Graph Masked Autoencoder for Sequential Recommendation

Yaowen Ye
elwin@connect.hku.hk
University of Hong Kong
Hong Kong SAR, China

Lianghao Xia
aka_xia@foxmail.com
University of Hong Kong
Hong Kong SAR, China

Chao Huang*
chaohuang75@gmail.com
University of Hong Kong
Hong Kong SAR, China

ABSTRACT

While some powerful neural network architectures (*e.g.*, Transformer, Graph Neural Networks) have achieved improved performance in sequential recommendation with high-order item dependency modeling, they may suffer from poor representation capability in label scarcity scenarios. To address the issue of insufficient labels, Contrastive Learning (CL) has attracted much attention in recent methods to perform data augmentation through embedding contrasting for self-supervision. However, due to the hand-crafted property of their contrastive view generation strategies, existing CL-enhanced models i) can hardly yield consistent performance on diverse sequential recommendation tasks; ii) may not be immune to user behavior data noise. In light of this, we propose a simple yet effective graph masked autoencoder that adaptively and dynamically distills global item transitional information for self-supervised augmentation. It naturally avoids the above issue of heavy reliance on constructing high-quality embedding contrastive views. Instead, an adaptive data reconstruction paradigm is designed to be integrated with the long-range item dependency modeling, for informative augmentation in sequential recommendation. Extensive experiments demonstrate that our method significantly outperforms state-of-the-art baseline models and can learn more accurate representations against data noise and sparsity. Our implemented model code is available at https://github.com/HKUDS/GMRec.

CCS CONCEPTS

• **Information systems** → **Recommender systems**.

KEYWORDS

Sequential Recommendation, Masked Autoencoder, Graph Neural Networks, Self-Supervised Learning

---

*Chao Huang is the corresponding author.

---

1 INTRODUCTION

Sequential recommendation aims to learn effective representations of users' interests and suggest future items that may be of interest to different users [43–45]. This task has attracted considerable attention, given that user preference is time-evolving in nature in real-life, such as e-commerce [34] and video streaming [37] sites. To capture high-order transition relationships between items, extensive research efforts [5, 25, 35, 38] have been devoted to proposing various graph neural networks for improving recommendation by recursively propagating information among adjacent items.

Behavior sequences of users in sequential recommender systems generally follow a long-tail distribution, in which a larger number of users merely interact with very few items [20, 23]. While Graph Neural Networks (GNNs) have achieved promising results for approaching the sequential recommendation task in a fully supervised manner, the label data sparsity issue could significantly degrade the model representation performance [40, 47]. To tackle the label insufficiency issue, recent attempts [24, 26, 40] have been made to bring the benefits of Contrastive Learning (CL) into sequential recommender systems to provide auxiliary self-supervision as augmentation signals. Following the mutual information maximization (InfoMax) framework, the key idea of these CL approaches is to reach an agreement between the embeddings encoded from two augmented views based on the InfoNCE objective. They introduce various augmentation schemes to corrupt item sequence structures based on different heuristics, *e.g.*, stochastic item masking and reordering in CL4SRec [40], dropout-based semantic preservation in DouRec [26], and substitute and insert operations in CoSeRec [24].

Despite recent advances, existing CL-based sequential recommender systems are severely limited by the following factors:

(1) **Hand-crafted Contrastive Augmentation**. State-of-the-art Contrastive Learning (CL) methods still heavily rely on manually designed data corruption schemes based on heuristics to construct views for embedding contrast. Then, the representation consistency is maximized within the positive pair and minimized between negative instances [40]. Guided by the inherent design of the InfoMax principle, the success of current contrastive self-supervision largely relies on high-quality augmentation with accurate contrastive view generation. However, it often requires domain knowledge with arduous labor, which can hardly be adaptable to diverse sequential recommendation scenarios with limited supervision. Furthermore, blindly corrupting the sequence structures for generating augmented views may damage the important transition structures and impair the representation learning over short sequences.

(2) **Noise Perturbation for Data Augmentation**. Data noise issues, such as interactive behavior noise and popularity bias, commonly exist in real-life recommender systems [3, 30, 39]. Trained
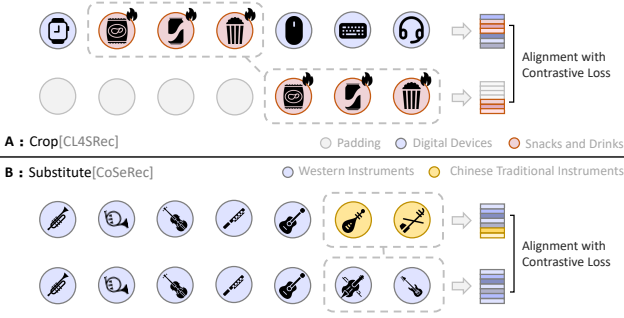
Figure 1: Motivating examples of noisy and biased data augmentation in existing CL-based sequential recommenders.

Self-Supervised Learning (SSL) models can be easily misled by spurious item transitional correlations, leading to suboptimal performance. Particularly, the noisy information can propagate through the item-wise connections, which negatively affects the representation learning with contrastive augmentation techniques. In the face of data noise issues, contrastive self-supervised learners are likely to overfit to the noisy labels and result in poor performance. Therefore, to build SSL-enhanced sequential recommenders that are robust to noise perturbation in practical scenarios, we need to mitigate the noisy effects in the self-supervision process with adaptive and selective augmentation for encoding interaction dynamics.

Taking the augmentation strategies used in existing CL-based sequential recommenders as examples, as depicted in Figure 1, user A is a digital enthusiast with the purchase of many digital products. Influenced by the New Year sale in online retailers, he/she also buys some popular snacks and drinks due to the position bias of items. However, the crop operations in CL4SRec [40] may drop important interaction data and keep biased information for augmentation. In such cases, contrastive alignment between noisy augmented views will unavoidably impair the self-supervision quality and result in misleading user preference modeling. Similarly, the substitute augmentation (in CoSeRec [24]) on the item sequence of user B may lose the limited but important interaction data on long-tail items (*e.g.*, *erhu*, *pipa*). Some other more popular instruments (*e.g.*, *guitar*) are used for substitution, which may worsen the recommendation performance on long-tail items with limited interaction labels.

In light of the aforementioned limitations, an interesting question naturally arises: *How to build SSL-enhanced sequential recommender systems that are easily adaptable and noise-resistant?* Inspired by the recent success of autoencoder-based masking techniques in image data augmentation [11, 13], generative self-supervised learning with the goal of masked data reconstruction can naturally alleviate the heavy reliance on manually constructing high-quality contrastive views for accurate embedding alignment.

Contribution. In this work, we develop a new way to enhance sequential recommender systems with robust and adaptable self-supervised learning. To do so, we propose a lightweight and principled graph masked autoencoder model (named GMRec) to automate the self-supervised augmentation process. GMRec adaptively and dynamically distills informative signals for reconstruction in response to changing sequential recommendation environments.

First, to mitigate the noisy effects during augmentation, we automatically identify items with higher semantic consistency through a learnable masking scheme. We introduce task-adaptive regularization to enhance self-supervision with an awareness of downstream recommendation tasks. *Second*, we selectively mask out item transition paths with higher helpfulness and feed them into the graph autoencoder for reconstruction. By doing so, adaptive augmentation is seamlessly integrated with the modeling of sequential item dependency to cancel out the bias introduced by interaction noise and task-irrelevant information. *Finally*, in accordance with adaptive augmentation by GMRec, we supplement the main recommendation task with noise-resistant SSL signals.

In this way, our GMRec model can not only conduct strong augmentation to train robust sequential recommenders but also regulate self-supervised learning with task-adaptive semantics via the generative reconstruction task. Our model requires no heuristics and is generalizable to various sequential interaction patterns on different datasets. Extensive experiments under various settings demonstrate the superiority of our algorithm compared to various baselines. The performed model ablation and robustness studies justify how adaptive SSL augmentation mitigates the impact of data noise and sparsity. In addition, our method achieves comparable efficiency when competing with baselines.

2 PRELIMINARIES AND RELATED WORK

Sequential Recommendation. Suppose $\mathcal{U}$ and $\mathcal{V}$ represent the sets of users and items, respectively. Each user ($u \in \mathcal{U}$) is associated with a temporally ordered sequence of their historical interacted items, denoted as $S^u = (s_1^u, s_2^u, \cdots, s_{l_u}^u)$. $s_t^u \in \mathcal{V}$ denotes the t-th interacted item of user u, and l_u is the length of u's item sequence S^u. Given an interaction sequence S^u of a user, the target of sequential recommendation is to predict the next item $s_{l_u+1}^u$ that user u is most likely to interact with (*e.g.*, click or purchase).

To study the sequential recommendation problem, a surge of approaches have been developed to encode sequential behavioral patterns using different techniques, *e.g.*, recurrent neural networks (RNNs) [15], convolutional neural networks [29, 41]. Inspired by the Transformer architecture, the self-attention mechanism has been adopted to deal with pairwise item correlations in user behavior sequences [9, 10, 19]. In addition, with the development of graph neural networks (GNNs), graph-augmented representation encoders are built upon GNN-based frameworks to capture the long-term sequential preference of users via multi-layer message passing, such as SRGNN [38], GCE-GNN [35], and SURGE [2]. However, most existing supervised approaches face the label sparsity issue, which significantly limits the model representation ability in real-life recommender systems.

Contrastive Learning (CL) for Sequential Recommenders. Recently, CL has attracted significant attention for addressing the challenge of label supervision deficiency [1] and learning accurate representations of item sequential dependencies [22]. In general, sequential recommendation models with contrastive learning aim to explore self-supervised signals from unlabeled data [42, 46]. For data augmentation, existing CL-based recommender systems excessively rely on handcrafted designs to construct contrastive representation views for reaching embedding agreement.

Data corruption is performed on user behavior sequences with various augmentation operations to generate contrastive views. For

example, CL4SRec [40] performs sequence randomly masking and reordering operations to establish augmented contrastive views. DuoRec [26] applies model-level augmentation with dropout masks. To capture diverse user intents of time-ordered interaction behaviors, ICLRec [7] proposes to model latent variables corresponding to user intents with contrastive self-supervision. The effectiveness of their contrastive learning methods largely depends on the high quality of generated positive and negative samples, which is not always guaranteed across different recommendation scenarios.

Masked Autoencoder for Representation Learning. Inspired by the success of generative self-supervised learning in language representation [8], masked autoencoders (MAE) have been proposed as effective learners for understanding images [11]. In these MAE-based vision learners, pixel-level image reconstruction is conducted using the encoder-decoder framework. Additionally, vision-language masked autoencoders have been developed to learn multimodal embeddings to pair image-text instances [13]. Motivated by this masking-then-prediction paradigm [6], MAE has been introduced to graph representation learning in recent attempts. For instance, GraphMAE [16] focuses on reconstructing node features through randomly masking based on the scaled cosine error measurement. S2GAE [28] randomly drops node-wise connections from the input graph structure for reconstruction. Building upon this line, our approach incorporates the benefits of masked autoencoding and advances it with adaptive graph masking to provide meaningful self-supervisory signals for sequential recommendation.

3 METHODOLOGY

In this section, we introduce our GMRec recommender system and illustrate the model architecture overview in Figure 2.

3.1 Adaptive Transition Path Masking

Different from the random masking strategy in some existing masked autoencoders [16, 28], our graph masking mechanism is designed to achieve the goal of adaptive self-supervised augmentation. In this way, the graph masked autoencoder is generalized to automatically and dynamically distill useful information for masking on diverse recommendation data. Furthermore, to empower our model to preserve the global context of long-range item transitions, we propose masking graph path information with high-order item connections, instead of masking single nodes or edges.

To supercharge our model with cross-sequence item transition information and unleash the potential of graph masked autoencoder, we construct a global item-item transition graph over sequences.

Global Item Transition Graph. To capture the dependence among different users, we generate a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent the transition relationships among different items by considering user behavior sequences (*e.g.*, S^u, $u \in \mathcal{U}$). The item set $\mathcal{V}$ serves as the vertex set of the transition graph. For the edge set $\mathcal{E}$, we go through all user sequences and build an edge between each item and its h-hop neighbors in each sequence. We count duplicated edges only once. Formally, the edge set of the transition graph is built as:

$$\mathcal{E} = \{(s_t^u, s_{t'}^u) : u \in \mathcal{U}, |t - t'| \le h, 1 \le t, t' \le l_u\} \quad (1)$$

Given the constructed item transition graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the overall adaptive path masking paradigm with learnable data augmentation

policy consists of several key modules: (i) **Learning to Mask**–automatically discovering the set of anchor nodes for masked path construction; (ii) **Transition Path Masking**–masking paths underlying the graph structures to inherit informative transition patterns; (iii) **Task-Adaptive Augmentation**–recalibrating the self-supervised graph masking process with task-adaptive contexts.

3.1.1 **Learning to Mask.** The target of the learning to mask module is to learn a strategy for generating the set of anchor nodes $\mathcal{V}_a$ for the learnable graph augmenter. Inspired by the InfoMax principle between path-level and graph-level embeddings [18, 31], we propose to measure the semantic relatedness between items by deriving their representation consistency over the graph structure of global item transitions $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In particular, we sample the k-hop transition neighbors of the target item node v on graph $\mathcal{G}$ to generate the transition subgraph for v and represent its surrounding context given the generated global item transition relationships.

Items that have noisy interactions or are biased towards popularity may have diverse distributions in their embeddings (encoded with spurious graph connections), which can lead to suppression of their semantic relatedness scores. Therefore, we sample items with higher semantic relatedness scores as anchor nodes $\mathcal{V}_a$ from $\mathcal{G}$.

To formalize this approach, we define the semantic relatedness $\gamma(v)$ between a node $v \in \mathcal{V}$ and its kth-order subgraph as follows:

$$\gamma(v) = \frac{1}{|\mathcal{N}_v^k|} \sum_{v' \in \mathcal{N}_v^k} \frac{\mathbf{e}_v^T \mathbf{e}_{v'}}{\|\mathbf{e}_v\| \|\mathbf{e}_{v'}\|} \quad (2)$$

where The set of k-hop neighbors of node v on the global item transition graph $\mathcal{G}$ is denoted by $\mathcal{N}_v^k$. The embeddings of items v and v' ($v' \in \mathcal{N}_v^k$) are denoted by $\mathbf{e}_v$ and $\mathbf{e}_{v'}$, respectively, where $\mathbf{e}_v, \mathbf{e}_{v'} \in \mathbb{R}^d$. We observe that nodes with higher semantic relatedness scores have higher structural consistency in their neighborhood, implying that the path mask generated from such nodes captures potential cross-sequence item transition patterns while containing less noise. This makes them more beneficial for the SSL reconstruction task.

InfoMax-based SSL Adaptation. To bolster the robustness of the learning-to-mask component for data augmentation, we incorporate Gumbel-distributed noise [17] into the process of determining the semantic relatedness of items. The formal representation is:

$$\gamma'(v) = \gamma(v) - \log(-\log(\mu)), \quad \mu \sim \text{Uniform}(0, 1). \quad (3)$$

The anchor set $\mathcal{V}_a$, with size α ($|\mathcal{V}_{ac}| = \alpha$), is formed by ranking all nodes based on their semantic relatedness and using this ranking to determine their selection probability.

To elevate the adaptability and learnability of the learning-to-mask component for data augmentation, we introduce self-supervised learning signals to the semantic relatedness by maximizing the InfoMax-based semantic relatedness. Formally, we have:

$$\mathcal{L}_{mask} = - \sum_{v \in \mathcal{V}} \gamma'(v) \quad (4)$$

We define the augmented loss objective $\mathcal{L}_{mask}$ to introduce additional self-supervision signals to the learning-to-mask paradigm. The main concept behind the optimized $\mathcal{L}_{mask}$ is to dynamically adjust the semantic relatedness derivation based on the impact of
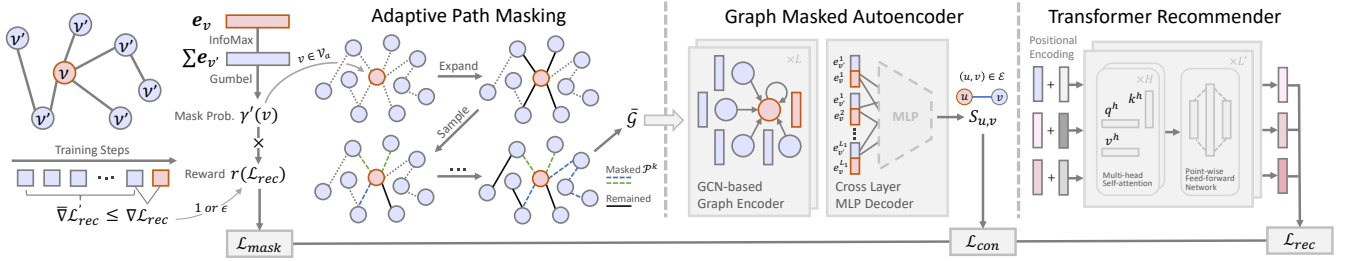
Figure 2: Overall architecture of the proposed GMRec. i) Adaptive path masking for automatically and dynamically distilling information transition patterns for self-supervision. ii) Reconstruction of masked item transitions is performed with a lightweight graph masked autoencoder. iii) Transformer as our backbone for sequence encoding in main recommendation task.

masking on downstream tasks. This automation of the learning-to-mask process enables it to adapt well to changing sequential recommendation scenarios.

3.1.2 **Transition Path Masking.** Once the anchor node set $\mathcal{V}_a$ has been generated using the learning-to-mask module, we propose to mask informative item-item transitional patterns with graph paths as SSL-based reconstruction signals. To achieve this, our transition path masking mechanism should possess two essential properties: (1) The reconstructed item transitional relations should not only contain intra-sequence item associations but also include inter-sequence item dependencies; (2) Diverse sequential patterns from both recent and past interactions should be preserved to reflect short-term and long-term item dependencies.

To achieve the above objectives, our transition path masking mechanism consists of two steps: *Expand* and *Sample*. Specifically, we determine the paths for masking based on a recursive random walk process that contains a succession of item-wise connections over the transition graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Given the k-th order semantic relatedness derivation with respect to representation consistency, our path masking mechanism is formally defined as follows:

$$
\mathcal{P}^k = \begin{cases} \mathcal{V}_a, & \text{if } k = 1 \\ \mathcal{P}^{k-1} \cup \varphi(\mathcal{N}(\mathcal{P}^{k-1}), p^k) & \text{otherwise} \end{cases} \quad (5)
$$

$\mathcal{P}^k$ denotes the set of included item transitional connections. Here, we define $\mathcal{N}(\mathcal{P}^{k-1})$ to represent the set of nodes that are directly connected to a node in $\mathcal{P}^{k-1}$ through an edge in $\mathcal{E}$. To guide our random walk process, we use a drop ratio $0 < p < 1$, which is used in the sample operation $\varphi(\cdot, \cdot)$ that we perform on the elements in a set. In the expand step of our framework, we start a random walk from the identified anchor nodes $\mathcal{V}_a$ to recursively add connected items for sampling transition paths through the function $\varphi(\cdot, \cdot)$. In the drop step, we select the item transition relationships that connect the dropout nodes as transition paths for reconstruction via masked autoencoder. This results in a generated path mask of an anchor node containing sequences of varied length, with a maximum length of $2k$. By doing so, our approach is capable of capturing the complex and dynamic nature of user behaviors. Furthermore, our framework diversifies the position of the anchor node in the sequence, which injects both short-term and long-term self-supervision signals.

3.1.3 **Task-Adaptive Augmentation.** To mitigate the impact of task-irrelevant information and data variance, we have enhanced our augmenter through task adaptation. This allows us to handle a wider variety of user behavior sequences. To achieve this, we introduce a task-adaptive function that guides the learning to mask paradigm, improving the model's generalization ability by masking more informative item transition paths for self-supervision. Otherwise, the reconstruction of task-irrelevant information would weaken the representation power of the self-supervised learning model. We introduce a task-adaptive function denoted by $r(\cdot)$:

$$
r(\mathcal{L}_{rec}) = \begin{cases} 1 & \text{if } \nabla \mathcal{L}_{rec} > \overline{\nabla} \mathcal{L}'_{rec} \\ \epsilon & \text{otherwise} \end{cases} \quad (6)
$$

$\mathcal{L}_{rec}$ represents the loss function of downstream tasks. $\nabla \mathcal{L}_{rec}$ denotes the difference between the current training step and the last step, while $\overline{\nabla} \mathcal{L}'_{rec}$ represents the averaged change of $\mathcal{L}_{rec}$ over the previous $\delta > 1$ steps. The contribution of data-specific masking is measured by the loss benefits brought by the corresponding reconstruction SSL loss. This function enables the learnable masking paradigm to adjust its masking strategy based on the effect of the current mask on the downstream task. We use $\epsilon < 1$ as a small constant. If the self-supervised reconstruction task of the current mask leads to a faster decrease rate of $\mathcal{L}_{rec}$, it corresponds to a smaller mask loss $\mathcal{L}_{mask}$. Otherwise, the mask loss will be greater, and the learning-to-mask module will be guided to be easily adaptable.

Given the estimated task-specific loss reward $r(\mathcal{L}_{rec})$, we enhance the InfoMax-based mask learning in Eq 4 with task-adaptive regularization $\mathcal{L}_{mask} = -r(\mathcal{L}_{rec}) \sum_{v \in \mathcal{V}} \gamma'(v)$. The overall adaptive path masking procedure can be formulated as follows:

$$
\overline{\mathcal{G}} = \text{Path-Mask}(\mathcal{G}, \mathcal{V}_a) = (\mathcal{V}, \mathcal{E} \backslash \mathcal{P}^k) \quad (7)
$$

where $\overline{\mathcal{G}}$ denotes the augmented graph with masked transition paths. $\mathcal{V}_a$ denotes the set of adaptively selected anchor nodes, and $\mathcal{P}^k$ denotes the set of edges in the paths generated from $\mathcal{V}_a$.

3.2 Graph Masked Autoencoder

After obtaining the graph $\overline{\mathcal{G}}$ through adaptive path masking, we feed it into a graph autoencoder framework to perform the reconstruction task for self-supervision. To improve the efficiency of the model, we use a lightweight encoder-decoder framework. In particular, we leverage a simplified graph convolutional network as the encoder for graph embedding, and a cross-layer MLP as the decoder for transition path reconstruction.

3.2.1 GCN-based Graph Encoder.

Inspired by the effectiveness and efficiency of the lightweight graph convolutional network proposed in [4, 14], we omit the heavy transformation and activation during the message passing on $\overline{\mathcal{G}}$ as follows:

$$\mathbf{e}_v^{l+1} = \mathbf{e}_v^l + \sum_{v' \in \mathcal{N}_v} \mathbf{e}_{v'}^l; \quad \tilde{\mathbf{e}}_v = \sum_{l=1}^{L} \mathbf{e}_v^l \qquad (8)$$

where L denotes the total number of layers, and $\mathcal{N}_v$ denotes the 1-hop neighborhood of v. $\mathbf{e}_v^l, \mathbf{e}_{v'}^l \in \mathbb{R}^d$ are the embeddings for $v, v' \in \mathcal{V}$ in the l-th layer. d denotes the embedding dimension of the latent space. In the last layer, the residual connection is added to alleviate the gradient vanishing problem [12]. In specific, we sum up the representations from all the hidden layers to obtain the final embedding $\tilde{\mathbf{e}}_v$ for an item v. With this GCN-based graph encoder, the proposed model can encode the structural information of the masked graph into item embeddings for recommendation as well as reconstruction of the masked edges.

3.2.2 Decoder for Transition Path Reconstruction.

After encoding the item embeddings, the decoder is responsible for reconstructing the missing item transition relationships with masked paths on the augmented $\overline{\mathcal{G}}$. To address the over-smoothing issue of GCNs, we use a cross-layer multi-layer perceptron (MLP) as the decoder. For a masked item-item edge (v, v'), we use the item embeddings of v and v' in each layer of the encoder to construct the edge embedding that corresponds to the item transitional patterns.

$$\mathbf{e}_{v,v'} = \Big\|_{i,j=1}^{L} \mathbf{e}_v^i \odot \mathbf{e}_{v'}^j \qquad (9)$$

The number of layers in the encoder GCN is denoted by L, while the concatenation operation is represented by $\|$ and the element-wise multiplication by $\odot$. Once the edge embedding is constructed, it is passed into the MLP to predict a label (either true or false) for the target item transition path.

3.2.3 SSL-based Reconstruction Objective.

The proposed model differs from conventional graph autoencoders in that it generates self-supervised signals by recovering the masked global item transition paths that are learned via our method. The supplementary objective of this self-supervised reconstruction task can be expressed:

$$\mathcal{L}_{con} = - \sum_{(v,v') \in \mathcal{E} \setminus \mathcal{P}^k} \log \frac{\exp(s_{v,v'})}{\sum_{v'' \in \mathcal{V}} \exp(s_{v,v''})} \qquad (10)$$

The outputs of the MLP corresponding to $\mathbf{e}_{v,v'}$ and $\mathbf{e}_{v,v''}$ are denoted by $s_{v,v'}$ and $s_{v,v''}$. They represent the probability of (v, v') being a masked item-item transition. To accelerate the optimization, negative sampling is employed to sample negative edges (v, v'').

3.3 Transformer as Sequence Encoder

The effectiveness of Transformer in learning sequential behavior patterns in recommender systems, as demonstrated by models like SASRec [19] and Bert4Rec [27], inspired us to use a Transformer for our main supervised sequential recommendation task.

3.3.1 Embedding Layer.

Each item $v \in \mathcal{V}$ is assigned a learnable embedding $\mathbf{e}_v$, which is used for generating the transition path mask during training and for graph encoding. In the embedding layer of our Transformer, a learnable positional embedding $\mathbf{p}_i \in \mathbb{R}^d$ is added to the initial embedding of the item at position i of the input sequence. The initial sequence embedding of an interacted item sequence S^u with length l for user u is then obtained as follows:

$$\mathbf{E}_u = [(\tilde{\mathbf{e}}_{s_1^u} + \mathbf{p}_1), (\tilde{\mathbf{e}}_{s_2^u} + \mathbf{p}_2), \cdots (\tilde{\mathbf{e}}_{s_l^u} + \mathbf{p}_l)] \qquad (11)$$

The generated item representation $\tilde{\mathbf{e}}_{s_l^u}$ is obtained by aggregating information across different graph layers, as presented in Eq 8.

3.3.2 Multi-Head Self-Attention Blocks.

The core component of the Transformer architecture is the multi-head self-attentive mechanism, which can be formulated as follows:

$$\hat{\mathbf{E}}_u^l = \Big\|_{h=1}^{H} \mathbf{A}^h \mathbf{E}_u^l \mathbf{W_V^h}; \quad \mathbf{A}^h = \frac{(\mathbf{E}_u^l \mathbf{W}_Q^h)(\mathbf{E}_u^l \mathbf{W}_K^h)^T}{\sqrt{d/H}}; \qquad (12)$$

$$\mathbf{E}_u^{l+1} = \text{ReLU}(\hat{\mathbf{E}}_u^l \mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2; \quad \tilde{\mathbf{E}}_u = \sum_{l=1}^{L'} \mathbf{E}_u^l \qquad (13)$$

where L' and H denotes the total number of multi-head self-attention blocks and the number of heads, respectively. $\mathbf{A}^h$ denotes the attention values of the h-th head. $\mathbf{W}_Q^h, \mathbf{W}_K^h, \mathbf{W}_V^h \in \mathbb{R}^{d \times (d/H)}$ are the h-th head projection matrices corresponding to query, key, and value in the attention mechanism. The parameters $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^d$ form a point-wise feed forward network. Residual connection is applied to obtain the final sequence embedding $\tilde{\mathbf{E}}_u$. Note that layer normalization and Dropout of attention values as well as inputs of each block are applied to enhance model performance.

3.4 Multi-Task Learning

To predict the probability of a user sequence interacting with item v, we obtain the final embedding of the sequence as the output of the last multi-head self-attention block and compute its dot product with the target item embedding, $\mathbf{e}_v$. We use the cross-entropy loss function to compute the optimized objective for the main supervised recommendation task. Similar to [19, 40], we consider all sub-sequences of each user sequence S^u with length l_u as training data, i.e., $(s_1^u), (s_1^u, s_2^u), \cdots, (s_1^u, \cdots, s_{l_u-1}^u)$. Therefore, the loss function for the recommendation task, $\mathcal{L}_{rec}$, is given by:

$$\mathcal{L}_{rec} = - \sum_{u \in \mathcal{U}} \sum_{1 \le t \le l_u} \log \sigma(\tilde{\mathbf{E}}_{u,t} \cdot \tilde{\mathbf{e}}_{s_u^{t+1}}) + \log(1 - \sigma(\tilde{\mathbf{E}}_{u,t} \cdot \tilde{\mathbf{e}}_{v_t^-})) \quad (14)$$

$\tilde{\mathbf{E}}_{u,t}$ denotes the sequence embedding of $(s_1^u, \cdots, s_t^u)$ and $v_t^- \notin S_u$ is a randomly sampled negative item for t-th interacted item.

To enhance our recommender using SSL augmentation, we incorporate generative SSL signals from a graph masked autoencoder by conducting model training with multi-task learning.

$$\mathcal{L} = \mathcal{L}_{rec} + \mathcal{L}_{mask} + \mathcal{L}_{con} + \lambda \|\Theta\|_F^2 \qquad (15)$$

$\mathcal{L}_{mask}$ is the loss for InfoMax-based SSL adaptation, while $\mathcal{L}_{con}$ denotes the SSL loss of masked item transition reconstruction. Θ denotes the model parameters, and λ is the weight decay coefficient.

3.5 In-Depth Model Discussion

To investigate how the proposed model avoids noisy data augmentation through adaptive transition path masking, we conduct a theoretical analysis on the mask loss of a specific item. Suppose

Table 1: Dataset Statistics

Dataset	#users	#items	#inter.	ave.len.	density
Books	93,403	54,756	506637	5.45	9.94×10^{-5}
Toys	116,429	54,784	478460	4.11	7.50×10^{-5}
Retailrocket	91,655	43,886	452546	4.94	1.12×10^{-4}

that item $v \in \mathcal{V}$ is a noisy item whose related reconstruction could negatively impact representation learning using the provided self-supervision signals. In this case, the gradient of its corresponding mask loss with respect to its embedding $\mathbf{e}_v$ is:

$$\frac{\partial \mathcal{L}_{mask}(v)}{\partial \mathbf{e}_v} = -\frac{r(\mathcal{L}_{rec})}{|\mathcal{N}_v^k|} \sum_{v' \in \mathcal{N}_v^k} \left(\frac{\mathbf{e}_{v'}}{\|\mathbf{e}_v\|\|\mathbf{e}_{v'}\|} - \frac{(\mathbf{e}_v^T \mathbf{e}_{v'})\mathbf{e}_v}{\|\mathbf{e}_v\|^3\|\mathbf{e}_{v'}\|} \right).$$

We further compute the norm of the gradient as follows:

$$\left\| \frac{\partial \mathcal{L}_{mask}(v)}{\partial \mathbf{e}_v} \right\| = -\frac{r(\mathcal{L}_{rec})}{|\mathcal{N}_v^k|\|\mathbf{e}_v\|} \sum_{v' \in \mathcal{N}_v^k} (1 - \bar{\mathbf{e}}_v^T \bar{\mathbf{e}}_{v'}) \qquad (16)$$

In the above equation, $\bar{\mathbf{e}}_v = \mathbf{e}_v/\|\mathbf{e}_v\|$ and $\bar{\mathbf{e}}_{v'} = \mathbf{e}_{v'}/\|\mathbf{e}_{v'}\|$, and the term $\bar{\mathbf{e}}_v^T \bar{\mathbf{e}}_{v'}$ represents the cosine similarity between the two items. Therefore, as the similarity score between v and its neighbor nodes decreases, the sum decreases and the norm increases. This emphasizes the gradient of items with low structural consistency, which improves the transition path masking paradigm with SSL adaptation. As a result, our model can mitigate the influence of behavior noise compared to existing contrastive learning solutions that use handcrafted and random augmentations.

4 EVALUATION

4.1 Experimental Settings

4.1.1 **Datasets.** We conduct experiments on real-world datasets that are widely used in the field of recommendation systems: Amazon Books, Amazon Toys, and Retailrocket. The first two datasets are obtained from the Amazon platform, and we use the version of users' interactions with items in the categories of books and toys. Another benchmark dataset–Retailrocket is collected from an e-commerce website. We follow the similar data pre-processing steps in [35, 40] to generate item sequence of user individual, and summarize data statistics in Table 1.

4.1.2 **Evaluation Protocols.** We adopt the commonly used leave-one-out strategy [24, 27] for sequential recommendation evaluation. Specifically, for each user sequence, we us the last interacted item as the test item. We employ two evaluation metrics: Hit Ratio (*HR@K*) and Normalized Discounted Cumulative Gain (*NDCG@K*) [33, 36] with (K=5,10,20), to evaluate the performance of all algorithms.

4.1.3 **Compared Algorithms.** We compare GMRec with 13 competitive baseline methods for sequential recommendation covering various techniques, which can be grouped into four categories: (1) RNN-based approaches (*i.e.*, GRU4Rec [15], NARM [19]). (2) Transformer-based methods (*i.e.*, SASRec [19], BERT4Rec [27]). (3) GNN-enhenced sequential recommenders (*i.e.*, SRGNN [38], GCE-GNN [35], HyRec [33], and SURGE [2]). (4) Self-supervised learning models (*i.e.*, ContraRec [32], CL4SRec [40], CoSeRec [24], DuoRec [26], and ICLRec [7]). Baselines are elaborated as follows:

- **GRU4Rec** [15]: This method uses GRU to encode user sequences and adopts a ranking based loss for model training.
- **NARM** [21]: It improves GRU-based sequential recommendation models with attention mechanism as hybrid sequence encoders.
- **SASRec** [19]. This is a seminal model that uses a uni-directional Transformer to encode user sequential patterns over items.
- **BERT4Rec** [27]: This model utilizes a bi-directional Transformer as backbone for sequence encoding. The self-attention is integrated with feed-forward network for item transition modeling.
- **SRGNN** [38]: This model uses a gated GNN to learn item embeddings over transition graph, and leverages self-attention mechanism to generate sequence embeddings for making predictions.
- **GCE-GNN** [35]: The proposed model includes two types of item embeddings. The global-level embeddings capture the overall relationships between items across different sequences. Local-level item embeddings encoded by graph convolutional networks.
- **HyRec** [33]: This model utilizes hypergraph convolutional networks to learn high-order correlations between items. It treats each user as a hyperedge to connect multiple interacted items.
- **SURGE** [2]: It is a state-of-the-art model which explicitly learns user interests through clustering and constructing item-item interest graphs based on metric learning. Then, it uses GNNs with graph pooling layers to generate embeddings for prediction.
- **ContraRec** [32]: This model uses the Transformer as backbone for sequence encoding of item transitions. For data augmentation, it proposes a holistic joint-learning paradigm to learn from multiple contrastive self-supervised signals.
- **CL4SRec** [40]: This is a contrastive method that performs random corruptions over sequences using item cropping, masking, and reordering to generate contrastive views. The method then maximizes representation consistency between the generated contrastive views with the stochastic augmentors.
- **CoSeRec** [32]: Apart from the three contrastive views proposed in CL4SRec, this model introduces two additional augmentation operators that leverage item correlations, *i.e.* substitution and insertion, for generating views for contrastive learning.
- **DuoRec** [26]: It introduces a contrastive regularization with model-level augmentation to improve the item embedding distribution and tackle the representation degeneration problem.
- **ICLRec** [7]: This model first learns latent variables to represent user intents through clustering. It then maximizes agreement between sequences and their corresponding intents.

4.1.4 **Hyperparameter Settings.** We implement our proposed GMRec using PyTorch, and use the Adam optimizer for parameter inference with a learning rate of $1e^{-3}$ and a batch size of 256. In our graph autoencoder principle, we set the number of GNN layers in the range of {1,2,3}. The number of Transformer layers is set to 2 with 4 heads for multi-dimensional representation. In the adaptive transition path masking module, we sample $\mathcal{V}_a$ every 10 epochs to improve efficiency. The number of anchor nodes α for each masking step was searched from {50, 100, 200, 400}. The parameter k is searched from {3, 5, 7} to determine the maximum length $2k$ of random walk paths. The weight-decay factor is chosen from {$1e^{-3}, 1e^{-5}, 1e^{-7}$} during model training process.

Table 2: Performance comparison of different methods on Amazon-books, Amazon-toys, and Retailrocket datasets.

Dataset	Metric	GRU4Rec	NARM	SASRec	ContraRec	BERT4Rec	SRGNN	GCE-GNN	HyRec	SURGE	CoSeRec	DuoRec	CL4SRec	ICLRec	Ours	p-val
Books	HR@5	0.3523	0.4064	0.3544	0.4312	0.5385	0.5472	0.4087	0.4550	0.4861	0.5684	0.5941	0.5089	0.5208	**0.6671**	$7e^{-9}$
	NDCG@5	0.2517	0.2955	0.2766	0.3366	0.4404	0.4404	0.3178	0.3454	0.3637	0.4540	0.4894	0.3969	0.4146	**0.5379**	$1e^{-7}$
	HR@10	0.4817	0.5340	0.4393	0.5304	0.6238	0.6458	0.5033	0.5619	0.6193	0.6666	0.6830	0.6144	0.6172	**0.7723**	$1e^{-8}$
	NDCG@10	0.2940	0.3396	0.3058	0.3685	0.4681	0.4723	0.3484	0.3800	0.4068	0.4858	0.5182	0.4310	0.4458	**0.5720**	$1e^{-7}$
	HR@20	0.6268	0.6725	0.5407	0.6440	0.7121	0.7503	0.6090	0.6778	0.7604	0.7622	0.7706	0.7216	0.7141	**0.8626**	$2e^{-7}$
	NDCG@20	0.3306	0.3737	0.3298	0.3966	0.4904	0.4987	0.3751	0.4092	0.4424	0.5100	0.5403	0.4581	0.4703	**0.5950**	$2e^{-7}$
Toys	HR@5	0.2247	0.2047	0.2454	0.2989	0.3717	0.3610	0.2597	0.2978	0.3075	0.3869	0.4412	0.3451	0.3483	**0.4796**	$4e^{-6}$
	NDCG@5	0.1543	0.1383	0.1854	0.2318	0.2885	0.2720	0.1880	0.2149	0.2231	0.2939	0.3450	0.2696	0.2655	**0.3679**	$3e^{-3}$
	HR@10	0.3262	0.3029	0.3299	0.3805	0.4665	0.4674	0.3628	0.4030	0.4210	0.4925	0.5424	0.4417	0.4416	**0.5840**	$7e^{-4}$
	NDCG@10	0.1888	0.1718	0.2155	0.2579	0.3191	0.3064	0.2211	0.2488	0.2596	0.3280	0.3777	0.3007	0.2971	**0.4017**	$1e^{-3}$
	HR@20	0.4700	0.5012	0.4333	0.4862	0.5780	0.5940	0.4952	0.5251	0.5752	0.6135	0.6583	0.5638	0.5652	**0.6901**	$8e^{-2}$
	NDCG@20	0.2240	0.2470	0.2413	0.2843	0.3472	0.3383	0.2545	0.2797	0.2984	0.3585	0.4070	0.3315	0.3271	**0.4285**	$3e^{-2}$
Retail	HR@5	0.4164	0.5586	0.6136	0.6587	0.7596	0.6973	0.4238	0.4801	0.7091	0.8063	0.7873	0.6663	0.7818	**0.8530**	$2e^{-10}$
	NDCG@5	0.2950	0.4238	0.5077	0.5553	0.6882	0.5974	0.3371	0.3792	0.5902	0.7117	0.7128	0.5752	0.6904	**0.7531**	$4e^{-8}$
	HR@10	0.4518	0.6757	0.6968	0.7387	0.8084	0.7719	0.5182	0.5722	0.7857	0.8530	0.8291	0.7336	0.8282	**0.8921**	$3e^{-8}$
	NDCG@10	0.3456	0.4600	0.5389	0.5823	0.7039	0.6216	0.3676	0.4091	0.6151	0.7269	0.7264	0.5969	0.7055	**0.7659**	$1e^{-7}$
	HR@20	0.6095	0.7879	0.7687	0.8140	0.8577	0.8425	0.6275	0.6361	0.8581	0.8951	0.8703	0.8018	0.8733	**0.9178**	$1e^{-4}$
	NDCG@20	0.3830	0.4972	0.5558	0.6009	0.7164	0.6394	0.3951	0.4256	0.6334	0.7376	0.7368	0.6142	0.7169	**0.7725**	$1e^{-6}$

4.2 Performance Comparison

Table 2 presents the performance of all methods on three benchmark datasets in terms of *HR@K* and *NDCG@K* under top-5, top-10, and top-20 settings. Additionally, we perform a significance test to demonstrate the superiority of GMRec over the strongest baselines, where a p-value < 0.05 indicates statistically significant improvement. The following observations can be made from the results:

- **Obs.1: Superiority over SOTA GNN-based models**. GMRec consistently outperforms all GNN-based baselines across different settings. Existing GNN-based models may suffer from the over-fitting issue over short sequences with insufficient training labels. Additionally, SSL-enhanced sequential recommenders perform better than GNN-based baselines, indicating the helpfulness of incorporating self-supervision signals to augment sequence encoding with limited labels. To mitigate the data scarcity problem, our GMRec incorporates generative self-supervision signals with graph masked autoencoder for effective data augmentation. As a result, informative as well as robust representations can be learned to preserve global item transition patterns.

- **Obs.2: Superiority over SOTA SSL methods**. GMRec achieves the best overall performance in comparison to all baselines with self-supervised augmentation. While these methods generally perform better than GNN-based models by introducing self-supervision signals, their random data augmentation still leads to sub-optimal performances. This is because random augmentation strategies (*e.g.*item cropping or reordering) used in common SSL models are performed over all user sequences and all items, making the model unstable to noise perturbations on both sequence and item levels, such as interaction noise and popularity bias. Instead of random augmentations, GMRec adopts a learnable strategy for introducing useful self-supervision signal by masking informative transition paths generated from adaptively selected items. As a result, the newly designed GMRec suppresses existing SSL baselines by a large margin with an adaptive graph masked autoencoder as a robust data augmentor.

- **Obs.3: Advantage of our mask-based data augmentation**. There are several baseline models that also utilize the mask operator for self-supervised learning, such as CL4SRec and CoSeRec, which mask items in a sequence to produce contrastive views. Additionally, BERT4Rec incorporates the Cloze objective into the sequence encoder for item embeddings. Benefitting from such mask-based data augmentation, BERT4Rec outperforms SASRec, which only uses a single-directional Transformer for sequence encoding. Meanwhile, CoSeRec and CL4SRec both achieve better results than many other baselines due to their data augmentation. However, our GMRec is the only model that automates the mask-and-reconstruct paradigm over item transitional patterns with effective task-adaptation, resulting in even better performance.

4.3 Ablation Study

This section presents the ablation study conducted on three variants of the proposed GMRec algorithm: **-L2M**, **-PA**, and **-TA**. The purpose of the study was to investigate the contribution of different components to the overall performance of the algorithm. The results of the study are reported in Table 3.

4.3.1 **Effect of the Mask Generator.** The first variant, **-L2M**, replaces the our designed mask generator with random masking of the same proportion. The results indicate that non-adaptive masking may harm important transition relations or introduce noisy information to the masked-based reconstruction task, which weakens the representations and leads to suboptimal performance.

4.3.2 **Effect of Transition Path Masking.** The second variant, **-PA**, studies the benefits of masking transition paths instead of single nodes. In this variant, the maximum length of a masked path is set to $k = 1$, which makes path masking equivalent to single node masking. The results demonstrate that the proposed transition path masking significantly improves the overall performance, highlighting the effectiveness of learning to mask item transition patterns from both intra-sequence and inter-sequence perspectives.

Table 3: Model ablation study with variants.

Method	Amazon books		Retailrocket	
	HR@10	NDCG@10	HR@10	NDCG@10
-L2M	0.7482	0.5510	0.8800	0.7547
-PA	0.7272	0.5239	0.8590	0.7056
-TA	0.7507	0.5542	0.8801	0.7564
Ours	**0.7723**	**0.5720**	**0.8921**	**0.7659**
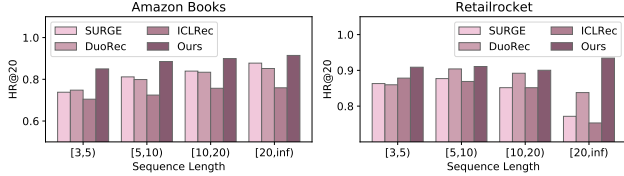
Figure 3: Performance *w.r.t* sequence length on Amazon Books and Retailrocket datasets in terms of *HR@20*.
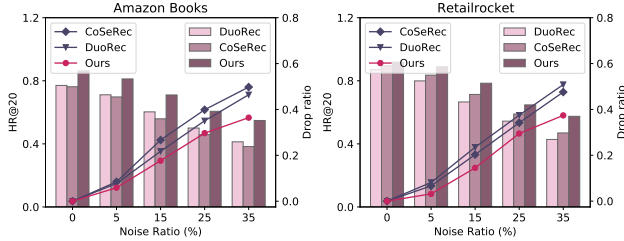
Figure 4: Performance *w.r.t* noise ratio on Amazon Books and Retailrocket. The bars show performance in terms of *HR@20*, and the lines show performance degradation ratio.

4.3.3 **Effect of Task-Adaptive Augmentation.** The adaptive masking loss with the task-adaptive function $r(\cdot)$ is also a crucial component of our GMRec algorithm. In order to study its impact, we remove the loss for learnable masking $\mathcal{L}_{mask}$ (Eq 4) in the variant **-TA**, disabling the task-adaptive regularization. The results show a significant drop in performance, indicating that the task-adaptive function guides the model training in a better direction according to the gradient from the target recommendation task. With the task-adaptive constraints, our GMRec algorithm can effectively prevent the data augmentor from learning task-irrelevant information by selectively applying self-supervision augmentation.

4.4 Advantage against Noise and Sparsity

4.4.1 **Performance against data sparsity.** The data sparsity issue is a common problem for sequential recommender systems with a large number of short sequences. To evaluate the robustness of our proposed GMRec algorithm against this problem, we partition the user sequences into four groups based on the length of their item sequences: $[3, 5)$, $[5, 10)$, $[10, 20)$, $[20, \infty)$. Figure 3 shows the performance of our model and baseline methods on both Amazon Books and Retailrocket datasets. The results demonstrate that our proposed model consistently outperforms the baseline methods at all sparsity levels. Specifically, our model performs well on short sequences, which can be beneficial for addressing the label insufficiency limitation in practical recommendation scenarios. This can be attributed to the SSL-enhanced transition path masking module,
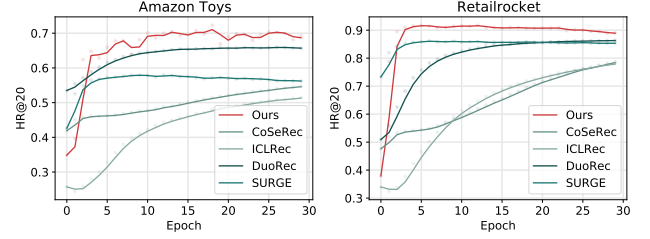
Figure 5: Test performance *w.r.t.* training epochs on Amazon Books and Retailrocket datasets for convergence analysis.

which preserves both local and global item dependencies in an automatic and adaptive manner. The handcrafted augmentation used in DuoRec and ICLRec may limit their representation generality in encoding accurate user preferences from short sequences, due to the diverse nature of interaction data. In contrast, our GMRec algorithm can effectively transfer knowledge from long item sequences into the modeling of short sequences, allowing it to learn meaningful representations from sparse data with limited labels.

4.4.2 **Performance against data noise.** Real-world recommendation systems often face noisy user interactions. To evaluate the robustness of our proposed GMRec algorithm against noisy data, we replace a certain proportion (*i.e.*, 5%, 15%, 25%, and 35%) of user interactions with randomly sampled negative items, and evaluate the models on datasets with artificial noise perturbations. As shown in Figure 4, our proposed model outperforms the baselines under all noise levels on both Amazon Books and Retailrocket datasets. Moreover, our model exhibits a consistently lower performance degradation ratio compared to the baselines.

We attribute this superiority to the adaptive transition path masking module in our algorithm. Firstly, the adaptive masking strategy avoids masking edges connected to noisy nodes using the InfoMax-based mask probability and mask loss, encouraging the model to mask and learn on nodes with reliable dependency relationships. This significantly increases the model's robustness against noisy perturbations. Secondly, the transition paths in our algorithm are generated in the local subgraph of the anchor nodes, resulting in higher semantic relatedness among items in such sequences compared to randomly generated augmented sequences, such as those generated by random masking, dropping, and reordering. As a result, it can be observed in Figure 4 that CoSeRec is relatively unstable against data noise, indicating the disadvantage of random augmentation methods compared to adaptive path masking.

4.5 Model Scalability Study

We evaluate the efficiency of our proposed model from two aspects: convergence speed and training time cost. All experiments are conducted on a single NVIDIA GeForce RTX 3090 GPU.

4.5.1 **Convergence Analysis.** To study the convergence efficiency of our proposed model, we compare its training process with several baselines, including both GCN-based methods (SURGE) and SSL-based methods (DuoRec, CoSeRec, ICLRec), on the Amazon Toys and Retailrocket datasets. The training process of these models is presented in Figure 5. From the results, we observe that our proposed model achieves the best performance with the least training epochs. It reaches the best reported performance at around 10

Table 4: Training time cost on Amazon toys and Retailrocket.

Data	SURGE	ICLRec	CoSeRec	DuoRec	Ours
Amazon	7.39h	5.28h	21.01h	3.89h	1.12h
Retail	2.67h	4.3h	16.18h	4.45h	1.82h

epochs on Amazon Toys and around 5 epochs on Retailrocket, while CoSeRec takes more than 200 epochs to converge. We attribute this superiority to the self-supervised learning task enhanced by graph masked autoencoder, which injects sufficient supervision signals for the model to learn informative item transition patterns on the masked item transition graph in an efficient way.

4.5.2 **Computational Cost Evaluation.** To study the training computational cost of our proposed model, we also record the training time of our model and the baselines, as shown in Table 4. The results show that our proposed model has a significant efficiency improvement over compared methods, which suggests that our model is a lightweight and strong recommender, enabling automated and task-adaptive masked autoencoding. Our model is supercharged with adaptive SSL augmentation, and thus requires much less training time compared to other SSL-based methods, which use non-adaptive (heuristic-based) augmentation operators for downstream recommendation tasks.
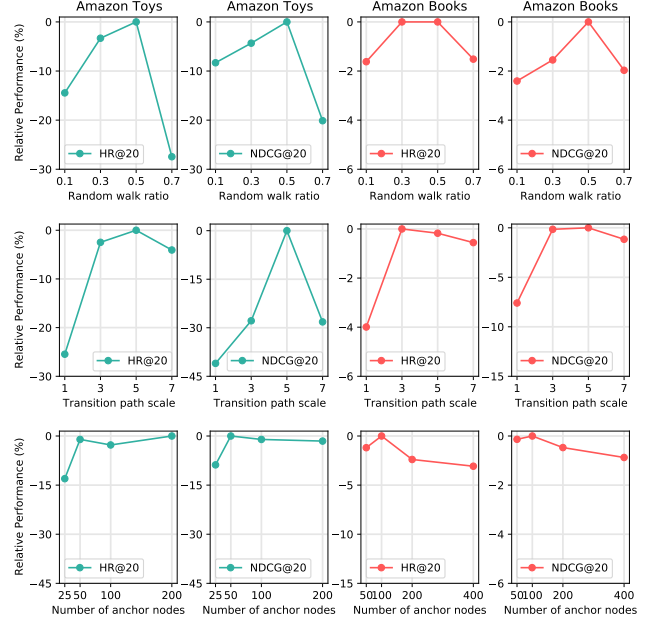
4.6 Hyperparameter Study

We conduct experiments to investigate the effect of several key hyperparameters on the performance of our proposed model. The experiment results are presented in Figure 6, in terms of the relative decrease of HR@20 and NDCG@20 on the Amazon Books and Amazon Toys datasets. The observations are shown below:

- **Random walk ratio p over item transitions**. This hyperparameter controls the number of masked edges triggered by the identified anchor node and affects the length of masked paths, which reflect item transitional patterns. Specifically, a larger value of p indicates a longer masked item transition path. Our experiments show that, $p = 0.5$ achieves the best performance. This indicates that masking transition paths with variable lengths benefits the model learning with more diverse SSL signals.

- **Masked transition path scale k**. This hyperparameter controls the maximum length $2k$ of the generated transition paths. Our experiments show that setting $k = 1$ (*i.e.*, only masking edges in the 1-hop of anchor nodes) results in significant performance degradation. Interestingly, we also observe that masking too long paths can lead to a performance drop, as insufficient information is provided for achieving accurate reconstruction results. This may impair the quality of the auxiliary SSL task for augmenting the target recommendation task.

- **Number of anchor nodes α.** α determines the number of seed nodes for transition path masking. Experiments show that masking transition paths from too few anchor nodes leads to worse performance. On the other hand, selecting too many anchor nodes for path masking may damage the structure of the transition graph, and hence negatively affect model performance.

4.7 Case Study

We perform a case study on the Retailrocket dataset to investigate the potential interpretation ability of our proposed model. Specifically, we select two items with high and low semantic relatedness,

Figure 6: Hyperparameter study of key parameters in GMRec.

Neighbors of v_{10919}, $\gamma(v_{10919}) = 0.96$ Neighbors of v_{5887}, $\gamma(v_{10919}) = -0.72$
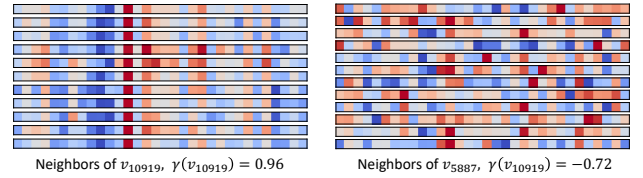
Figure 7: Learned semantic relatedness reflects representation consistency among items connected in transition graph.

respectively, and visualize the embeddings of their neighbor items on the item transition graph in Figure 7. The results suggest that our learning-to-mask module is able to discover items with reliable dependency relationships to be anchor nodes, which benefits the reconstruction task. In contrast, items with low semantic relatedness (*e.g.*, misclicked items) have inconsistent neighborhoods in terms of their embeddings. Masking transition paths from such items may result in noisy data augmentation and misguide the self-supervision process for sequential pattern encoding.

5 CONCLUSION

This work proposes an adaptive data augmentation approach to enhance sequential recommender systems through a new graph masked autoencoder. Our proposed GMRec provides controllable and task-adaptive augmentation with strong self-supervision signals, thanks to the designed adaptive path masking. We conduct extensive experiments on three real-world datasets and demonstrat the superior performance of our GMRec compared with state-of-the-art baselines. In future work, we plan to improve the stability of SSL training by generalizing our approach to out-of-distribution sequences. This will help to address the data distribution shift between the training and test data in sequential recommendation, and generalize the model to newly arrived item sequences in future.

REFERENCES

[1] Xuheng Cai, Chao Huang, Lianghao Xia, and Xubin Ren. 2023. LightGCL: Simple Yet Effective Graph Contrastive Learning for Recommendation. In *International Conference on Learning Representations (ICLR)*.

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

[3] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2020. Bias and debias in recommender system: A survey and future directions. *Transactions on Information Systems (TOIS)* (2020).

[4] Lei Chen, Le Wu, Richang Hong, Kun Zhang, and Meng Wang. 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach. In *AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 34. 27–34.

[5] Tianwen Chen and Raymond Chi-Wing Wong. 2020. Handling information loss of graph neural networks for session-based recommendation. In *International Conference on Knowledge Discovery and Data Mining (KDD)*. 1172–1180.

[6] Yabo Chen, Yuchen Liu, Dongsheng Jiang, Xiaopeng Zhang, Wenrui Dai, Hongkai Xiong, and Qi Tian. 2022. Sdae: Self-distillated masked autoencoder. In *European Conference on Computer Vision (ECCV)*. Springer, 108–124.

[7] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent Contrastive Learning for Sequential Recommendation. In *ACM Web Conference (WWW)*. 2172–2182.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *North American Chapter of ACL (NAACL)*.

[9] Xinyan Fan, Zheng Liu, Jianxun Lian, Wayne Xin Zhao, Xing Xie, and Ji-Rong Wen. 2021. Lighter and better: low-rank decomposed self-attention networks for next-item recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 1733–1737.

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

[11] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 16000–16009.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778.

[13] Sunan He, Taian Guo, Tao Dai, Ruizhi Qiao, Chen Wu, Xiujun Shu, and Bo Ren. 2022. VLMAE: Vision-Language Masked Autoencoder. *arXiv preprint arXiv:2208.09374* (2022).

[14] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 639–648.

[15] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations (ICLR)*.

[16] Zhenyu Hou, Xiao Liu, Yuxiao Dong, Chunjie Wang, Jie Tang, et al. 2022. Graph-MAE: Self-Supervised Masked Graph Autoencoders. In *International Conference on Knowledge Discovery and Data Mining (KDD)*.

[17] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparametrization with Gumble-Softmax. In *International Conference on Learning Representations (ICLR)*.

[18] Baoyu Jing, Chanyoung Park, and Hanghang Tong. 2021. Hdmi: High-order deep multiplex infomax. In *The Web Conference (WWW)*. 2414–2424.

[19] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[20] Yejin Kim, Kwangseob Kim, Chanyoung Park, and Hwanjo Yu. 2019. Sequential and Diverse Recommendation with Long Tail.. In *International Joint Conference on Artificial Intelligence (IJCAI)*, Vol. 19. 2740–2746.

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

[22] Guanyu Lin, Chen Gao, Yinfeng Li, Yu Zheng, Zhiheng Li, Depeng Jin, and Yong Li. 2022. Dual contrastive network for sequential recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 2686–2691.

[23] Siyi Liu and Yujia Zheng. 2020. Long-tail session-based recommendation. In *ACM Conference on Recommender Systems (Recsys)*. 509–514.

[24] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S Yu, Julian McAuley, and Caiming Xiong. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479* (2021).

[25] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory augmented graph neural networks for sequential recommendation. In *AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 34. 5045–5052.

[26] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *International Conference on Web Search and Data Mining (WSDM)*. 813–823.

[27] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *International Conference on Information and Knowledge Management (CIKM)*. 1441–1450.

[28] Qiaoyu Tan, Ninghao Liu, Xiao Huang, Rui Chen, Soo-Hyun Choi, and Xia Hu. 2023. S2GAE: Self-Supervised Graph Autoencoders Are Generalizable Learners with Graph Masking. In *International Conference on Web Search and Data Mining (WSDM)*.

[29] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *International Conference on Web Search and Data Mining (WSDM)*. 565–573.

[30] Changxin Tian, Yuexiang Xie, Yaliang Li, Nan Yang, and Wayne Xin Zhao. 2022. Learning to Denoise Unreliable Interactions for Graph Collaborative Filtering. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 122–132.

[31] Petar Velickovic, William Fedus, William L Hamilton, Pietro Liò, Yoshua Bengio, and R Devon Hjelm. 2019. Deep Graph Infomax. In *International Conference on Learning Representations (ICLR)*.

[32] Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. Sequential recommendation with multiple contrast signals. *Transactions on Information Systems (TOIS)* 41, 1 (2023), 1–27.

[33] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 1101–1110.

[34] Jianling Wang, Raphael Louca, Diane Hu, Caitlin Cellier, James Caverlee, and Liangjie Hong. 2020. Time to Shop for Valentine's Day: Shopping Occasions and Sequential Recommendation in E-commerce. In *International Conference on Web Search and Data Mining (WSDM)*. 645–653.

[35] Ziyang Wang, Wei Wei, Gao Cong, Xiao-Li Li, Xian-Ling Mao, and Minghui Qiu. 2020. Global context enhanced graph neural networks for session-based recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 169–178.

[36] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive meta learning with behavior multiplicity for recommendation. In *International Conference on Web Search and Data Mining (WSDM)*. 1120–1128.

[37] Yinwei Wei, Xiang Wang, Liqiang Nie, Xiangnan He, Richang Hong, and Tat-Seng Chua. 2019. MMGCN: Multi-modal graph convolution network for personalized recommendation of micro-video. In *International Conference on Multimedia (MM)*. 1437–1445.

[38] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *AAAI Conference on Artificial Intelligence (AAAI)*, Vol. 33. 346–353.

[39] Lianghao Xia, Chao Huang, and Chuxu Zhang. 2022. Self-supervised hypergraph transformer for recommender systems. In *International Conference on Knowledge Discovery and Data Mining (KDD)*. 2100–2109.

[40] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Bolin Ding, and Bin Cui. 2021. Contrastive learning for sequential recommendation. In *International Conference on Data Engineering (ICDE)*.

[41] An Yan, Shuo Cheng, Wang-Cheng Kang, Mengting Wan, and Julian McAuley. 2019. CosRec: 2D convolutional neural networks for sequential recommendation. In *International Conference on Information and Knowledge Management (CIKM)*. 2173–2176.

[42] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. 2023. Debiased Contrastive Learning for Sequential Recommendation. In *ACM Web Conference (WWW)*. 1063–1073.

[43] Yuhao Yang, Chao Huang, Lianghao Xia, Yuxuan Liang, Yanwei Yu, and Chenliang Li. 2022. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *International Conference on Knowledge Discovery and Data Mining (KDD)*. 2263–2274.

[44] Wenwen Ye, Shuaiqiang Wang, Xu Chen, Xuepeng Wang, Zheng Qin, and Dawei Yin. 2020. Time matters: Sequential recommendation with complex temporal information. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 1459–1468.

[45] Shengyu Zhang, Dong Yao, Zhou Zhao, Tat-Seng Chua, and Fei Wu. 2021. Causerec: Counterfactual user sequence synthesis for sequential recommendation. In *International Conference on Research and Development in Information Retrieval (SIGIR)*. 367–377.

[46] Yixin Zhang, Yong Liu, Yonghui Xu, Hao Xiong, Chenyi Lei, Wei He, Lizhen Cui, and Chunyan Miao. 2022. Enhancing Sequential Recommendation with Graph Contrastive Learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

[47] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2021. Graph contrastive learning with adaptive augmentation. In *ACM Web Conference (WWW)*. 2069–2080.