Improving Sequential Recommendation Consistency with Self-Supervised Imitation

Xu Yuan\textsuperscript{1,2,3}, Hongshen Chen\textsuperscript{3}, Yonghao Song\textsuperscript{1}, Xiaofang Zhao\textsuperscript{1} and Zhuoye Ding\textsuperscript{3},

\textsuperscript{1}Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
\textsuperscript{2}University of Chinese Academy of Sciences, Beijing, China
\textsuperscript{3}JD.com, China

\{yuanxu19g, songyonghao\}@ict.ac.cn, ac@chenhongshen.com

Abstract

Most sequential recommendation models capture the features of consecutive items in a user-item interaction history. Though effective, their representation expressiveness is still hindered by the sparse learning signals. As a result, the sequential recommender is prone to make inconsistent predictions. In this paper, we propose a model, SSI, to improve sequential recommendation consistency with Self-Supervised Imitation. Precisely, we extract the consistency knowledge by utilizing three self-supervised pre-training tasks, where temporal consistency and persona consistency capture user-interaction dynamics in terms of the chronological order and persona sensitivities, respectively. Furthermore, to provide the model with a global perspective, global session consistency is introduced by maximizing the mutual information among global and local interaction sequences. Finally, to comprehensively take advantage of all three independent aspects of consistency-enhanced knowledge, we establish an integrated imitation learning framework. The consistency knowledge is effectively internalized and transferred to the student model by imitating the conventional prediction logit as well as the consistency-enhanced item representations. In addition, the flexible self-supervised imitation framework can also benefit other student recommenders. Experiments on four real-world datasets show that SSI effectively outperforms the state-of-the-art sequential recommendation methods.

1 Introduction

With the development of mobile devices and Internet services, the past few years have seen the prosperity of a broad spectrum of recommendation systems (RS). It facilitates individuals making decisions over innumerable choices on the web. RS attracts a growing number of online retailers and e-commerce platforms to meet diverse user demands, enrich and promote their online shopping experiences.

In real-world applications, users’ current interests are affected by their historical behaviors. When a user orders a smartphone, he/she will subsequently choose and purchase accessories like chargers, mobile phone covers, etc. Such sequential user-item dependencies prevail and motivate the rising of sequential recommendation, where the user history interaction sequence is treated as a dynamic sequence. And the sequential dependencies are taken into account to characterize the current user preference to make more accurate recommendations\cite{Chen2018}.

As for the sequential recommendation, a surge of methods have been proposed to capture the sequential dynamics within user historical behaviors and predict the next use-interested item(s): Markov Chains\cite{Zimdars2001} recurrent neural networks (RNNs)\cite{Hidasi2016}, convolutional neural networks (CNNs)\cite{Tang2018; Yuan2019}, Graph Neural Network (GNN)\cite{Wu2019}, self-attention mechanisms\cite{Ying2018}. Markov chain-based models adopt K-order user-item interaction sequential transitions. CNN handles the transition within a sliding window, whereas RNN-based methods apply GRU or LSTM to compress dynamic user interests with hidden states. GNN-based methods take advantage of directed graphs to model complex user-item transitions in structured relation datasets. And self-attention mechanisms emphasize relevant and essential interactions with different historical user-item interaction weights.

Though previous methods show promising results, current sequential recommendation system learning relies heavily on the observed user-item action prediction. As a result, their representations and expressiveness are still limited. Such representation learning signal is too sparse to train expressive-enough item representations\cite{Rendle2010; Song2019}. Not to mention that those subtle but diverse persona differences among users are also neglected. In this paper, we attempt to abstract the consistency knowledge with self-supervised pre-training tasks by exploring three aspects of consistency. To handle user-interaction dynamics, we propose to abstract consistency information with temporal and persona consistency, where temporal consistency reflects that the recommender is expected to organize and display the items in a proper order to satisfy users’ interests. Persona consistency stands on the fact that the recommender should be capable of perceiving those diverse persona distinc-
tions from the user-item interaction sequence.

Temporal and persona consistency knowledge is quite a straightforward approach to improve sequential recommendation systems by modeling instant interaction dynamics. However, long-term global session consistency is overlooked. Without a global perspective, the model still suffers from noisy actions and makes inconsistent predictions. A typical case is that when a user unconsciously clicks wrong items, the system will be easily affected by short-term click and instantly make unrelated predictions. To mitigate such defect, we further bring global session consistency into the sequential recommendation, which enhances the item representations by maximizing the mutual information between global and local parts of the interaction sequence.

Though self-supervised consistency knowledge can be quite beneficial, directly optimizing the consistency is intractable. It’s because training is performed on the whole session, which is not attainable in inference. Furthermore, comprehensively integrating all three aspects of consistency pre-training knowledge plays a vital role in estimating user’s preferences more accurately and consistently. Therefore, we propose an integrated imitation learning framework. Consistency-enhanced pre-training models serve as teachers, and a sequential recommendation model is treated as a student. The consistency knowledge is effectively internalized and transferred to the student model by imitating the conventional prediction logit as well as the consistency-enhanced item representations. Additionally, with the flexibility of Self-Supervised Imitation (SSI) framework, the self-supervised learning can be easily transferred to other student recommenders as demand.

Experiments on four real-world datasets show that our self-supervised imitation framework effectively outperforms several state-of-the-art baselines.

2 Methodology

2.1 Problem Statement

Suppose \( I \) is the set of items and \( U \) is the set of users. For a specific user \( u \in U \), list \( I_u = \{q_1^{(u)}, q_2^{(u)}, ..., q_k^{(u)}\} \) denote the corresponding interaction sequence in chronological order where \( q_k \in I \) and \( k \) is the time step. Given the interaction history \( I_u \), sequential recommendation aims to predict the item that user \( u \) will interact with at time step \( k + 1 \).

2.2 Teacher Base Model

Our teacher base model is a BERT-based structure[Sun et al., 2019], built upon the popular self-attention layer. The model randomly masks items in the user history interaction sequence in the training phase and replaces them with a unique token [MASK]. Then predicts the original ids of the masked items based on the context. In the test phase, the model appends the special token [MASK] at the end of the input sequence and then predicts the next item based on the final hidden representation of this token.

2.3 Self-supervised Consistency Pre-training

Existing studies[Hidasi et al., 2016b; Hidasi et al., 2016a; Kang and McAuley, 2018; Logeswaran and Lee, 2018] mainly emphasize the effect of sequential characteristics using user-item action prediction objective alone. Solely relying on such sparse learning signals may result in underlearned item representations. We enrich learning representations’ expressiveness by extracting consistency knowledge with self-supervised learning objectives to mitigate such defects. As exemplified in Figure 1, we introduce three aspects of recommendation consistency: temporal, persona and global session consistency, respectively, where temporal consistency and persona consistency capture user-interaction dynamics and global session consistency enhances the model with the global perspective.

**Temporal Consistency**

Temporal consistency captures users’ interaction sequence so that the recommendation system can display the items in a proper order that will satisfy users’ interests better. Therefore, we design a user-item interaction sequence chronological order recognition task to enhance the model temporal sensitivity. First, we draw positive and negative samples by randomly exchanging the order of some n-gram sub-sequences in the user’s interaction history with Bernoulli distribution. Then we add a unique token named [INT] at the end of the sequence and let the corresponding hidden representation go through a classifier to predict whether the interaction sequence is in the original order or not. The loss of temporal consistency is formulated as:

\[
    L_{\text{temp}} = \sum_{i \in D} -y'_i \log h'(x'_i) - (1 - y'_i) \log(1 - h'(x'_i)),
\]

where \( D \) is the set of input, \( x'_i \) is the output embedding, \( y'_i \) is label, \( L_{\text{temp}} \) is calculated by cross-entropy, and \( h'(\cdot) \) is a MLP.
Persona Consistency

Person consistency models the diverse persona distinctions among different users. Positive samples are the interactions from one user. In contrast, for negative samples, we replace some user-item interaction n-grams with other items. Similarly, we add a unique token [INT] at the end of the sequence, and a classifier predicts whether the user will interact with the current sequence. Persona consistency knowledge is perceived by differentiating whether the given user-item sequence is from one user. The loss of persona consistency is formulated as:

$$L_{pers} = \sum_{i \in D} -y_i^p \log h^p(x_i^p) - (1 - y_i^p) \log(1 - h^p(x_i^p)),$$  

(2)

where $D$ is the set of input, $x_i^p$ is the output embedding, $y_i^p$ is label, $L_{pers}$ is calculated by cross-entropy, and $h^p(\cdot)$ is a MLP.

Global Session Consistency

Researches on image representation learning show that mutual information maximization between an image representation and local regions improves the quality of the representation [Hjelm et al., 2019]. Similar observations are also found in sentence representation learning [Kong et al., 2019]. Inspired by previous studies, to avoid noise and provide the model with a global perspective, we further introduce the global session consistency as a self-supervision task for the sequential recommendation. Given a user interaction sequence $I = \{q_1, q_2, ..., q_t\}$, we consider the local representation to be the encoded representations of n-grams in the sequence. The global representation is the rest of the masked sequence representation, which corresponds to the last token hidden state. We maximize the mutual information between the global representation and local representation. Denote an n-gram by $q_{am}$ and a masked sequence at position $n$ to $m$ by $\hat{q}_{nm}$. We define $L_{global}$ as:

$$L_{global} = -E_{p}(\hat{q}_{nm}, q_{nm})d(\hat{q}_{nm})^\top d(q_{nm})$$

$$- \log \sum_{\hat{q}_{nm} \in \hat{S}} \exp(d(\hat{q}_{nm})^\top d(q_{nm}))),$$  

(3)

where $\hat{q}_{nm}$ is an n-gram from a set $\hat{S}$ that consists of the positive sample $q_{nm}$ and negative n-grams from other sequence in the same batch, $d(\cdot)$ is the base model.

Consistency Pre-Training

To ensure that each aspect of consistency is well-learned, we establish three independent models upon BERT4Rec [Sun et al., 2019] to induce temporal consistency, persona consistency, and global consistency information. The training objectives are defined as:

$$L^1_t = L_{MLM} + \lambda_1 L_{temp},$$

$$L^2_t = L_{MLM} + \lambda_2 L_{pers},$$

$$L^3_t = L_{MLM} + \lambda_3 L_{global},$$  

(4)

where $L_{MLM}$ is the loss of the base model, $L^t_1$ is the loss of each model. $L_{temp}$, $L_{pers}$ and $L_{global}$ are augmented as a kind of regularization to abstract the consistency knowledge from the sequence. $\lambda_i$ is a hyper-parameter balancing the importance of the two losses.

2.4 Self-supervised Imitation Learning

So far, three aspects of consistency knowledge acquired from the proposed self-supervised learning tasks. However, sequential recommendation system still faces two major roadblocks in augmenting them: 1) in our approach, consistency learning is conducted on the entire user-item interaction session, whereas only the historical user behaviors are available during inference and future interactions are unattainable; 2) given three aspects of consistency-enhanced representations, it is paramount to explore a comprehensive integration of them to estimate a user’s preference with better accuracy and consistency.

We propose an imitation learning framework to overcome the obstacles, where we treat the consistency-enhanced models as teachers and another sequential recommendation model as a student. The student is designated to distill consistency knowledge from three self-supervision-enhanced teachers by performing both the prediction distribution imitation and item representation imitation. Figure 2 illustrates the general self-supervised imitation learning framework.

Prediction Distribution Imitation

In prediction distribution imitation, the student model distills the merits from the teacher model with prediction behaviors. In particular, we minimize the KL divergence of the predicted probability distribution between the teacher and the student:

$$L^1_{KL}(\theta_1, \theta_2) = D_{KL}(p(q = Q(\theta_1)) | p(q = Q(\theta_2))).$$  

(5)

where $Q$ is the set of candidate items, $\theta_1$ and $\theta_2$ are the parameters of the teacher and student model, respectively.

Item Representation Imitation

Item representation imitation learns from the teacher by imitating the consistency-enhanced representations. In detail, we restrict the item representations discrepancy between the student and the teacher:

$$L^2_{KL}(\theta_1, \theta_2) = f(g^b(\theta_1), g^s(\theta_2)),$$  

(6)

$g^b(\cdot)$ and $g^s(\cdot)$ are the base model and student model.
where $g'$ and $g^t$ are the outputs of item representations in teacher and student respectively. $f(\cdot)$ is the mean-squared-error (MSE) loss.

### Integrated Imitation Training

To effectively integrate three independent consistency teachers with the student, given the loss $L^i_{it}$ in Equation 5 and the loss $L^j_{it}$ in Equation 6, the final objective function of the student is formulated as:

$$L' = L_i + \sum_{i=1}^{n} \lambda_i (L^i_{it} + L^j_{it}),$$  \hspace{1cm} (7)

where $L_i$ is the loss of student model, $\lambda_i$ is the teacher importance weight, $n$ is the number of teachers. To further enable the model to control the effects of the teacher adaptively, we set $\lambda_i$ to be learnable parameters that are optimized during training regularization:

$$L = L' + \sum_{i=1}^{n} \frac{1}{\lambda_i} + \left( \sum_{i=1}^{n} \frac{1}{\lambda_i^2} \right)^2,$$  \hspace{1cm} (8)

where the initial value of $\lambda_i$ is set to 1. The $1/\lambda_i$ and $1/\lambda_i^2$ regularizations induce the model to learn to choose more effective teachers.

Note that, with the flexibility of the self-supervised imitation framework, consistency knowledge can be easily transferred to many student recommenders.

## 3 Experiments

### 3.1 Experimental Settings

#### Data Sets

We conduct our experiments on real-world datasets from Amazon review datasets. In this work, we select four subcategories: “Beauty”, “Toys and Games”, “Sports and Outdoors” and “Musical Instruments”. We select users with at least five interaction records for experiments. The data statistics after preprocessing are listed in Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Beauty</th>
<th>Toys</th>
<th>Sports</th>
<th>Musical</th>
</tr>
</thead>
<tbody>
<tr>
<td># Users</td>
<td>22,363</td>
<td>19,412</td>
<td>35,598</td>
<td>10,057</td>
</tr>
<tr>
<td># Items</td>
<td>12,102</td>
<td>11,925</td>
<td>18,358</td>
<td>36,098</td>
</tr>
<tr>
<td># Actions</td>
<td>176,139</td>
<td>148,185</td>
<td>260,739</td>
<td>77,733</td>
</tr>
<tr>
<td># Avg. Actions/User</td>
<td>7.9</td>
<td>7.6</td>
<td>7.3</td>
<td>7.3</td>
</tr>
<tr>
<td># Avg. Actions/Item</td>
<td>14.6</td>
<td>12.4</td>
<td>14.2</td>
<td>2.2</td>
</tr>
</tbody>
</table>

Table 1: The statistics of four datasets after preprocessing.

To evaluate the recommendation’s performance, we split each dataset into training/validation/testing sets. We hold out each dataset into training/testing sets. We hold out testing sets. We hold out each dataset into training/testing sets. We hold out testing sets. We hold out each dataset into training/testing sets. We hold out testing sets. We hold out each dataset into training/testing sets. We hold out testing sets.

#### Evaluation Metrics

Usually, the recommendation system suggests a few items at once. The desired item should be amongst the first few listed items. Therefore, we employ Recall@N and NDCG@N [Li et al., 2013] to evaluate the recommendation performance. In general, higher metric values indicate better ranking accuracy. Recall@k indicates the proportion of cases when the rated item is amongst the top-k items. NDCG@k is the normalized discounted cumulative gain at k, which takes the rank of recommended items into account and assigns larger weights on higher positions. To avoid high computation cost on all user-items in evaluation, following the strategy in [Guo et al., 2019], we randomly draw 99 negative items that have not been engaged with the user and rank the ground-truth item among the 100 items.

### Baselines

We compare with the following sequential recommendation models to justify the effectiveness of our approaches.

- **MostPop** recommends the item according to the popularity measured by the number of user-interactions, which provides a simple non-personalized baseline.
- **GRU4Rec**[Hidasi et al., 2016b] adopts GRU to capture sequential dependencies and makes predictions for session-based recommendation.
- **BERT4Rec**[Sun et al., 2019] learns a bidirectional representation model to make recommendations by learning a cloze objective with reference to BERT.
- **HGN**[Ma et al., 2019] captures both the long-term and short-term user interests by hierarchical gating network.
- **S^3-Rec**[Zhou et al., 2020] is a recently proposed self-supervised learning framework under mutual information maximization principle for the sequential recommendation, which is based on self-attentive neural architecture.
- **GREC**[Yuan et al., 2020] is an encoder-decoder framework that trains the encoder and decoder by a gap-filling mechanism.

### Parameter Settings

For MostPop and GRU4Rec, we implement them with PyTorch. For S^3-Rec, we use two auxiliary self-supervised objectives without extra attribute information to ensure fairness. For other methods, we use the source code provided by their authors. All hyper-parameters are set following the suggestions from the original papers. For SSI, our teacher model bases on BERT4Rec[Sun et al., 2019] and our student model bases on HGN[Ma et al., 2019]. We set the number of the self-attention blocks and the attention heads as 8 and 4. The dimension of the embedding is 256. The hyper-parameters are set as $\lambda_1 = \lambda_2 = \lambda_3 = 1$. We use the Adam optimizer[Kingma and Ba, 2015] with a learning rate of 0.001, where the batch size is set as 256 in the teacher and student model, respectively.

### 3.2 Results and Analysis

#### Overall Results

Table 2 presents the performance comparisons between several baselines and our model (SSI). SSI consistently achieves the best performance on four datasets with all evaluation metrics, verifying our model’s superiority. Compared with BERT4Rec, our model shows much better performance. In
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Baseline</th>
<th>Recall@5</th>
<th>NDCG@5</th>
<th>Recall@10</th>
<th>NDCG@10</th>
<th>Recall@15</th>
<th>NDCG@15</th>
<th>Recall@20</th>
<th>NDCG@20</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MostPop</td>
<td>0.204</td>
<td>0.124</td>
<td>0.297</td>
<td>0.172</td>
<td>0.346</td>
<td>0.188</td>
<td>0.412</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>S3-Rec</td>
<td>0.369</td>
<td>0.285</td>
<td>0.470</td>
<td>0.321</td>
<td>0.539</td>
<td>0.338</td>
<td>0.588</td>
<td>0.351</td>
</tr>
<tr>
<td></td>
<td>GREC</td>
<td>0.373</td>
<td>0.282</td>
<td>0.470</td>
<td>0.314</td>
<td>0.533</td>
<td>0.333</td>
<td>0.576</td>
<td>0.343</td>
</tr>
<tr>
<td></td>
<td>BERT4Rec</td>
<td>0.378</td>
<td>0.290</td>
<td>0.471</td>
<td>0.320</td>
<td>0.541</td>
<td>0.338</td>
<td>0.591</td>
<td>0.350</td>
</tr>
<tr>
<td></td>
<td>GRU4Rec</td>
<td>0.279</td>
<td>0.206</td>
<td>0.366</td>
<td>0.234</td>
<td>0.429</td>
<td>0.250</td>
<td>0.482</td>
<td>0.263</td>
</tr>
<tr>
<td></td>
<td>HGN</td>
<td>0.378</td>
<td>0.288</td>
<td>0.470</td>
<td>0.317</td>
<td>0.531</td>
<td>0.334</td>
<td>0.580</td>
<td>0.345</td>
</tr>
<tr>
<td></td>
<td>SSI</td>
<td>0.389</td>
<td>0.298</td>
<td><strong>0.487</strong></td>
<td><strong>0.329</strong></td>
<td><strong>0.551</strong></td>
<td><strong>0.345</strong></td>
<td><strong>0.605</strong></td>
<td><strong>0.358</strong></td>
</tr>
<tr>
<td>Beauty</td>
<td>MostPop</td>
<td>0.182</td>
<td>0.103</td>
<td>0.235</td>
<td>0.141</td>
<td>0.299</td>
<td>0.169</td>
<td>0.332</td>
<td>0.178</td>
</tr>
<tr>
<td></td>
<td>S3-Rec</td>
<td>0.350</td>
<td>0.265</td>
<td>0.458</td>
<td>0.303</td>
<td>0.530</td>
<td>0.332</td>
<td>0.586</td>
<td>0.336</td>
</tr>
<tr>
<td></td>
<td>GREC</td>
<td>0.359</td>
<td>0.262</td>
<td>0.460</td>
<td>0.305</td>
<td>0.525</td>
<td>0.319</td>
<td>0.576</td>
<td>0.329</td>
</tr>
<tr>
<td></td>
<td>BERT4Rec</td>
<td>0.353</td>
<td>0.268</td>
<td>0.459</td>
<td>0.302</td>
<td>0.529</td>
<td>0.321</td>
<td>0.584</td>
<td>0.334</td>
</tr>
<tr>
<td></td>
<td>GRU4Rec</td>
<td>0.216</td>
<td>0.146</td>
<td>0.313</td>
<td>0.177</td>
<td>0.381</td>
<td>0.195</td>
<td>0.438</td>
<td>0.209</td>
</tr>
<tr>
<td></td>
<td>HGN</td>
<td>0.364</td>
<td>0.275</td>
<td>0.463</td>
<td>0.307</td>
<td>0.528</td>
<td>0.325</td>
<td>0.580</td>
<td>0.335</td>
</tr>
<tr>
<td></td>
<td>SSI</td>
<td>0.369</td>
<td>0.280</td>
<td><strong>0.470</strong></td>
<td><strong>0.313</strong></td>
<td><strong>0.539</strong></td>
<td><strong>0.331</strong></td>
<td><strong>0.595</strong></td>
<td><strong>0.344</strong></td>
</tr>
<tr>
<td>Toys</td>
<td>MostPop</td>
<td>0.199</td>
<td>0.122</td>
<td>0.274</td>
<td>0.145</td>
<td>0.301</td>
<td>0.183</td>
<td>0.368</td>
<td>0.194</td>
</tr>
<tr>
<td></td>
<td>S3-Rec</td>
<td>0.345</td>
<td>0.252</td>
<td>0.462</td>
<td>0.285</td>
<td>0.540</td>
<td>0.307</td>
<td>0.605</td>
<td>0.324</td>
</tr>
<tr>
<td></td>
<td>GREC</td>
<td>0.288</td>
<td>0.203</td>
<td>0.378</td>
<td>0.241</td>
<td>0.464</td>
<td>0.267</td>
<td>0.502</td>
<td>0.282</td>
</tr>
<tr>
<td></td>
<td>BERT4Rec</td>
<td>0.347</td>
<td>0.252</td>
<td>0.463</td>
<td>0.290</td>
<td>0.541</td>
<td>0.311</td>
<td>0.603</td>
<td>0.325</td>
</tr>
<tr>
<td></td>
<td>GRU4Rec</td>
<td>0.234</td>
<td>0.170</td>
<td>0.335</td>
<td>0.193</td>
<td>0.405</td>
<td>0.217</td>
<td>0.464</td>
<td>0.225</td>
</tr>
<tr>
<td></td>
<td>HGN</td>
<td>0.309</td>
<td>0.227</td>
<td>0.416</td>
<td>0.261</td>
<td>0.489</td>
<td>0.281</td>
<td>0.547</td>
<td>0.294</td>
</tr>
<tr>
<td></td>
<td>SSI</td>
<td>0.361</td>
<td>0.263</td>
<td><strong>0.479</strong></td>
<td><strong>0.301</strong></td>
<td><strong>0.558</strong></td>
<td><strong>0.322</strong></td>
<td><strong>0.619</strong></td>
<td><strong>0.336</strong></td>
</tr>
<tr>
<td>Sports</td>
<td>MostPop</td>
<td>0.126</td>
<td>0.048</td>
<td>0.179</td>
<td>0.079</td>
<td>0.206</td>
<td>0.103</td>
<td>0.236</td>
<td>0.117</td>
</tr>
<tr>
<td></td>
<td>S3-Rec</td>
<td>0.243</td>
<td>0.179</td>
<td>0.322</td>
<td>0.206</td>
<td>0.374</td>
<td>0.211</td>
<td>0.419</td>
<td>0.224</td>
</tr>
<tr>
<td></td>
<td>GREC</td>
<td>0.232</td>
<td>0.175</td>
<td>0.298</td>
<td>0.194</td>
<td>0.356</td>
<td>0.211</td>
<td>0.401</td>
<td>0.223</td>
</tr>
<tr>
<td></td>
<td>BERT4Rec</td>
<td>0.244</td>
<td>0.177</td>
<td>0.326</td>
<td>0.204</td>
<td>0.381</td>
<td>0.218</td>
<td>0.422</td>
<td>0.228</td>
</tr>
<tr>
<td></td>
<td>GRU4Rec</td>
<td>0.199</td>
<td>0.148</td>
<td>0.279</td>
<td>0.174</td>
<td>0.341</td>
<td>0.190</td>
<td>0.394</td>
<td>0.203</td>
</tr>
<tr>
<td></td>
<td>HGN</td>
<td>0.227</td>
<td>0.171</td>
<td>0.299</td>
<td>0.194</td>
<td>0.355</td>
<td>0.209</td>
<td>0.399</td>
<td>0.219</td>
</tr>
<tr>
<td></td>
<td>SSI</td>
<td>0.234</td>
<td>0.172</td>
<td>0.328</td>
<td>0.202</td>
<td>0.389</td>
<td>0.218</td>
<td>0.428</td>
<td>0.228</td>
</tr>
<tr>
<td>Musical</td>
<td>MostPop</td>
<td>0.246</td>
<td>0.188</td>
<td>0.331</td>
<td>0.213</td>
<td>0.388</td>
<td>0.228</td>
<td>0.427</td>
<td>0.239</td>
</tr>
</tbody>
</table>

Table 2: Performance comparison of baselines and our approaches, where our approach SSI’s best results are in bold. The underlined numbers are the best results besides SSI.

Figure 3: Our best model’s performance comparison with different aspects of consistency knowledge on four datasets (NDCG@10).

our model, bidirectional item representations are further improved with three aspects of consistency. The integrated imitation network effectively distills consistency-enhanced pre-training knowledge into the student model, which combines the merits of the pre-training teachers and student network. SSI outperforms HGN. The reason is that HGN undertakes the student network in our model. The performance of HGN is effectively improved by imitating self-supervised consistency knowledge. The performance of S3-Rec is inferior to SSI. Although the absence of extra attribute information is a potential influencing factor, another primary reason is that S3-Rec ignores the consistency of sequential recommendation.

**Flexibility of Self-supervised Imitation**

In our self-supervised imitation framework, the student recommender can be any sequential recommendation system. To verify its flexibility, we also list the performance of GRU4Rec as the student model. In Table 2, we notice that GRU4Rec and HGN are both improved by self-supervised imitation (SSIGRU4Rec, SSI), which demonstrates its applicability. More encouragingly, concerning the performance of the base model, GRU4Rec is much inferior to HGN. However, after being augmented with self-supervised imitation, the performance of SSIGRU4Rec is on par with SSI. In conclusion, the self-supervised consistency knowledge can be easily transferred to other student recommenders with the flexibility of the self-supervised imitation framework.

**Impact of Different Aspects of Consistency**

In this paper, we propose three aspects of self-supervised pre-training tasks to abstract consistency knowledge. Figure 3 shows the comparison of effects between the proposed three aspects of consistency, temporal, persona, and global session consistency. Compared with the base model, we observe that independently utilizing temporal, personal, or global session consistency effectively boosts the model performance. The
gains from different aspects of consistency vary on different datasets. However, jointly imitating all three aspects of consistency-enhanced knowledge achieves the best performance.

Impact of Pre-training Model Size
In our model, consistency pre-training is performed upon BERT4Rec. As self-attention based pre-training is computation-expensive, we would like to verify the model performance on different pre-training model layers. Figure 4 illustrates the model performance degradation on both models along with the increased number of self-attention layers. What’s more, the performance gap increases when comparing a relative deep model with eight self-attention layers and a shallow one with only one self-attention layer. Obviously, our model is more robust to computational cost than BERT4Rec. We conjecture that this is because our model effectively enriches the item representation expressiveness with three consistency-distillation objectives.

Impact of Integrated Imitation Learning
Integrated imitation learning transfers consistency-enhanced knowledge from three aspects of teachers to the student. In our framework, we employ the prediction logit imitation as well as the item representation imitation. Table 3 shows the performance differences. Both kinds of imitations contribute to the final performance. Though prediction distribution regularization is quite useful for consistency knowledge transferring, item representation imitation further improves the knowledge distillation efficiency. We speculate that item representation provides a short-cut for consistency knowledge distillation.

Table 3: Effect of different imitation learning methods on Beauty.

<table>
<thead>
<tr>
<th>Imitation Methods</th>
<th>Recall@10</th>
<th>NDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>w/o item representation imitation</td>
<td>0.487</td>
<td>0.329</td>
</tr>
<tr>
<td>w/o prediction distribution imitation</td>
<td>0.483</td>
<td>0.325</td>
</tr>
<tr>
<td>w/o both</td>
<td>0.470</td>
<td>0.317</td>
</tr>
</tbody>
</table>

5 Conclusion
In this work, we improve sequential recommendation consistency with self-supervised imitation. First, three aspects of consistency knowledge are extracted with self-supervision tasks, where temporal and persona consistencies capture user-item dynamics, and the global session consistency provides a global perspective with interaction mutual information. Then an imitation framework integrates the consistency knowledge and transfers it to the student. Due to its special merits of flexibility, consistency knowledge can easily benefit other student recommenders as demand. Experimental results and analysis demonstrate the superiority of the proposed model.

Acknowledgments
This work was supported by National key R&D Program of China (Grant No. 2018YFB0904503). Hongshen Chen and Yonghao Song are the corresponding authors.
References


