X}) = \mathbf{X} \odot \mathbf{M}, \tag{17}$$

where $\mathbf{M}$ is the masking matrix that $\mathbf{M}_{i,j} = 0$ if the j-th element of vector i is masked/dropped, otherwise $\mathbf{M}_{i,j} = 1$. The matrix $\mathbf{M}$ is generated by Bernoulli distribution.

Feature Shuffling The feature shuffling [29], [28], [81] switches rows and columns in the feature matrix $\mathbf{X}$. By randomly changing the contextual information, $\mathbf{X}$ is corrupted to yield augmentations. This method can be formulated as:

$$\tilde{\mathbf{X}} = \mathcal{T}_{\text{shuffling}}(\mathbf{X}) = \mathbf{P}_r \mathbf{X} \mathbf{P}_c, \tag{18}$$

where $\mathbf{P}_r$ and $\mathbf{P}_r$ are permutation matrices that have exactly one entry of 1 in each row and each column and 0s elsewhere.

Feature Clustering. This augmentation method [82], [83], [84] bridges CL with clustering, which assumes that there are prototypes in the feature/representation space and user/item representations should be closer to their assigned prototypes. The augmented prototype representations can be learned via clustering within the expectation-maximization (EM) framework. This method can be formulated as:

$$\tilde{\mathbf{C}} = \mathcal{T}_{\text{clustering}}(\mathbf{X}) = \text{EM}(\mathbf{X}, \mathcal{C}), \tag{19}$$

where $\mathcal{C}$ is the presupposed clusters (prototypes) and $\tilde{\mathbf{C}}$ is the augmented prototype representations.

Feature Mixing. This augmentation method [85], [71] mixes the original user/item features with features from other users/items or previous versions to synthesize informative negative/positive examples [86]. It usually interpolates two samples in the following way:

$$\tilde{\mathbf{x}}_i = \mathcal{T}_{\text{mixing}}(\mathbf{x}_i) = \alpha \mathbf{x}_i + (1 - \alpha)\mathbf{x}_j^{'}, \tag{20}$$

where $\alpha \in [0, 1]$ is the mixing coefficient that controls the proportion of information from $\mathbf{x}_i$.

4 Contrastive Methods

The flexible data augmentation approaches and various data types spawn a various forms of contrastive pretext tasks. According to where the self-supervision signals come from, we divide them into three categories: **structure-level contrast**, **feature-level contrast**, and **model-level contrast**. A summary of surveyed contrastive methods is presented in Table 1.

4.1 Structure-Level Contrast

The user behavior data is often organized as graphs or sequences. The graph/sequence structures with slight perturbations may have similar semantics. By contrasting different structures, the shared invariance to structural perturbations are obtained as self-supervision signals. We borrow the taxonomy from [35], [36] to subdivide the structure-level contrast into: **same-scale contrast** and **cross-scale contrast**. In the former category, the views to be contrasted are from two objects at the same scale. In the latter category, the views to be contrasted are from two objects at different scales. We further subdivide the same-scale contrast into two levels: **local-local** and **global-global**. Analogously, the cross-scale contrast is subdivided into: **Local-Global**, and **Local-Context**. For graph structures, *local* refers to nodes, and *global* refers to graphs. For sequence structures, *local* refers to items, and *global* refers to sequences.

4.1.1 Local-Local Contrast

This type of contrast is often conducted in the graph-based SSR models to maximize the mutual information between user/item node representations, which can be formulated as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\big(g_{\phi_s}(\tilde{\mathbf{h}}_i, \tilde{\mathbf{h}}_j)\big), \tag{21}$$

where $\tilde{\mathbf{h}}_i$ and $\tilde{\mathbf{h}}_j$ are node representations learned from two augmented views via the shared encoder f_θ, and $\mathcal{L}_{\mathcal{MI}}$ is the contrastive loss which will be introduced in Section 4.4.

For local level contrast, dropout-based augmentations are the most preferred methods to create perturbed local views. **SGL** [30], as a representative graph CL-based recommendation model, applies three types of stochastic graph augmentations: node dropout, edge dropout, and random walk (multi-layer edge dropout) on the user-item bipartite graph. It first generates two augmented graphs with the same type of augmentation operator. Then it applies a shared graph LightGCN encoder f_θ [87] to learn node embeddings from the augmented graphs. The node-level contrast is conducted by optimizing the InfoNCE loss [13] with in-batch negative sampling. Finally, SGL jointly optimizes the above InfoNCE loss and the Bayesian personalized ranking (BPR) loss [88] for recommendation. Similar to SGL, **DCL** [89] also employs stochastic edge dropout to perturb the L-hop ego-network of a given node, resulting in two augmented neighborhood subgraphs. And then it maximizes agreement between node representations learned on the two subgraphs.

Considering that dropping an important node may lead to highly-skewed local structures, **HHGR** [69] proposes a double-scale node dropout method to create effective self-supervised signals in the scenario of group recommendation [90]. The proposed coarse-grained dropout scheme removes a portion of users nodes from all groups and the fine-grained dropout scheme only drops the randomly selected member nodes from a specific group. The method then maximizes the mutual information between the user node representations learned from these two views with different dropout granularities.

The subgraph sampling is the other popular augmentation method in the local-level graph contrast. **CCDR** [76]

applies CL to cross-domain recommendation. It devises two types of contrastive tasks: intra-CL and inter-CL. The intra-CL task is almost the same as the contrastive task in DCL [89], which is conducted in the target domain and employs a graph attention network [91] as the encoder. The inter-CL task aims to maximize mutual information between representations of the same object learned in the source and target domain. A concurrent work **PCRec** [77] also connects cross-domain recommendation with CL. It samples r-hop ego-network with random walks to augment data. The contrastive task between sampled subgraphs pre-trains a GIN [92] encoder in the source domain. The parameters are then transferred to initialize a MF [93] model which is fine-tuned with the interaction data for recommendation in the target domain.

4.1.2 Global-Global Contrast

The global level contrast is often conducted in sequential recommendation models where a sequence is considered a user's global view, which can be formulated as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\big(g_{\phi_s}(\text{Agg}(f_\theta(\tilde{S}_i)), \text{Agg}(f_\theta(\tilde{S}_j)))\big), \quad (22)$$

where $\tilde{S}_i$ and $\tilde{S}_j$ are two sequence augmentations, and Agg is the aggregating function that synthesizes sequence representations.

CL4SRec [61] uses three random augmentation operators: item masking, item cropping, and item reordering to augment sequences. Given N sequences, CLS4Rec applies two augmentation operators and obtains $2N$ augmented sequences $\big[S'_{u_1}, S''_{u_1}, S'_{u_2}, S''_{u_2}, \ldots, S'_{u_N}, S''_{u_N}\big]$. Then it treats (S'_u, S''_u) as the positive pair, and treats other 2(N-1) augmented sequences within the mini-batch as negative samples. A Transformer-based encoder [54] is used to encode the augmented sequences and learn user representations for the global-level contrast. The very similar idea is also found in **H^{2}SeqRec** [62] where the sequence-level contrastive task is used to pre-train a Transformer-based encoder. Another two similar works **CoSeRec** [67] and **ContraRec** [94] also follow the pipeline of CL4SRec to contrast sequences. CoSeRec further proposes to substitute items in short sequences with correlated items or insert correlated items into short sequences for robust data augmentation. ContraRec not only contrasts sequences augmented from the same input but also considers sequences which have the same target item as positive pairs. In the scenario of session-based recommendation, **DHCN** [28] creates two hypergraph-based views of a given session by modeling the intra-session and inter-session structural information, respectively. It regards the representations of the same session as a positive pair, and regards the corrupted representations (obtained by feature shuffling in the embedding space) of different sessions as the negative samples.

4.1.3 Local-Global Contrast

The local-global contrast is expected to encode high-level global information into local structure representations and unify global and local semantics. It is often applied to the graph learning scenario, which can be formulated as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\big(g_{\phi_s}(\tilde{\mathbf{h}}, \mathcal{R}(f_\theta(\tilde{\mathcal{G}}, \tilde{\mathbf{A}}))\big), \quad (23)$$

where $\mathcal{R}$ is the readout function that generates global-level graph representation.

EGLN [73] proposes to reach a local-global consistency by contrasting the merged user-item pair representations with the global representation which is an average of all the user-item pair representations. To mine more self-supervision signals, it adopts graph diffusion for data augmentation. The augmented graph adjacency matrix is obtained by calculating the similarities between users and items and retaining top-K similarities. The matrix and the user/item representations iteratively learn from each other and get updated via a graph encoder. In **BiGI** [75], the similar local-global contrast is also performed. The difference is that, when generating the user-item pair representation, only its h-hop subgraph (i.e., any node v_k in this subgraph satisfies $\text{dist}(v_i, v_k) \leq h$ or $\text{dist}(v_j, v_k) \leq h$) is sampled for feature aggregation. In **HGCL** [95], the user and item node-type specific homogeneous graphs are constructed. For each homogeneous graph, it follows the pipeline of DGI [15] to maximize the mutual information between local patches of a graph and the global representation of the entire graph. Additionally, a cross-type contrast is proposed to measure local and global information across different types of homogeneous graphs.

4.1.4 Local-Context Contrast

The local-context contrast is observed in both the graph and sequence-based scenarios. The context is often constructed by sampling ego-network or clustering. This type of contrast can be formulated as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\big(g_{\phi_s}(\mathbf{h}_i, \mathcal{R}(f_\theta(\mathcal{C}_j))\big), \quad (24)$$

where $\mathcal{C}_j$ denotes the context of node (sequence) j.

NCL [83] follows [84] to design a prototypical contrastive objective to capture the correlations between a user/item and its prototype. The prototype can be seen as the context of each user/item which represents a group of semantic neighbors even that they are not structurally connected in the user-item graph. Regarding the prototype learning as a type of feature clustering for data augmentation, it obtains prototypes by clustering over all the user or item embeddings with K-means algorithm. Then the EM algorithm is used to recursively adjust the prototypes. **ICL** [82] has almost the same pipeline, and the only difference is that ICL is designed for sequential recommendation where the semantic prototypes in NCL are modeled as user intents in ICL, and the belonged sequence here is a local view of the prototype.

In social networks, users are usually similar to their topological contexts. **MHCN** [29] is the first work that applies SSL to social recommendation [96], [97]. It defines three types of triangle social relations and models them with a multi-channel hypergraph encoder. For each user in each channel, MHCN hierarchically maximizes the mutual information among the user representation, the user's ego hypergraph representation, and the global hypergraph representation. The follow-up work **SMIN** [98] inherits the idea in MHCN to contrast nodes with their contexts. The difference is that the context is obtained by aggregating

information from a chain of user-item adjacency matrices with different orders.

Besides, there are also methods which span different contrast levels. **S^3-Rec** [25] applies two operators: item masking and item cropping to augment sequences. It then devises four contrastive tasks: item-attribute mutual information maximization (MIM), sequence-item MIM, sequence-attribute MIM. and sequence-sequence MIM to pre-train a bidirectional Transformer for next-item prediction.

4.2 Feature-Level Contrast

Compared to the structure-level contrast, the feature-level contrast is relatively less explored, because the feature/attribute information is not always available in the datasets for academic use. However, in industry, the data is often organized in a multi-field format and a large number of categorical features such as the user profile and item category can be exploited. In general, this type of contrast can be formally defined as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\big(g_{\phi_s}(f_\theta(\tilde{\mathbf{x}}_i), f_\theta(\tilde{\mathbf{x}}_j))\big), \tag{25}$$

where $\tilde{\mathbf{x}}_i$ and $\tilde{\mathbf{x}}_j$ are feature-level augmentations which can be obtained by modifying the input feature or learned by models.

CFM [78] adopts a two-tower architecture and applies correlated feature masking and dropout on the item features for more meaningful feature augmentations. It seeks to mask highly correlated features together whose correlation is measured by mutual information. As a result, the contrastive task becomes difficult because the retained features can hardly remedy the semantics behind the masked features. **MISS** [99] argues that a user behavior sequence may contain multiple interests and directly perturbing the sequence will cause semantically dissimilar augmentations. Instead, it uses a CNN-based multi-interest extractor to transform the user sample that contains the behavior data and categorical features into a group of implicit interest representations, which augments the user sample at the feature-level. The contrastive task is then conducted on the extracted interest representations.

4.3 Model-Level Contrast

The former two categories extract self-supervision signals from the data perspective, which are not implemented in a fully end-to-end fashion. An alternative way is to keep the input unchanged and dynamically modify the model architecture so that the view pairs are augmented on-the-fly. The contrast between these model-level augmentations can be formally defined as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{\mathcal{MI}}\big(g_{\phi_s}(f_{\theta'}(\mathcal{D}), f_{\theta''}(\mathcal{D}))\big), \tag{26}$$

where $f_{\theta'}$ and $f_{\theta''}$ are perturbed versions of f_θ. This equation can be seen as a special case of Eq. (2) which augments the intermediate hidden representations of the same input.

The neuron masking is a commonly used technique to perturb the model. **DuoRec** [79] follows the successful experience in SimCSE [100] to apply two different sets of dropout masks to a Transformer-based backbone for two model-level representation augmentations. It then maximizes the mutual information between the two representations. Though it appears strikingly simple, this method shows significant performance in the next-item prediction task. Opposite to the neuron masking which discards some information in the hidden representations, **SimGCL** [101] directly adds random uniform noises to the hidden representations for augmentations. It experimentally proves that optimizing the InfoNCE loss in fact learns more uniform node representations and adjusting the noise magnitude can provide a finer-grained regulation of representation uniformity, which mitigates the popularity bias issue [39]. Benefiting from the noised-based augmentations, SimGCL shows distinct advantages over SGL on both recommendation accuracy and model training efficiency. In **SRMA** [102], apart from the neuron-level perturbations, it proposes to randomly drop a fraction of layers during training. Considering that dropping important layers would mislead the output, it randomly drops some layers of the feed-forward network in the Transformer for model-level augmentations. Moreover, it further introduces another pre-trained encoder which has the same architecture but trained with the recommendation task to generate different views for contrast.

Beyond the taxonomy, there are a few papers [103], [104] which claim that their proposed methods are contrastive self-supervised. They deal with the multi-behavior data, and conduct the data augmentation by incorporating auxiliary behavior data as supervision signals. Behavior views of the same user are considered positive pairs, and the views of different users are sampled as negative pairs. These methods then encourage consistency between representations of positive pairs by conducting the behavior-level contrast. However, we think it is farfetched to call them self-supervised, because they, in fact, do not transform the original data for creating any new views.

4.4 Contrastive Loss

The contrastive loss has become a new research hotspot in the field of CV, and is also drawing increasing attention in SSR. Generally, the optimization goal of the contrastive loss is to maximize the mutual information (MI) between two representations (views) $\mathbf{h}_i$ and $\mathbf{h}_j$ which is given by:

$$\mathcal{MI}\left(\mathbf{h}_i, \mathbf{h}_j\right) = \mathbb{E}_{P(\mathbf{h}_i, \mathbf{h}_j)} \log \frac{P\left(\mathbf{h}_i, \mathbf{h}_j\right)}{P\left(\mathbf{h}_i\right) P\left(\mathbf{h}_j\right)} \tag{27}$$

However, directly maximizing MI is difficult and a practical way is to maximize its lower bound. In this section, we review two most commonly used lower bounds: Jensen-Shannon Estimator [105] and Noise-Contrastive Estimator (a.k.a. InfoNCE [13]).

4.4.1 *Jensen-Shannon Estimator*

As a MI estimator for SSL, Jensen-Shannon divergence (JSD) first appears in DGI [15]. Compared to the Donsker-Varadhan estimator [106] which provides a tight lower bound of MI, JSD can be more efficiently optimized and guarantees a stable performance (we refer you to [107] for

TABLE 1: A summary of the surveyed papers on contrastive self-supervised recommendation.

Method	Scenario	Data Augmentation	Contrast Type	Contrastive Objective	Training Scheme
SGL[30]	Graph	Edge/Node Dropout	Node-Node	InfoNCE	JL
DCL [89]	Graph	Edge Dropout	Node-Node	InfoNCE	JL
CCDR [76]	Graph (Cross-domain)	Subgraph Sampling	Node-Node	InfoNCE	JL
PCRec [77]	Graph (Cross-domain)	Subgraph Sampling	Node-Node	InfoNCE	PF
HHGR [69]	Graph (Group)	Node Dropout	User-User	Cross-Entropy	JL
CL4SRec [61]	Sequential	Item Masking/Reordering/Cropping	Sequence-Sequence	InfoNCE	JL
H^2SeqRec [62]	Sequential	Item Masking/Cropping	Sequence-Sequence	InfoNCE	PF
CoSeRec [67]	Sequential	Item Substitution/Insertion	Sequence-Sequence	InfoNCE	JL
ContraRec [94]	Sequential	Item Masking/Reordering/overlapping	Sequence-Sequence	InfoNCE	JL
DHCN [28]	Session	Feature Shuffling	Session-Session	Cross-Entropy	JL
EGLN [73]	Graph	Graph Diffusion	Pair-Graph	Cross-Entropy	JL
BiGI [75]	Graph	Subgraph Sampling	Pair-Graph	Cross-Entropy	JL
HGCL [95]	Graph	Feature Shuffling	Node-Graph	Cross-Entropy	JL
MHCN [29]	Graph (Social)	Subgraph Sampling/Feature Shuffling	User-Hypergraph	Triplet-loss	JL
SMIN [98]	Graph (Social)	Graph Diffusion	Node-Context	Cross-Entropy	JL
NCL [83]	Graph	Feature Clustering	Node-Cluster	InfoNCE	JL
ICL [82]	Sequential	Feature Clustering	Sequence-Cluster	InfoNCE	JL
S^3-Rec [25]	Sequential	Item Masking/Cropping	Item-Context	InfoNCE	PF
CFM [78]	Feature-Based	Feature Dropout	Item Feature-Item Feature	InfoNCE	JL
MISS [99]	CTR Prediction	Feature Extractor	User Feature-User Feature	InfoNCE	JL
DuoRec [79]	Sequential	Neuron Masking	Sequence-Sequence	InfoNCE	JL
SimGCL [101]	Graph	Feature Noises	Node-Node	InfoNCE	JL
SRMA [102]	Sequential	Neuron Masking/Layer Dropping /Encoder Complementing	Sequence-Sequence	InfoNCE	JL

a detailed derivative of how JSD estimates MI). It is widely used in the graph scenario, and can be formulated as:

$$\begin{aligned}\mathcal{MI}_{JSD}\left(\mathbf{h}_i,\mathbf{h}_j\right) = &-\mathbb{E}_{\mathcal{P}}\left[\log\left(f_{\mathrm{D}}\left(\mathbf{h}_i,\mathbf{h}_j\right)\right)\right] \\ &-\mathbb{E}_{\mathcal{P}\times\tilde{\mathcal{P}}}\left[\log\left(1-f_{\mathrm{D}}\left(\mathbf{h}_i,\tilde{\mathbf{h}}_j\right)\right)\right]\end{aligned} \tag{28}$$

where $\mathcal{P}$ denotes the joint distribution of $\mathbf{h}_i$ and $\mathbf{h}_j$, and $\mathcal{P}\times\tilde{\mathcal{P}}$ denotes the product of marginal distributions. The discriminator $f_{\mathrm{D}}:\mathbb{R}^d\times\mathbb{R}^d\to\mathbb{R}$ can be implemented in various forms. The vanilla implementation in [15] is $f_{\mathrm{D}}=\mathbf{h}_i^\top\mathbf{W}\mathbf{h}_j$ which is called bilinear scoring function and is directly applied in [73], [75]. In [28], [98], its dot-product form is used and shows comparable performance.

4.4.2 Noise-Contrastive Estimator

The InfoNCE [13] formulation follows a softmax-based version of NCE [108] to identify the positive sample amongst a set of negative samples. It also has been proved that minimizing the InfoNCE loss is equivalent to maximizing a lower bound of MI, and [107] find that using InfoNCE often outperforms JSD in CV tasks. As the most popular contrastive loss in SSR, it is formulated as:

$$\mathcal{MI}_{NCE}=-\mathbb{E}\left[\log\frac{e^{f_{\mathrm{D}}(\mathbf{h}_i,\mathbf{h}_j)}}{\sum_{n\in\mathcal{N}_i^-\cup\{j\}}e^{f_{\mathrm{D}}(\mathbf{h}_i,\mathbf{h}_n)}}\right] \tag{29}$$

where $\mathcal{N}_i^-$ is the negative sample set of i, and is often sampled within a batch. When $f_{\mathrm{D}}(\mathbf{h}_i,\mathbf{h}_j)=\frac{\mathbf{h}_i^\top\mathbf{h}_j}{\|\mathbf{h}_i\|\|\mathbf{h}_j\|}/\tau$, it is also called NT-Xent loss [10] where τ is the temperature (e.g., 0.2), which is its well-known version.

Despite its effectiveness, how the InfoNCE behaves in SSR has not been paid enough attention. Wang and Isola [109] identify two key properties related to the InfoNCE (NT-Xent loss): *alignment* (closeness) of representations from positive pairs and *uniformity* of the normalized representations on the hypersphere. By rewriting the NT-Xent version of Eq. (31), we can get:

$$=\underbrace{\mathbb{E}\left[-\mathbf{z}_i^\top\mathbf{z}_j/\tau\right]}_{\text{alignment}}+\underbrace{\mathbb{E}\left[\log\left(e^{\mathbf{z}_i^\top\mathbf{z}_j/\tau}+\sum_{n\in\mathcal{N}_i^-}e^{\mathbf{z}_i^\top\mathbf{z}_n/\tau}\right)\right]}_{\text{uniformity}} \tag{30}$$

where the first term corresponds to *alignment* and the second corresponds to *uniformity* of sample vectors on a hypersphere given the normalization $\mathbf{z}_i=\frac{\mathbf{h}_i}{\|\mathbf{h}_i\|}$. In SSR, a few works also report their existence. Qiu *et al.* [79] and Yu *et al.* [101] demonstrate that optimizing the InfoNCE loss learns a more uniform distribution of item/node representations, which can mitigate the representation degeneration problem in sequential recommendation and address the popularity bias in graph-based recommendation [110]. In addition, Wang and Liu [111] show that the InfoNCE loss is hardness-aware, and the temperature τ controls the strength of penalties on hard negative samples. Such a property is also mentioned in [30] when it is applied to recommendation.

Meanwhile, there is also a potential defect of the InfoNCE loss that should be noticed. For each instance in the input, the InfoNCE pushes it away from other instances except its augmentation counterpart in the representation space. However, similar users/items widely exist in recommender systems. As a result, many semantically similar instances are mixed with the unrelated instances and are fed to the loss as false negative samples, which will impair recommendation performance. To tackle this problem, inspired by [112], a few works [79], [56], [55] propose to incorporate multiple positive samples into the InfoNCE loss, inducing a

modified version defined as:

$$\mathcal{MI}^{+}_{NCE} = -\mathbb{E}\left[\log \frac{\sum_{j\in\mathcal{N}_i^+} e^{-\mathbf{z}_i^\top \mathbf{z}_j/\tau}}{\sum_{n\in\mathcal{N}_i^-\cup\mathcal{N}_i^+} e^{-\mathbf{z}_i^\top \mathbf{z}_n/\tau}}\right] \tag{31}$$

Qiu *et al.* [79] proposes to identify semantically positive sequences by checking if two sequences have the same item to be predicted. Yu *et al.* [56] and Xia *et al.* [55] build multiple encoders on semantically similar graphs to predict positive examples for a given instance.

4.5 Pros and Cons

Due to the flexibility to augment data and set pretext tasks, contrastive methods expand rapidly in recent years and reach most recommendation topics. Although there have been no reports showing that the contrastive SSR has overwhelming advantages over other SSR paradigms, it has shown remarkable effectiveness on improving recommendation with lightweight architectures. However, it is often compromised by the **unknown criterion for high-quality data augmentations** [59]. Existing contrastive methods are mostly based on arbitrary data augmentations and are selected by trial-and-error. There have been neither rigorous understanding of how and why they work nor rules or guidelines clearly telling what good augmentations are for recommendation. Besides, some common augmentations, which were considered to be useful, recently even have been proved to has a negative impact on the recommendation performance [101]. As such, without knowing what augmentations are informative, the contrastive task may fall short of expectation.

5 Generative Methods

Generative SSR methods are based on the idea that by reconstructing the original input with its corrupted version, the intrinsic correlations in the data can be encoded to benefit the recommendation task. In this section, we mainly focus on the MLM-based generative SSR methods which are one of the current trends. According to the reconstruction objectives, we divide generative SSR methods into two categories: **Structure Generation** and **Feature Generation**.

5.1 Structure Generation

This branch of methods leverages the structural information to supervise the model. By applying the masking/dropout-based augmentation operators (see Section 3) to the original structure, its corrupted versions are obtained. In the scenario of sequence-based recommendation, recovering the structure can be formulated as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}\left(f_\theta(\tilde{\mathcal{S}})\right), S\right) \tag{32}$$

Where $\tilde{\mathcal{S}}$ denotes the corrupted sequence in which a portion of items are masked (replaced with a special token [mask]).

BERT4Rec [23] is the first instantiation of BERT [12] in sequential recommendation (**BERT4SessRec** [24] is a concurrent work with almost the same idea). It upgrades the left-to-right training scheme in SASRec [113], and proposes to learn a bidirectional representation model. Technically, it randomly masks some items in the input sequences, and then predicts the ids of those masked items based on their surrounding items. To align the prediction phase with the training phase, it appends the token [mask] at the end of the input sequence to indicate the item to be predicted. The objective is formulated as:

$$\mathcal{L} = \frac{1}{|S_u^m|} \sum_{i_m \in S_u^m} -\log P\left(i_m = i_m^* \mid \tilde{S}_u\right), \tag{33}$$

where $\tilde{S}_u$ is the corrupted version of S_u, S_u^m is the randomly masked items, and i_m^* is the true masked item. This loss is widely used in the BERT-like generative SSR models, serving as the optimization objective of their generative pretext tasks.

Inspired by the success of BERT4Rec, follow-up works apply the masked-item-prediction training to more specific scenarios. **UNBERT** [114] and Wu *et al.* [115] explore the use of this technique for news recommendation in almost the same way. The input of UNBERT is a combination of news sentences and user sentences with a set of special symbols. It randomly masks some word-piece tokens to pre-train the token representations with the Cloze task, and then fine-tunes the model on the news recommendation task. Another similar work **U-BERT** [116] uses the review comments to pre-train the encoder with masked-word-token-prediction in the source domain, and then fine-tunes the encoder with an added layer in the target domain where comments are insufficient for rating prediction. **GRec** [117] develops a gap-filling mechanism with the encoder-decoder setting. The encoder takes a partially-complete session sequence as input, and the decoder predicts the masked items conditioned on the both the output of the encoder and its own complete embeddings. **UPRec** [118] further modifies BERT4Rec to enable it to exploit the heterogeneous information such as user attributes and social networks to enhance the sequence modeling.

Although the above generative models have achieved promising results, they are mainly pre-trained for one type of recommendation task. There is also a line of research whose goal is to learn general-purpose representation through generative pre-training [125], [119], [120] so as to benefit multiple downstream recommendation tasks. **PeterRec** [119] makes the first attempt to transfer the model parameters pre-trained by the masked-item-prediction to user-related tasks. Instead of fine-tuning the pre-trained parameters, it injects a series of small grafting neural networks into the pre-trained original model, and only trains these patches to adapt to specific tasks. Similarly, **ShopperBERT** [120] is pre-trained with nine generative pretext tasks including the masked-purchase-prediction, and the learned universal user representations can serve six downstream recommendation-related tasks. Benefiting from the large-scale pre-training datasets, it shows superiority over the task-specific Transformer-based models which are learned from scratch.

In the graph-based recommendation scenario, the structure generation approach is formulated as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ssl}\left(g_{\phi_s}\left(f_\theta(\tilde{\mathcal{G}})\right), \mathbf{A}\right) \tag{34}$$

TABLE 2: A summary of the surveyed papers on Generative self-supervised recommendation.

Method	Scenario	Data Augmentation	Branch	Training Scheme
BERT4Rec [23]	Sequential	Item Masking	Structure Generation	IL
BERT4SessRec [24]	Sequential (Session)	Item Masking	Structure Generation	PF
UNBERT [114]	Sequential (News)	Word Masking	Structure Generation	PF
U-BERT [116]	Sequential	Word Masking	Structure Generation	PF
GRec [117]	Sequential (Session)	Item Masking	Structure Generation	IL
UPRec [118]	Sequential	Item Masking	Structure Generation	PF
PeterRec [119]	Sequential	Item Masking	Structure Generation	PF
ShopperBERT [120]	Sequential	Item Masking	Structure Generation	PF
G-BERT [121]	Graph (Medication)	Node Masking	Structure Generation	PF
PMGT [122]	Graph	Subgraph Sampling/Node Masking	Structure & Feature Generation	PF
Ma *et al.* [123]	sequential	Sequence Splitting	Feature Generation	JL
MMInfoRec [33]	sequential	Sequence Splitting/Feature Dropout	Feature Generation	IL
PT-GNN [124]	Graph	Subgraph Sampling	Feature Generation	PF

G-BERT [121] combines the power of GNNs and BERT for medication recommendation. It models the diagnosis and medication codes in electrical health records as two tree-like graphs and employs GNNS to learn the graph representations. Then the representations are fed to the BERT encoder and are pre-trained with two generative pretext tasks: self-prediction and dual-prediction. The self-prediction task reconstructs the masked codes with the same type of graph, and the dual prediction task reconstructs the masked codes with the other type of graph. **PMGT** [122] conducts a graph reconstruction task with the sampled subgraph. It develops a sampling method to sample subgraph for each item node, and reorganizes the sampled subgraph as an ordered sequence according to the neighbor importance. The subgraph is then fed to a Transformer-based encoder, and the method pre-trains the item representations with the missing neighboring item prediction.

5.2 Feature Generation

The feature generation task can be formulated as:

$$f_\theta^* = \arg\min_{f_\theta, g_{\phi_s}} \left\| g_{\phi_s}\left(f_\theta(\tilde{\mathcal{D}})\right) - \mathbf{X} \right\|^2, \tag{35}$$

where $\|\cdot\|^2$ is the MSE loss, and $\mathbf{X}$ is a general expression of features that can be the user profile attributes, item textual features, or learned user/item representations.

In **PMGT** [122], besides the graph reconstruction task, the feature reconstruction task is also used to pre-tain the Transformer-based recommendation model. The method extracts item's textual and image features in advance and initializes item embeddings with the extracted features. It then masks a portion of sampled nodes and uses the rest nodes to recover the features of the masked nodes. As for the sequence feature generation, Ma *et al.* [123] propose to reconstruct the representation of the future sequence with the past behaviors. Specifically, they disentangle the intentions behind any given sequence of behaviors and the reconstruction is conducted between any pairs of sub-sequences that involve a shared intention. Similarly, in **MMInfoRec** [33] given a sequence with t items, it encodes the sequence and predicts the next item's representation at time step t+1. The augmented sequence representation is then contrasted with the real t+1 item's representation (ground-truth). An auto-regressive prediction module is designed to include more futuristic information by predicting the t+k item with item t+1 to item t+i-1. A dropout function is used in the encoder to create multiple semantically similar item representations for improving CL.

To enhance cold-start users and items representation, **PT-GNN** [124] proposes to pre-train GNN models by mimicking the meta-learning setting. It picks the users/items with sufficient interactions as target users/items, and performs graph convolution on sampled K neighbors for the targets to predict their ground-truth embeddings learned from the whole graph. Optimizing this reconstruction loss directly improves the embedding capacity, making the model easily and rapidly adapt to cold-start users/items.

5.3 Pros and Cons

The newest generative SSR methods mostly follow the pipeline of masked language models. Relying on this standard pipeline and the capacity of Transformer, these methods have shown significant results. The successful training experience of BERT on large-scale corpus also paves the way for the application of large MLM-based recommendation models. However, this branch of methods may be confronted with the challenge of **heavy computation**. Due to the small scale of most open-source datasets, the Transformer-based generative methods usually adopt a one- or two-layer setting. However, when training with large-scale datasets for news recommendation or general-purpose representations, the computation is incredibly heavy. Particularly, considering that scaling-up pre-training dataset really helps, there will be a dilemma between performance and high computational overhead, especially for the research groups which are with limited computing resources.

6 Predictive Methods

Unlike the generative SSR methods which corrupt the original data to acquire self-supervision, the predictive SSR methods deal with the self-generated supervisory signals obtained from the complete original data. According to what the predictive pretext task predicts, we divide predictive methods into two branches: **Sample Prediction** and **Pseudo-Labels Prediction**.

6.1 Sample Prediction

Under this branch, self-training [126] (a flavor of semi-supervised learning) is linked to SSL. The SSR model is first pre-trained with the original data. Based on the pre-trained parameters, the model predicts the potential samples which are informative to the recommendation task. The acquired samples act as the augmented data and are then fed to the model to enhance the recommendation task or to recursively generate better samples. The difference between SSL-based sample prediction and the pure self-training is that in the setting of semi-supervised learning, a finite number of unlabeled samples are ready, while in the setting of SSL, the samples are dynamically generated.

Sequential recommendation models often perform poorly on short sequences due to limited user behaviors. To improve the model performance, **ASReP** [127] proposes to augment the short sequences with pseudo-prior items. Given sequences in a left-to-right chronological order, ASRep first pre-trains a Transformer-based encoder SASRec [113] in a reverse manner (i.e., from right-to-left) so that the encoder is capable of predicting the pseudo-prior items. It then obtains an augmented sequence (e.g., $\tilde{S} = [\tilde{i}_{-1}, \tilde{i}_0, i_1, i_2, ..., i_k]$) where the fabricated subsequence is appended to the beginning of the original sequence. After that, the encoder is fine-tuned on the augmented sequences in a left-to-right way to predict the next item i_{k+1} in the original sequence. A follow-up work **BiCAT** [128] argues that the reverse augmentation may be inconsistent with the original correlation. It further proposes to simultaneously pre-train the encoder from both left-to-right and right-to-left directions. The bidirectional training can bridge the gap between the reverse augmentation and the forward recommendation. Such reverse augmentations in this method are recursively generated and fed to the encoder in a self-training fashion.

In the graph scenario, the samples can also be predicted based on node feature/semantic similarities. When there are multiple encoders built on different graphs, they can recursively predict samples for other encoders where the self-training is upgraded to co-training [129]. In **SEPT** [56] and **COTREC** [55], such ideas are presented. We introduce them in Section 7 because these methods assemble multiple pretext tasks.

6.2 Pseudo-Label Prediction

Under this branch, the pseudo-labels are presented in two forms: pre-defined discrete values and pre-computed/learned continuous values. The former usually describe a type of relation between two objects. The goal of the corresponding pretext task is to predict whether such a relation exists between the given pair of objects. The latter often describes the attribute values of the given object (e.g., node degree [130]), a probability distribution, or a feature vector. The corresponding pretext task aims to minimize the difference between the output and the pre-computed continuous values. We can formulate the two prediction tasks as: **Relation Prediction** and **Similarity Prediction**.

6.2.1 Relation Prediction

The relation prediction task can be formulated as a classification problem where the pre-defined relations, serving as the pseudo-labels, are automatically self-generated without any cost. We can refine Eq. (4) to provide this branch of methods with the formulation as:

$$f_{\theta^*} = \arg\min_{f_\theta, g_{\phi_s}} \mathcal{L}_{ce}\left(g_{\phi_s}\left(f_\theta(o_i, o_j)\right), \mathcal{T}(o_i, o_j)\right), \tag{36}$$

where o_i and o_j are a pair of objects from $\mathcal{D}$, $\mathcal{T}$ is the class label generator, and $\mathcal{L}_{ce}$ is the cross-entropy loss.

Inspired by the next sentence prediction (NSP) in BERT [12] (i.e., predicting if sentence B comes after sentence A), a few predictive self-supervised sequential recommendation models propose to predict the relation between two sequences. **PTUM** [131] duplicates NSP in BERT to split a user behavior sequence into two non-overlapped subsequences, respectively representing the past behaviors and future behaviors. It then predicts if a candidate behavior is the future behavior based on the past behaviors. **SSI** [68] pre-trains a Transformed-based recommendation model with a pretext task that shuffles/replaces a portion of items in a given sequence, and then predicts if the modified sequence is in the original order/from the same user.

In the graph scenario, the pseudo-relation is often built by random-walks. **CHEST** [70] proposes to conduct pre-defined meta-path-based random walks on heterogeneous user-item graphs to connect user-item pairs. It considers meta-path type prediction as a predictive task to pre-train a Transformer-based recommendation model. Given a user-item pair, the pretext task predicts if there exists a path instance of a specific meta-path between them.

6.2.2 Similarity Prediction

The similarity prediction task can be formulated as a regression problem, where the pre-computed/learned continuous values serve as the targets that the model's output needs to approximate. We can refine Eq. (4) to provide this branch of methods with the formulation as:

$$f_{\theta^*} = \arg\min_{f_\theta, g_{\phi_s}} \left\| g_{\phi_s}\left(f_\theta(\mathcal{D})\right) - \mathcal{T}(\mathcal{D}) \right\|^2, \tag{37}$$

where $\mathcal{T}$ is the label generator that yields the targets. It can be an encoder to learn the targets or a series of actions to pre-compute the targets.

BUIR [31] is a representative predictive SSR method which imitates the vision model BYOL [9] and relies on two asymmetric graph encoders, referred to as online and target networks, to supervise each other without negative sampling. Given a user-item pair, the online network is fed with the user representation and is trained to predict the item representation output by the target network and vice versa. In this way, BUIR fulfils self-supervision by boostrapping representations, which means using estimated values to estimate its target values. Particularly, the online network is updated in an end-to-end fashion while the target encoder is updated by momentum-based moving average to slowly approximate the online encoder, which encourages to provide enhanced representations as the target for the online encoder. **SelfCF** [71] inherits the merits of BUIR and further simplifies it by only using one shared encoder in the two networks. To obtain more supervision signals to learn discriminative representations, it perturbs the output of the target network. Another very similar concurrent work

TABLE 3: A summary of the surveyed papers on predictive self-supervised recommendation.

Method	Scenario	Data Augmentation	Branch	Training Scheme
ASReP [127]	Sequential	Prior Items	Sample Prediction	PF
BiCAT [128]	Sequential	Prior Items	Sample Prediction	PF
PTUM [131]	Sequential	Matching Label	Relation Prediction	PF
SSI [68]	Sequential	Matching Label	Relation Prediction	PF
CHEST [70]	Graph	Path Label	Relation Prediction	PF
BUIR [31]	Graph	Bootstrapped representation	Similarity Prediction	IL
SELFCF [71]	Graph	Bootstrapped representation	Similarity Prediction	IL
CLUE [60]	Sequential	Bootstrapped representation	Similarity Prediction	IL
RDC [132]	Graph	Learned Rating Distribution	Similarity Prediction	JL
MrTransformer [133]	Sequential	Recombined Preference Feature	Similarity Prediction	JL
DUAL [134]	Graph (Social)	Pre-computed Probability	Similarity Prediction	JL

CLUE [60], which is an instantiation of BYOL [9] in sequential recommendation, also employs one shared encoder. The main pipeline difference between SELFCF and CLUE is that CLUE augments the input and SELF augments the output representations.

The similarity prediction is also used to alleviate the selection bias [39] in recommender systems. In **RDC** [132], two types of users are defined, which are pivot users who select and rate items at random and non-pivot users who are more likely to rate items that they strongly like or dislike and hence get biased recommendations. RDC forces the rating distribution features of the non-pivot users to be close to that of their similar pivot users so as to correct biases, where the rating distribution features are dynamically computed and serve as the self-supervision signals. Such tasks are also used for the user preference disentanglement [135]. In **MrTransformer** [133], given a pair of sequences representing two users, each user preference representation is separated into two parts: common preference representation and unique preference representation. The method then swaps the common preference representations to recombine the user preference representations and requires that the recombined preference should be as similar as possible to the original preference representations so that the model can learn how to identify and edit user preferences in order to learn more discriminative user representations.

When heterogeneous information graphs are involved, the similarity prediction task can also be used to capture the rich semantics. In **DUAL** [134], the meta-path-based random walks which connect user-item pairs are conducted. For each user-item pair, the number of meta-path instances are recorded and scaled to measure the interacted probability of the pair. Then a path regression pretext task is assigned to predict the pre-computed probability and is expected to integrate the path semantics into node representations to enhance recommendation.

6.3 Pros and Cons

Compared with the contrastive methods and generative methods which mostly rely on static augmentation operators, the predictive methods acquire samples and pseudo-labels in more dynamic and flexible ways. Particularly, the samples are predicted based on evolving model parameters, which directly refine the self-supervision signals and align them with the optimization objective, probably leading to better recommendation performance. However, we also should **be cautious about using the pre-augmented labels**. Most existing methods collect the pseudo-labels based on heuristics, without assessing how relevant these labels and predictive tasks are to recommendation. Considering that the user-item interactions and the associated attributes/relations generate with rationales (e.g., social dynamics), it is necessary to incorporate expert knowledge as a prior into the collection of pseudo-labels, which increases the expense of developing predictive SSR approaches.

7 Hybrid Methods

Hybrid methods assemble multiple pretext tasks to take advantage of various types of supervision signals. We divide surveyed hybrid methods into two groups according to how their pretext tasks function, including **Collaborative** and **Independent**. A summary of surveyed contrastive methods is presented in Table 4.

7.1 Collaborative Methods

Under this branch, different pretext tasks collaborate in a way that other pretext tasks often serve the contrastive pretext task by creating more informative samples for contrast.

CCL [63] proposes a curriculum learning [136] strategy to pre-train a Transformer-based model by linking a generative pretext task to a contrastive pretext task for sequential recommendation. Given a sequence, the generative task is for the masked-item-prediction. With the predicted probabilities, the method samples items to fill the masked part for sequence augmentations. Then these augmented sequences are fed to the contrastive task to be contrasted with the original sequence. According to the augmented sequences' ability to restore users' attribute information, they are divided into N bins for an easy-to-difficult contrastive course arrangement.

The predictive task can also serve the downstream contrastive task. **SEPT** [56] is the first SSR model that unifies SSL and tri-training [137] (a special type of semi-supervised learning) through a sample-based predictive task for social recommendation. It builds three graph encoders (one for recommendation) over three views with different social semantics and any two encoders can predict semantically

TABLE 4: A summary of the surveyed self-supervised recommendation methods with hybrid pretext tasks.

Method	Scenario	Data Augmentation	Hybrid Type	Branch	Training Scheme
CCL [63]	Sequential	Item Masking	Generative + Contrastive	Collaborative	PF
SEPT [56]	Graph (Social)	Edge Dropout Predicted samples	Predictive + Contrastive	Collaborative	JL
COTREC [55]	Sequential	Predicted samples	Predictive + Contrastive	Collaborative	JL
CHEST [70]	Graph	Node/Edge Masking Path Removal/Insertion	Generative + Contrastive	Independent	PF
MPT [72]	Graph	Node Masking/Substitution/Deletion	Generative + Contrastive	Independent	PF
SSL [68]	Sequential	Item Substitution/Reordering	Predictive + Contrastive	Independent	PF
PTUM [131]	Sequential	Item Masking	Generative + Predictive	Independent	PF
UPRec [118]	Sequential	Item Masking	Generative + Predictive	Independent	PF

similar samples for rest encoder. These samples are incorporated as supplementary positive self-supervision signals in the linked contrastive task to improve encoders. The improved encoders in turn recursively predict more informative samples. **COTREC** [55] follows the framework of SEPT and reduces the number of used encoders to two for session-based recommendation. The two encoders are built over two session-induced temporal graphs and iteratively predict samples to improve each other through the contrastive task. To prevent the mode collapse problem in two-encoder-based co-training [129], it imposes a divergence constraint on the two encoders to keep them slightly different by exploiting adversarial examples [138].

7.2 Independent Methods

Under this branch, there are no correlations between different pretext tasks and they work independently.

Similar to CCL [63], **CHEST** [70] also connects curriculum learning with SSL to pre-train a Transformer-based recommendation model on heterogeneous information networks. However, its pretext tasks are unlinked. Given the user-item graph and associated attributes, CHEST conducts meta-path-based random walks starting from a user node and ending with an item node to form interaction-specific subgraphs, each of which consists of multiple meta-path instances. The generative task in CHEST predicts the masked node/edge in a subgraph with the rest, which exploits the local context information and is considered the elementary course. The contrastive task learns subgraph-level semantics for user-item interaction by pulling the original subgraph and augmented subgraphs closer, which exploits the global correlations and is considered the advanced course. **MPT** [72] extends PT-GNN [124] which only considers the intra-correlations within user and item subgraphs in a generative pretext task to enhance representation capacity for cold-start recommendation. To capture the inter-correlations across different subgraphs, it adds contrastive pretext tasks performed on graphs and random walk-based sequences where node deletion/substitution/masking are used to augment data for training the GNN and Transformer-based encoders. The contrastive tasks and the generative tasks are conducted in parallel with their own parameters. Finally, the parameters are merged and fine-tuned for recommendation.

Besides the combination of generative and contrastive tasks, **SSI** [68] assembles a label-based predictive task and a contrastive task to pre-train models for sequential recommendation. Given a sequence, two consistency constraint: temporal consistency and persona consistency are imposed through the predictive learning. For temporal consistency, this method needs to predict whether the input sequence is in the original order or shuffled. For persona consistency, it needs to differentiate whether the input sequence is from one certain user or with some items replaced by unrelated items. Meanwhile, an item masking-based contrast is conducted between the masked item representation and the sequence representation. As for the combination of generative and predictive tasks, it is witnessed in **PTUM** [131] which imitates BERT [12] to conduct the masked-item-generation task and the next-item-prediction task. Moreover, **UPRec** [118] connects BERT with heterogeneous user information, and pre-trains the encoder with the social relation and user attribute prediction tasks in parallel.

7.3 Pros and Cons

Hybrid methods assemble multiple pretext tasks and therefore can get enhanced and comprehensive self-supervision. Particularly, for methods under the collaborative branch in which the generative/predictive task serves the contrastive task by dynamically yielding samples for contrast [56], [55], [136], they have distinct advantage over their static counterparts in training effectiveness. However, hybrid methods are confronted with the problem of **coordinating multiple pretext tasks**. For methods under the independent branch [72], [68], it is challenging for them to make a tradeoff between different pretext tasks. A cumbersome manual search by trial-and-errors or expensive domain knowledge for hyperparameters is required. Besides, different pretext tasks may also interfere with each other, which may require heavier architectures with more parameters such as multi-gate mixture-of-experts networks [139] to separate task-shared and task-specific information. As a consequence, a strong hybrid SSR model comes at a higher training cost compared to the models with only a single pretext task.

8 SELFREC: A LIBRARY FOR SELF-SUPERVISED RECOMMENDATION

SSR is now enjoying a period of prosperity, and more and more SSR models are developed and claim themselves as the state-of-the-art. Despite the increasing numbers, we notice that the empirical comparisons between different SSR models in the collected papers are often unfair. Besides,

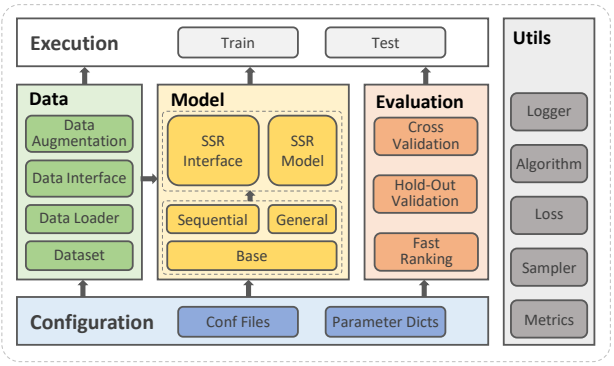


Fig. 6: The architecture of the SSR library - SELFRec.

some code implementations of these methods are inefficient and the readability is poor. Although there have been some open-source repositories such as RecBole [140] and QRec [56] which have implemented a few SSR models with a unified framework and provide standard evaluation protocols, they are designed to cover all kinds of recommendation models and their architectures are not efficient for implementing SSR models. To facilitate the development and standardize the evaluation of SSR models, we release an open-source library - **SELFRec**, which inherits the merits of RecBole and QRec, whose architecture is shown in Fig. 6.

In SELFRec, we incorporate multiple high-quality datasets which are widely used in the surveyed papers such as Amazon-Book [30], Yelp-2018 [87], and MovieLens-1M for both sequential and general scenarios. More than 10 metrics including ranking-based such as MRR@K and NDCG@K and rating-based such as MSE and RMSE are integrated. There are more than 10+ state-of-the-art SSR methods implemented in SELFRec for empirical comparison such as SGL [30], CL4SRec [61], DHCN [28], etc. Some important features of SELFRec are summarized as follows:

- **Fast Execution**: SELFRec is developed with Python 3.7+, Tensorflow 1.14+ and Pytorch 1.7+. All models run on GPUs. Particularly, we optimize the time-consuming procedure of item ranking, drastically reducing the ranking time to seconds.
- **Easy Expansion**: SELFRec provides a set of simple and high-level interfaces, by which new SSR models can be easily added in a plug-and-play fashion.
- **Highly Modularized**: SELFRec is divided into multiple discrete and independent modules/layers. This design decouples the model design from other procedures. For users of SELFRec, they just need to focus on the logic of their method, which streamlines the development.
- **SSR-Specific**: SELFRec is designed for SSR. For the data augmentation and self-supervised tasks, it provides specific modules and interfaces for rapid development.

Due to the limited space, we refer you to the homepage of SELFRec https://github.com/Coder-Yu/SELFRec for the usage and detailed documents.

9 Discussion

In this section, we analyze the limitations of existing SSR and outline some promising research directions which are worth exploring.

9.1 Theory for Augmentation Selection

Data augmentation is a key ingredient of SSR, which is critical to the performance. The overwhelming majority of existing SSR methods augment the original data by imitating the approaches from the fields of CV, NLP, and graph learning. However, these approaches cannot be seamlessly transplanted to recommendation to deal with the user behavior data which is tightly coupled with the scenario and blended with noises and randomness. Besides, most methods augment data based on heuristics, and search the appropriate augmentations by the cumbersome trial-and-error. Although there have been some theories that try to demystify the visual view choices in contrastive learning [141], [59], the principle for augmentation selection in recommendation is seldomly studied. A solid recommendation-specific theoretical foundation which can streamline the selection process and free people from the tedious trial-and-error work is therefore urgently needed.

9.2 Explainable Self-Supervised Recommendation

Despite the desired performance of existing SSR models, what lead to these performance gains are not theoretically justified in most of them. They are considered as black-box models and the goal of them is just to reach a higher performance. Some components like the augmentations and the objectives of pretext tasks, which are presumed useful, lack reliable interpretability to demonstrate their effectiveness. In a very recent research [101], it has been experimentally proved that some graph augmentations, which used to be considered informative, may even impair the performance. Meanwhile, without solid explanations, it is unclear whether the models exchange other properties such as robustness for the performance improvement. For more reliable SSR models, it is necessary to figure out what they have learned or how the model have changed through self-supervised training.

9.3 Attacking and Defending in Pre-trained Recommendation Models

Due to the open nature, recommender systems are vulnerable to the data poisoning attack which injects deliberately crafted user-item interaction data into the model training set to tamper with the model parameters and manipulate recommendation results as desired [142], [143]. The attack and the corresponding defence approaches for supervised recommender systems have been well-studied. However, it remains unknown that if the recommendation models pre-trained in a self-supervised way are robust to such attacks. We also notice that a few pioneering works have made attempts to attack the pre-trained encoders in vision and graph classification tasks [144], [145]. We believe that developing new-type attacks and defending self-supervised pre-trained recommender systems against these attacks will be an interesting future research direction.

9.4 On-Device Self-Supervised Recommendation

Modern recommender systems operate in a fully server-based fashion to cater to millions of users, which come at a cost of a huge carbon footprint and raise privacy

concerns. Decentralized recommender systems [146], [147] which are deployed on resource-constrained devices such as smartphones have become increasingly popular. However, on-device recommender systems are compromised by the highly shrinked model size and limited labeled data. SSL can be a potential solution to these problems. Particularly, when combined with the technique of knowledge distillation [148], [149], SSL may largely compensate for the accuracy degradation of on-device recommendation models. Currently, on-device self-supervised recommendation has not been explored, and we believe it deserves further study.

9.5 Towards General-Purpose Pre-training

In industry, the data in recommender systems is multi-modal and the scenario are rather diverse. A variety of deep recommendation models are trained across different modalities (e.g., video, image, end text) for different recommendation tasks [150]. The training and tasks are usually independent and consume a large amount of computing resource. Since the data of different modalities is usually related, it is natural to explore general-purpose recommendation models which are pre-trained with the multi-modal SSL on large-scale data and can adapt to multiple downstream recommendation tasks with the cheap fine-tuning. Particularly, for the new tasks or scenarios where the training data is very sparse, transferring supervision from other modalities is really helpful. Though there have been several works towards developing general-purpose recommendation models [120], [125], they are all trained in a BERT-like fashion with similar architectures. It is meaningful to seek more efficient training strategies and more effective model architectures.

10 Conclusion

In this survey, we provide a timely and systematical review of the research efforts on SSR. Specifically, we first provide a definition of SSR by which we investigated a large number of related papers and classify existing research on SSR into four categories: contrastive, generative, predictive, and hybrid. For each category, its concept and formulation, the involved methods, and the pros and cons are introduced in turn. Moreover, to facilitate the development and evaluation of SSR, we release an open-source library in which the common benchmark datasets and metrics are incorporated, and a number of state-of-the-art SSR methods are implemented for empirical comparison. Finally, we identify the limitations in current research efforts, and outline some promising future research to advance SSR.

References

[1] F. Ricci, L. Rokach, and B. Shapira, "Introduction to recommender systems handbook," in *Recommender systems handbook*. Springer, 2011, pp. 1–35.

[2] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *RecSys*, 2016, pp. 191–198.

[3] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[4] G. Zhou, X. Zhu, C. Song, Y. Fan, H. Zhu, X. Ma, Y. Yan, J. Jin, H. Li, and K. Gai, "Deep interest network for click-through rate prediction," in *KDD*, 2018, pp. 1059–1068.

[5] B. M. Sarwar, *Sparsity, scalability, and distribution in recommender systems*. University of Minnesota, 2001.

[6] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[7] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE TKDE*, 2021.

[8] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *CVPR*, 2020, pp. 9729–9738.

[9] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar *et al.*, "Bootstrap your own latent: A new approach to self-supervised learning," *arXiv preprint arXiv:2006.07733*, 2020.

[10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICML*. PMLR, 2020, pp. 1597–1607.

[11] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, and R. Soricut, "Albert: A lite bert for self-supervised learning of language representations," *arXiv preprint arXiv:1909.11942*, 2019.

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[13] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[14] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "Gcc: Graph contrastive coding for graph neural network pre-training," in *KDD*, 2020, pp. 1150–1160.

[15] P. Velickovic, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax." in *ICLR (Poster)*, 2019.

[16] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-n recommender systems," in *WSDM*, 2016, pp. 153–162.

[17] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *CIKM*, 2015, pp. 811–820.

[18] M. Gao, L. Chen, X. He, and A. Zhou, "Bine: Bipartite network embedding," in *SIGIR*, 2018, pp. 715–724.

[19] C. Zhang, L. Yu, Y. Wang, C. Shah, and X. Zhang, "Collaborative user network embedding for social recommender systems," in *ICDM*. SIAM, 2017, pp. 381–389.

[20] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014, pp. 2672–2680.

[21] Q. Wang, H. Yin, H. Wang, Q. V. H. Nguyen, Z. Huang, and L. Cui, "Enhancing collaborative filtering with generative augmentation," in *KDD*, 2019, pp. 548–556.

[22] J. Yu, H. Yin, J. Li, M. Gao, Z. Huang, and L. Cui, "Enhance social recommendation with adversarial graph convolutional networks," *IEEE TKDE*, 2020.

[23] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, "Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer," in *KDD*, 2019, pp. 1441–1450.

[24] X. Chen, D. Liu, C. Lei, R. Li, Z.-J. Zha, and Z. Xiong, "Bert4sessrec: Content-based video relevance prediction with bidirectional encoder representations from transformer," in *ACM Multimedia*, 2019, pp. 2597–2601.

[25] K. Zhou, H. Wang, W. X. Zhao, Y. Zhu, S. Wang, F. Zhang, Z. Wang, and J.-R. Wen, "S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization," in *CIKM*, 2020, pp. 1893–1902.

[26] P. Goyal, M. Caron, B. Lefaudeux, M. Xu, P. Wang, V. Pai, M. Singh, V. Liptchinsky, I. Misra, A. Joulin *et al.*, "Self-supervised pretraining of visual features in the wild," *arXiv preprint arXiv:2103.01988*, 2021.

[27] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A survey on contrastive self-supervised learning," *Technologies*, vol. 9, no. 1, p. 2, 2021.

[28] X. Xia, H. Yin, J. Yu, Q. Wang, L. Cui, and X. Zhang, "Self-supervised hypergraph convolutional networks for session-based recommendation," in *AAAI*, 2021, pp. 4503–4511.

[29] J. Yu, H. Yin, J. Li, Q. Wang, N. Q. V. Hung, and X. Zhang, "Self-supervised multi-channel hypergraph convolutional network for social recommendation," in *WWW*, 2021, pp. 413–424.

[30] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *SIGIR*, 2021, pp. 726–735.

[31] D. Lee, S. Kang, H. Ju, C. Park, and H. Yu, "Bootstrapping user and item representations for one-class collaborative filtering," *arXiv preprint arXiv:2105.06323*, 2021.

[32] X. Xin, A. Karatzoglou, I. Arapakis, and J. M. Jose, "Self-supervised reinforcement learning for recommender systems," in *SIGIR*, 2020, pp. 931–940.

[33] R. Qiu, Z. Huang, and H. Yin, "Memory augmented multi-instance contrastive predictive coding for sequential recommendation," in *ICDM*, 2021.

[34] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE TPAMI*, 2020.

[35] L. Wu, H. Lin, Z. Gao, C. Tan, S. Li *et al.*, "Self-supervised on graphs: Contrastive, generative, or predictive," *arXiv preprint arXiv:2105.07342*, 2021.

[36] Y. Liu, S. Pan, M. Jin, C. Zhou, F. Xia, and P. S. Yu, "Graph self-supervised learning: A survey," *arXiv preprint arXiv:2103.00111*, 2021.

[37] Y. Xie, Z. Xu, J. Zhang, Z. Wang, and S. Ji, "Self-supervised learning of graph neural networks: A unified review," *arXiv preprint arXiv:2102.10757*, 2021.

[38] H. Yin, B. Cui, J. Li, J. Yao, and C. Chen, "Challenging the long tail recommendation," *VLDB*, vol. 5, no. 9, pp. 896–907, 2012.

[39] J. Chen, H. Dong, X. Wang, F. Feng, M. Wang, and X. He, "Bias and debias in recommender system: A survey and future directions," *arXiv preprint arXiv:2010.03240*, 2020.

[40] W.-C. Kang, D. Z. Cheng, T. Yao, X. Yi, T. Chen, L. Hong, and E. H. Chi, "Learning to embed categorical features without embedding tables for recommendation," in *KDD*, 2021, pp. 840–850.

[41] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, "A survey on neural recommendation: From collaborative filtering to content and context enriched recommendation," *arXiv preprint arXiv:2104.13030*, 2021.

[42] M. Gao, J. Zhang, J. Yu, J. Li, J. Wen, and Q. Xiong, "Recommender systems based on generative adversarial networks: A problem-driven perspective," *Information Sciences*, vol. 546, pp. 1166–1185, 2021.

[43] Y. Deldjoo, T. D. Noia, and F. A. Merra, "A survey on adversarial recommender systems: from attack/defense strategies to generative adversarial networks," *ACM Computing Surveys (CSUR)*, vol. 54, no. 2, pp. 1–38, 2021.

[44] J. Xia, Y. Zhu, Y. Du, and S. Z. Li, "A survey of pretraining on graphs: Taxonomy, methods, and applications," *arXiv preprint arXiv:2202.07893*, 2022.

[45] Z. Zeng, C. Xiao, Y. Yao, R. Xie, Z. Liu, F. Lin, L. Lin, and M. Sun, "Knowledge transfer via pre-training for recommendation: A review and prospect," *Frontiers in big Data*, vol. 4, 2021.

[46] Z. Meng, S. Liu, C. Macdonald, and I. Ounis, "Graph neural pre-training for enhancing recommendations using side information," *arXiv preprint arXiv:2107.03936*, 2021.

[47] G. d. S. P. Moreira, S. Rabhi, R. Ak, M. Y. Kabir, and E. Oldridge, "Transformers with multi-modal features and post-fusion context for e-commerce session-based recommendation," *arXiv preprint arXiv:2107.05124*, 2021.

[48] A. Sankar, Y. Wu, Y. Wu, W. Zhang, H. Yang, and H. Sundaram, "Groupim: A mutual information maximization framework for neural group recommendation," *arXiv preprint arXiv:2006.03736*, 2020.

[49] Y. Wei, X. Wang, Q. Li, L. Nie, Y. Li, X. Li, and T.-S. Chua, "Contrastive learning for cold-start recommendation," in *ACM Multimedia*, 2021, pp. 5382–5390.

[50] C. Lei, Y. Liu, L. Zhang, G. Wang, H. Tang, H. Li, and C. Miao, "Semi: A sequential multi-modal information transfer network for e-commerce micro-video recommendations," in *KDD*, 2021, pp. 3161–3171.

[51] Y. Qin, P. Wang, and C. Li, "The world is binary: Contrastive learning for denoising next basket recommendation," in *SIGIR*, 2021, pp. 859–868.

[52] C. Zhou, J. Ma, J. Zhang, J. Zhou, and H. Yang, "Contrastive learning for debiased candidate generation in large-scale recommender systems," in *KDD*, 2021, pp. 3985–3995.

[53] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE TNNLS*, 2020.

[54] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[55] X. Xia, H. Yin, J. Yu, Y. Shao, and L. Cui, "Self-supervised graph co-training for session-based recommendation," in *CIKM*, 2021.

[56] J. Yu, H. Yin, M. Gao, X. Xia, X. Zhang, and N. Q. V. Hung, "Socially-aware self-supervised tri-training for recommendation," in *KDD*. ACM, 2021, pp. 2084–2092.

[57] K. Ding, Z. Xu, H. Tong, and H. Liu, "Data augmentation for deep graph learning: A survey," *arXiv preprint arXiv:2202.08235*, 2022.

[58] B. Li, Y. Hou, and W. Che, "Data augmentation approaches in natural language processing: A survey," *arXiv preprint arXiv:2110.01852*, 2021.

[59] Y. Tian, C. Sun, B. Poole, D. Krishnan, C. Schmid, and P. Isola, "What makes for good views for contrastive learning?" *NeurIPS*, vol. 33, pp. 6827–6839, 2020.

[60] M. Cheng, F. Yuan, Q. Liu, X. Xin, and E. Chen, "Learning transferable user representations with sequential behaviors via contrastive pre-training," in *ICDM*. IEEE, 2021, pp. 51–60.

[61] X. Xie, F. Sun, Z. Liu, S. Wu, J. Gao, B. Ding, and B. Cui, "Contrastive learning for sequential recommendation," *arXiv preprint arXiv:2010.14395*, 2020.

[62] Y. Li, H. Chen, X. Sun, Z. Sun, L. Li, L. Cui, P. S. Yu, and G. Xu, "Hyperbolic hypergraphs for sequential recommendation," in *CIKM*, 2021, pp. 988–997.

[63] S. Bian, W. X. Zhao, K. Zhou, J. Cai, Y. He, C. Yin, and J.-R. Wen, "Contrastive curriculum learning for sequential user behavior modeling via data augmentation," in *CIKM*, 2021, pp. 3737–3746.

[64] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-based recommendations with recurrent neural networks," in *ICLR*, 2016.

[65] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme, "Factorizing personalized markov chains for next-basket recommendation," in *WWW*, 2010, pp. 811–820.

[66] S. Wang, L. Hu, Y. Wang, L. Cao, Q. Z. Sheng, and M. A. Orgun, "Sequential recommender systems: Challenges, progress and prospects," in *IJCAI*, 2019, pp. 6332–6338.

[67] Z. Liu, Y. Chen, J. Li, P. S. Yu, J. McAuley, and C. Xiong, "Contrastive self-supervised sequential recommendation with robust augmentation," *arXiv preprint arXiv:2108.06479*, 2021.

[68] X. Yuan, H. Chen, Y. Song, X. Zhao, and Z. Ding, "Improving sequential recommendation consistency with self-supervised imitation," in *IJCAI*, 2021, pp. 3321–3327.

[69] J. Zhang, M. Gao, J. Yu, L. Guo, J. Li, and H. Yin, "Double-scale self-supervised hypergraph learning for group recommendation," in *CIKM*, 2021.

[70] H. Wang, K. Zhou, W. X. Zhao, J. Wang, and J.-R. Wen, "Curriculum pre-training heterogeneous subgraph transformer for top-n recommendation," *arXiv preprint arXiv:2106.06722*, 2021.

[71] X. Zhou, A. Sun, Y. Liu, J. Zhang, and C. Miao, "Selfcf: A simple framework for self-supervised collaborative filtering," *arXiv preprint arXiv:2107.03019*, 2021.

[72] B. Hao, H. Yin, J. Zhang, C. Li, and H. Chen, "A multi-strategy based pre-training method for cold-start recommendation," *arXiv preprint arXiv:2112.02275*, 2021.

[73] Y. Yang, L. Wu, R. Hong, K. Zhang, and M. Wang, "Enhanced graph learning for collaborative filtering via mutual information maximization," in *SIGIR*, 2021, pp. 71–80.

[74] H. Yang, H. Chen, L. Li, S. Y. Philip, and G. Xu, "Hyper meta-path contrastive learning for multi-behavior recommendation," in *ICDM*. IEEE, 2021, pp. 787–796.

[75] J. Cao, X. Lin, S. Guo, L. Liu, T. Liu, and B. Wang, "Bipartite graph embedding via mutual information maximization," in *WSDM*, 2021, pp. 635–643.

[76] R. Xie, Q. Liu, L. Wang, S. Liu, B. Zhang, and L. Lin, "Contrastive cross-domain recommendation in matching," *arXiv preprint arXiv:2112.00999*, 2021.

[77] C. Wang, Y. Liang, Z. Liu, T. Zhang, and P. S. Yu, "Pre-training graph neural network for cross domain recommendation," *arXiv preprint arXiv:2111.08268*, 2021.

[78] T. Yao, X. Yi, D. Z. Cheng, F. Yu, T. Chen, A. Menon, L. Hong, E. H. Chi, S. Tjoa, J. Kang *et al.*, "Self-supervised learning for large-scale item recommendations," in *CIKM*, 2021, pp. 4321–4330.

[79] R. Qiu, Z. Huang, H. Yin, and Z. Wang, "Contrastive learning for representation degeneration problem in sequential recommendation," in *WSDM*. ACM, 2022, pp. 813–823.

[80] Z. Liu, Y. Chen, J. Li, M. Luo, S. Y. Philip, and C. Xiong, "Self-supervised learning for sequential recommendation with model augmentation," 2021.

[81] Z. Xie, C. Liu, Y. Zhang, H. Lu, D. Wang, and Y. Ding, "Adversarial and contrastive variational autoencoder for sequential recommendation," in *WWW*, 2021, pp. 449–459.

[82] Y. Chen, Z. Liu, J. Li, J. McAuley, and C. Xiong, "Intent contrastive learning for sequential recommendation," *arXiv preprint arXiv:2202.02519*, 2022.

[83] Z. Lin, C. Tian, Y. Hou, and W. X. Zhao, "Improving graph collaborative filtering with neighborhood-enriched contrastive learning," *arXiv preprint arXiv:2202.06200*, 2022.

[84] J. Li, P. Zhou, C. Xiong, and S. C. Hoi, "Prototypical contrastive learning of unsupervised representations," *arXiv preprint arXiv:2005.04966*, 2020.

[85] T. Huang, Y. Dong, M. Ding, Z. Yang, W. Feng, X. Wang, and J. Tang, "Mixgcf: An improved training method for graph neural network-based recommender systems," 2021.

[86] Y. Kalantidis, M. B. Sariyildiz, N. Pion, P. Weinzaepfel, and D. Larlus, "Hard negative mixing for contrastive learning," in *NeurIPS 2020*, 2020.

[87] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *SIGIR*. ACM, 2020, pp. 639–648.

[88] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," in *UAI*. AUAI Press, 2009, pp. 452–461.

[89] Z. Liu, Y. Ma, Y. Ouyang, and Z. Xiong, "Contrastive learning for recommender system," *arXiv preprint arXiv:2101.01317*, 2021.

[90] H. Yin, Q. Wang, K. Zheng, Z. Li, J. Yang, and X. Zhou, "Social influence-based group representation learning for group recommendation," in *ICDE*. IEEE, 2019, pp. 566–577.

[91] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR*, 2018.

[92] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?" in *ICLR*, 2019.

[93] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[94] C. Wang, W. Ma, and C. Chen, "Sequential recommendation with multiple contrast signals," *ACM TOIS*, 2022.

[95] D. Cai, S. Qian, Q. Fang, J. Hu, W. Ding, and C. Xu, "Heterogeneous graph contrastive learning network for personalized micro-video recommendation," *IEEE Transactions on Multimedia*, 2022.

[96] J. Yu, M. Gao, J. Li, H. Yin, and H. Liu, "Adaptive implicit friends identification over heterogeneous network for social recommendation," in *CIKM*. ACM, 2018, pp. 357–366.

[97] J. Yu, M. Gao, H. Yin, J. Li, C. Gao, and Q. Wang, "Generating reliable friends via adversarial training to improve social recommendation," in *ICDM*. IEEE, 2019, pp. 768–777.

[98] X. Long, C. Huang, Y. Xu, H. Xu, P. Dai, L. Xia, and L. Bo, "Social recommendation with self-supervised metagraph informax network," in *CIKM*, 2021, pp. 1160–1169.

[99] W. Guo, C. Zhang, Z. He, J. Qin, H. Guo, B. Chen, R. Tang, X. He, and R. Zhang, "Miss: Multi-interest self-supervised learning framework for click-through rate prediction," *arXiv preprint arXiv:2111.15068*, 2021.

[100] T. Gao, X. Yao, and D. Chen, "Simcse: Simple contrastive learning of sentence embeddings," *arXiv preprint arXiv:2104.08821*, 2021.

[101] J. Yu, H. Yin, X. Xia, L. Cui, and Q. V. H. Nguyen, "Graph augmentation-free contrastive learning for recommendation," *arXiv preprint arXiv:2112.08679*, 2021.

[102] Z. Liu, Y. Chen, J. Li, M. Luo, S. Y. Philip, and C. Xiong, "Self-supervised learning for sequential recommendation with model augmentation," 2021.

[103] W. Wei, C. Huang, L. Xia, Y. Xu, J. Zhao, and D. Yin, "Contrastive meta learning with behavior multiplicity for recommendation," in *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 2022, pp. 1120–1128.

[104] Y. Wu, R. Xie, Y. Zhu, X. Ao, X. Chen, X. Zhang, F. Zhuang, L. Lin, and Q. He, "Multi-view multi-behavior contrastive learning in recommendation," *arXiv preprint arXiv:2203.10576*, 2022.

[105] S. Nowozin, B. Cseke, and R. Tomioka, "f-gan: Training generative neural samplers using variational divergence minimization," *NeurIPS*, vol. 29, 2016.

[106] M. I. Belghazi, A. Baratin, S. Rajeshwar, S. Ozair, Y. Bengio, A. Courville, and D. Hjelm, "Mutual information neural estimation," in *ICML*. PMLR, 2018, pp. 531–540.

[107] R. D. Hjelm, A. Fedorov, S. Lavoie-Marchildon, K. Grewal, P. Bachman, A. Trischler, and Y. Bengio, "Learning deep representations by mutual information estimation and maximization," *arXiv preprint arXiv:1808.06670*, 2018.

[108] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*. JMLR Workshop and Conference Proceedings, 2010, pp. 297–304.

[109] T. Wang and P. Isola, "Understanding contrastive representation learning through alignment and uniformity on the hypersphere," in *ICML*. PMLR, 2020, pp. 9929–9939.

[110] J. Wu, X. Wang, X. Gao, J. Chen, H. Fu, T. Qiu, and X. He, "On the effectiveness of sampled softmax loss for item recommendation," *arXiv preprint arXiv:2201.02327*, 2022.

[111] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *CVPR*, 2021, pp. 2495–2504.

[112] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *NeurIPS*, vol. 33, pp. 18 661–18 673, 2020.

[113] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *ICDM*. IEEE, 2018, pp. 197–206.

[114] Q. Zhang, J. Li, Q. Jia, C. Wang, J. Zhu, Z. Wang, and X. He, "Unbert: User-news matching bert for news recommendation," in *IJCAI*, 2021, pp. 3356–3362.

[115] C. Wu, F. Wu, T. Qi, and Y. Huang, "Empowering news recommendation with pre-trained language models," in *SIGIR*, 2021, pp. 1652–1656.

[116] Z. Qiu, X. Wu, J. Gao, and W. Fan, "U-bert: Pre-training user representations for improved recommendation," in *AAAI*, 2021, pp. 1–8.

[117] F. Yuan, X. He, H. Jiang, G. Guo, J. Xiong, Z. Xu, and Y. Xiong, "Future data helps training: Modeling future contexts for session-based recommendation," in *WWW*, 2020, pp. 303–313.

[118] C. Xiao, R. Xie, Y. Yao, Z. Liu, M. Sun, X. Zhang, and L. Lin, "Uprec: User-aware pre-training for recommender systems," *arXiv preprint arXiv:2102.10989*, 2021.

[119] F. Yuan, X. He, A. Karatzoglou, and L. Zhang, "Parameter-efficient transfer from sequential behaviors for user modeling and recommendation," in *SIGIR*, 2020, pp. 1469–1478.

[120] K. Shin, H. Kwak, K.-M. Kim, M. Kim, Y.-J. Park, J. Jeong, and S. Jung, "One4all user representation for recommender systems in e-commerce," *arXiv preprint arXiv:2106.00573*, 2021.

[121] J. Shang, T. Ma, C. Xiao, and J. Sun, "Pre-training of graph augmented transformers for medication recommendation," *arXiv preprint arXiv:1906.00346*, 2019.

[122] Y. Liu, S. Yang, C. Lei, G. Wang, H. Tang, J. Zhang, A. Sun, and C. Miao, "Pre-training graph transformer with multimodal side information for recommendation," in *ACM Multimedia*, 2021, pp. 2853–2861.

[123] J. Ma, C. Zhou, H. Yang, P. Cui, X. Wang, and W. Zhu, "Disentangled self-supervision in sequential recommenders," in *KDD*, 2020, pp. 483–491.

[124] B. Hao, J. Zhang, H. Yin, C. Li, and H. Chen, "Pre-training graph neural networks for cold-start users and items representation," in *WSDM*, 2021, pp. 265–273.

[125] J. Zhang, B. Bai, Y. Lin, J. Liang, K. Bai, and F. Wang, "General-purpose user embeddings based on mobile app usage," in *KDD*, 2020, pp. 2831–2840.

[126] B. Zoph, G. Ghiasi, T.-Y. Lin, Y. Cui, H. Liu, E. D. Cubuk, and Q. Le, "Rethinking pre-training and self-training," *NeurIPS*, vol. 33, pp. 3833–3845, 2020.

[127] Z. Liu, Z. Fan, Y. Wang, and P. S. Yu, "Augmenting sequential recommendation with pseudo-prior items via reversely pre-training transformer," in *SIGIR*, 2021, pp. 1608–1612.

[128] J. Jiang, Y. Luo, J. B. Kim, K. Zhang, and S. Kim, "Sequential recommendation with bidirectional chronological augmentation of transformer," *arXiv preprint arXiv:2112.06460*, 2021.

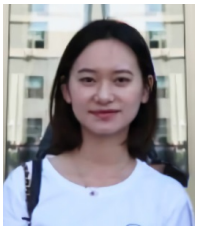
[129] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *CLOT*, 1998, pp. 92–100.

[130] H. Chen, H. Yin, T. Chen, Q. V. H. Nguyen, W.-C. Peng, and X. Li, "Exploiting centrality information with graph convolutions for network representation learning," in *ICDE*. IEEE, 2019, pp. 590–601.

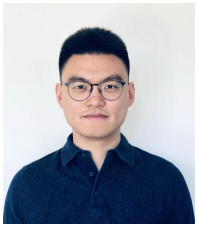
[131] C. Wu, F. Wu, T. Qi, J. Lian, Y. Huang, and X. Xie, "PTUM: pre-training user model from unlabeled user behaviors via self-supervision," in *EMNLP*, 2020.
[132] H. Liu, D. Tang, J. Yang, X. Zhao, J. Tang, and Y. Cheng, "Self-supervised learning for alleviating selection bias in recommendation systems," 2021.
[133] M. Ma, P. Ren, Z. Chen, Z. Ren, H. Liang, J. Ma, and M. de Rijke, "Improving transformer-based sequential recommenders through preference editing," *arXiv preprint arXiv:2106.12120*, 2021.
[134] Y. Tao, M. Gao, J. Yu, Z. Wang, Q. Xiong, and X. Wang, "Predictive and contrastive: Dual-auxiliary learning for recommendation," *arXiv preprint arXiv:2203.03982*, 2022.
[135] J. Ma, C. Zhou, P. Cui, H. Yang, and W. Zhu, "Learning disentangled representations for recommendation," *NeurIPS*, vol. 32, 2019.
[136] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *ICML*, 2009, pp. 41–48.
[137] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE TKDE*, vol. 17, no. 11, pp. 1529–1541, 2005.
[138] M. Kim, J. Tack, and S. J. Hwang, "Adversarial self-supervised contrastive learning," *NeurIPS*, vol. 33, pp. 2983–2994, 2020.
[139] J. Ma, Z. Zhao, X. Yi, J. Chen, L. Hong, and E. H. Chi, "Modeling task relationships in multi-task learning with multi-gate mixture-of-experts," in *KDD*, 2018, pp. 1930–1939.
[140] W. X. Zhao, S. Mu, Y. Hou, Z. Lin, Y. Chen, X. Pan, K. Li, Y. Lu, H. Wang, C. Tian *et al.*, "Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms," in *CIKM*, 2021, pp. 4653–4664.
[141] T. Xiao, X. Wang, A. A. Efros, and T. Darrell, "What should not be contrastive in contrastive learning," in *ICLR*, 2021.
[142] Z. Wang, M. Gao, J. Li, J. Zhang, and J. Zhong, "Gray-box shilling attack: An adversarial learning approach," *ACM TIST*, 2022.
[143] H. Zhang, C. Tian, Y. Li, L. Su, N. Yang, W. X. Zhao, and J. Gao, "Data poisoning attack against recommender system using incomplete and perturbed data," in *KDD*, 2021, pp. 2154–2164.
[144] A. Saha, A. Tejankar, S. A. Koohpayegani, and H. Pirsiavash, "Backdoor attacks on self-supervised learning," *arXiv preprint arXiv:2105.10123*, 2021.
[145] S. Zhang, H. Chen, X. Sun, Y. Li, and G. Xu, "Unsupervised graph poisoning attack via contrastive loss back-propagation," *arXiv preprint arXiv:2201.07986*, 2022.
[146] Q. Wang, H. Yin, T. Chen, J. Yu, A. Zhou, and X. Zhang, "Fast-adapting and privacy-preserving federated recommender system," *The VLDB Journal*, pp. 1–20, 2021.
[147] Q. Wang, H. Yin, T. Chen, Z. Huang, H. Wang, Y. Zhao, and N. Q. Viet Hung, "Next point-of-interest recommendation on resource-constrained mobile devices," in *WWW*, 2020, pp. 906–916.
[148] Y. Yu, F. Wu, C. Wu, J. Yi, T. Qi, and Q. Liu, "Tiny-newsrec: Efficient and effective plm-based news recommendation," *arXiv preprint arXiv:2112.00944*, 2021.
[149] Y. Wang, Y. Zhang, and M. Coates, "Graph structure aware contrastive knowledge distillation for incremental learning in recommender systems," in *CIKM*, 2021, pp. 3518–3522.
[150] L. Huang, Y. Liu, X. Zhou, A. You, M. Li, B. Wang, Y. Zhang, P. Pan, and X. Yinghui, "Once and for all: Self-supervised multi-modal co-training on one-billion videos at alibaba," in *ACM MM*, 2021, pp. 1148–1156.

Junliang Yu received his B.S. and M.S degrees in Software Engineering from Chongqing University, China. Currently, he is a third-year Ph.D. candidate at the School of Information Technology and Electrical Engineering, the University of Queensland. His research interests include recommender systems, social media analytics, and self-supervised learning.

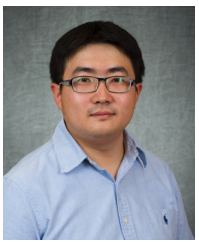
Hongzhi Yin is an associate professor and ARC Future Fellow at the University of Queensland. He received his Ph.D. degree from Peking University, in 2014. His research interests include recommendation system, deep learning, social media mining, and federated learning. He is currently serving as Associate Editor/Guest Editor/Editorial Board for ACM Transactions on Information Systems (TOIS), ACM Transactions on Intelligent Systems and Technology (TIST), etc.



Xin Xia received her B.S. degree in Software Engineering from Jilin University, China. Currently, she is a third-year Ph.D. candidate at the School of Information Technology and Electrical Engineering, the University of Queensland. Her research interests include knowledge distillation, sequence modeling, and self-supervised learning.



Tong Chen received his PhD degree in computer science from The University of Queensland in 2020. He is currently a Lecturer with the Data Science research group, School of Information Technology and Electrical Engineering, The University of Queensland. His research interests include data mining, recommender systems, user behavior modelling and predictive analytics.



Jundong Li is an assistant professor at the University of Virginia. Prior to joining UVA, he received his Ph.D. degree in Computer Science at Arizona State University in 2019, M.Sc. degree in Computer Science at University of Alberta in 2014, and B.Eng. degree in Software Engineering at Zhejiang University in 2012. His research interests are broadly in data mining and machine learning, with a particular focus on feature learning, graph mining, and social computing.

Zi Huang is a Professor and ARC Future Fellow in School of ITEE, The University of Queensland (UQ). She received her BSc degree from Department of Computer Science, Tsinghua University, China, and her PhD in Computer Science from UQ in 2001 and 2007 respectively. Dr. Huang's research interests mainly include multimedia indexing and search, social data analysis and knowledge discovery. She is currently an Associate Editor of The VLDB Journal, ACM Transactions on Information Systems (TOIS), etc.