Decoupled Progressive Distillation for Sequential Prediction with Interaction Dynamics

KAIXI HU, School of Computer Science, University of Technology Sydney, Australia and School of Computer Science and Artificial Intelligence, Wuhan University of Technology, China
LIN LI* and QING XIE, School of Computer Science and Artificial Intelligence, Wuhan University of Technology, China
JIANQUAN LIU, Visual Intelligence Research Laboratories, NEC Corporation, Japan
XIAOHUI TAO, School of Mathematics, Physics and Computing, University of Southern Queensland, Australia
GUANDONG XU, School of Computer Science, University of Technology Sydney, Australia

Sequential prediction has great value for resource allocation due to its capability in analyzing intents for next prediction. A fundamental challenge arises from real-world interaction dynamics where similar sequences involving multiple intents may exhibit different next items. More importantly, the character of volume candidate items in sequential prediction may amplify such dynamics, making deep networks hard to capture comprehensive intents. This paper presents a sequential prediction framework with Decoupled Progressive Distillation (DePoD), drawing on the progressive nature of human cognition. We redefine target and non-target item distillation according to their different effects in the decoupled formulation. This can be achieved through two aspects: (1) Regarding how to learn, our target item distillation with progressive difficulty increases the contribution of low-confidence samples in the later training phase while keeping high-confidence samples in the earlier phase. And, the non-target item distillation starts from a small subset of non-target items from which size increases according to the item frequency. (2) Regarding whom to learn from, a difference evaluator is utilized to progressively select an expert that provides informative knowledge among items from the cohort of peers. Extensive experiments on four public datasets show DePoD outperforms state-of-the-art methods in terms of accuracy-based metrics.

CCS Concepts: • **Applied computing** → **Sociology**; • **Computing methodologies** → **Neural networks**.

*Corresponding author

This work is partially supported by the National Natural Science Foundation of China (No. 62276196), and Key Research and Development Program of Hubei Province (No. 2021BAA030). The first author has performed this work as a visiting PhD student at University of Technology Sydney (2022–2023) with the support from China Scholarship Council (No. 202106950041).

Authors' addresses: Kaixi Hu, School of Computer Science, University of Technology Sydney, 15 Broadway Road, Ultimo, Sydney, New South Wales, 2007, Australia and School of Computer Science and Artificial Intelligence, Wuhan University of Technology, 122 Luoshi Road, Wuhan, Hubei, 430070, China, issac_hkx@whut.edu.cn; Lin Li, cathylilin@whut.edu.cn; Qing Xie, felixxq@whut.edu.cn, School of Computer Science and Artificial Intelligence, Wuhan University of Technology, 122 Luoshi Road, Wuhan, Hubei, 430070, China; Jianquan Liu, Visual Intelligence Research Laboratories, NEC Corporation, 1753 Shimonumabe, Nakahara Ward, Kawasaki, Kanagawa, 211-0011, Japan, jqliu@nec.com; Xiaohui Tao, School of Mathematics, Physics and Computing, University of Southern Queensland, 487-535 West Street, Darling Heights, Toowoomba, Queensland, 4350, Australia, Xiaohui.Tao@unisq.edu.au; Guandong Xu, School of Computer Science, University of Technology Sydney, 15 Broadway Road, Ultimo, Sydney, New South Wales, 2007, Australia, Guandong.Xu@uts.edu.au.

Additional Key Words and Phrases: Sequential Prediction, Representation Learning, Interaction Dynamics, Knowledge Distillation

ACM Reference Format:
Kaixi Hu, Lin Li, Qing Xie, Jianquan Liu, Xiaohui Tao, and Guandong Xu. 2023. Decoupled Progressive Distillation for Sequential Prediction with Interaction Dynamics. *ACM Trans. Inf. Syst.* 37, 4, Article 111 (November 2023), 35 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Sequential prediction can infer what the next item will be from a volume of candidate items by modeling sequential intents. Discovering the best practice for these methods would be useful for improving demand-based resource allocation, such as user interests in recommender systems [9, 10, 62–64], criminal intents in predictive policing [22–24], and transit intents in location prediction [44, 47]. Recently, sequential prediction methods have evolved from conventional Markov chain models [19, 28, 50] to deep neural models. Many efforts in this space have achieved impressive progress in learning dense vector representations of intents from observed item-item transitions. Furthermore, high-quality intent representations are straightforward at improving prediction accuracy through calculating the correlation between the inferred intents and candidate items [24, 55].

However, in real-world interactions, human activities are continually affected by complicated and volatile environments. Different next items may manifest after similar historical sequences, which leads to inconsistent training samples. As shown in the left part of Figure 1, we summarize three types of interaction dynamics that widely exist in long or short sequences:

- **Sequence 1**. *Multifarious intents co-occur in a long interaction sequence.* The evolving process of real-world interactions usually reflects multiple and alternate sequential intents, for example, both intent A and intent B are within Sequence 1. Even if two sequences share the same items, the intents corresponding to the next item may also differ. As such, a part of dynamics can be tracked from interaction diversity between individuals and items.
- **Sequence 2**. *Short sequences suffer from the lack of discriminative information.* This type of interaction sequence may contain the shared fragment of intent A and intent B, which lacks discriminative information. As such, multiple candidate next items may satisfy the observed sequence and present in the next interaction. Therefore, inactive interactions bring further dynamics into intent analysis.
- **Sequence 3**. *Ubiquitous noise affects the learning of sequential intents.* Environmental noise makes people rarely observe sequences with clear discriminative information. Therefore, the discovery of sequential intents is inherently difficult. It is a necessity to incorporate the knowledge about dynamics into the dense representations of intents.

Despite the common existence of the above dynamics in interaction intents, it is worth noting that with the increment of candidate items, such dynamics may be further amplified [16, 36, 75]. In this situation, the training process may be susceptible to the observed inconsistency in sequence samples, posing challenges to capture a comprehensive intent. More specifically, based on the observed inconsistent training samples, deep prediction networks will suffer from model uncertainty [7, 29], wherein different networks generate distinctive model responses (see an example in Figure 8(a)).

Modeling dynamics of interaction intents in Sequential Prediction. Current sequential prediction methods generally expand the representation space of learnable intents, thus characterizing the intent dynamics. As shown in the right part of Figure 1, distribution-based representation [9, 10] and multiple vector-based representation [24, 41, 42, 52, 64] are two of the most common types of approaches. For distribution-based representations (the dashed curve in Figure 1), several works [9, 10] typically employ Gaussian distributions to represent the dynamic uncertainty

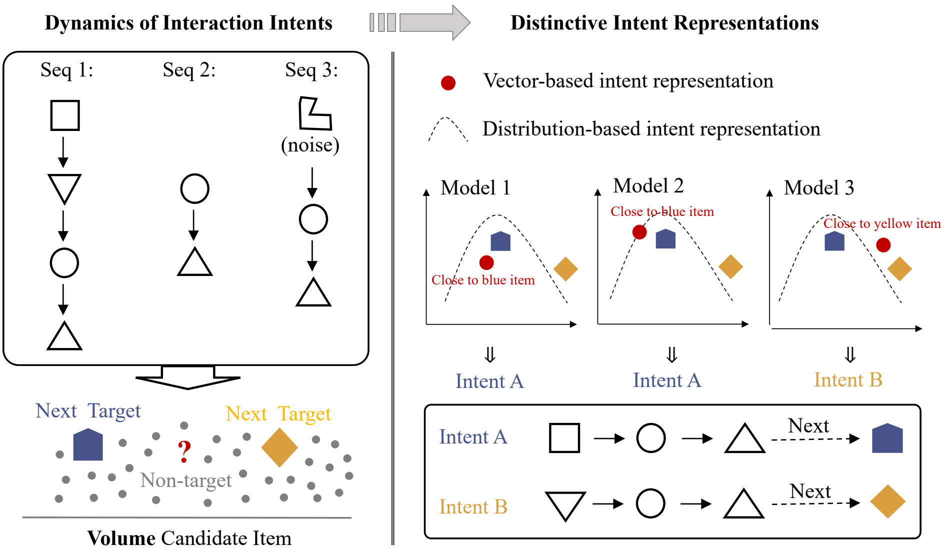


Fig. 1. An illustration of interaction dynamics involving multiple intents and distinctive intent representations. The left part of the diagram describes three types of sequences that may present different next items in the effect of interaction dynamics. And, the right part explains the distinctive representations of inferred intents. The dashed curve and red dot denote the distribution-based [9, 10] and multiple vector-based intent representations, respectively, each of which are common approaches used to model the interaction dynamics.

of inferred intent representations. Meanwhile, to facilitate the computation of representations, most works [24, 52, 56, 63, 64] adopt multiple vector-based representations (the red dots in Figure 1) to capture various learnable intents and then integrate them into a comprehensive intent representation. For example, the intent representations from Model 1 and Model 2 are both close to the blue item, while the intent representation from Model 3 is close to the yellow item. How we can fully exploit the difference among high-level intent representations is not a trivial problem. As a promising knowledge transfer manner, knowledge distillation (KD) is able to make multiple deep networks mimic each other and derive comprehensive intent representations simultaneously. This technique has been widely applied in various fields, such as computer vision (CV) [6, 48], natural language processing (NLP) [33, 49] and recommendation system (RS) [35, 82].

Knowledge Distillation Among Volume Items. As shown in the left bottom part of Figure 1, each input sequence is associated with one next target item, while other related intents are covered in volume non-target items. Vanilla KD-based sequential prediction methods [24, 31, 35, 82] mainly distill knowledge among target and non-target item classes in a unified manner. When confronted with volume classes, some recent works [13, 51, 75] in CV show that such a unified manner makes the training model prone to high-confidence samples and suppresses the learning of other related intents covered in non-target items, since more classes increase training difficulty. However, the most difficult ImageNet dataset discussed in [75] consists of 1,000 classes. In sequential prediction tasks, the candidate items that need to be classified typically range from hundreds of urban events [24] to tens of thousands of web behaviors [10, 79]. As such, it still remains a challenge to effectively accomplish KD within volume items.

To address the aforementioned problem, we propose a sequential prediction framework with decoupled progressive distillation (DePoD for short) that consists of multi-peer prediction networks and a decoupled progressive distillation strategy. Basically, given multiple peer prediction networks, they can independently infer distinctive intents corresponding to the next item and finally generate different responses in the form of probability distributions. To better utilize the response differences in volume candidates items, we revisit the mimicry strategy in KD and reveal different effects of target and non-target items. Inspired by the progressive nature of human cognition [2, 37], we believe that the knowledge among target and non-target items can be transferred between multi-peer prediction networks by *starting from sequence samples with easy target items and small non-target items, and then gradually increasing their training difficulty level.* As such, the proposed DePoD is constructed on the foundation of three principles: (1) **Progressive Difficulty**. For target item distillation in vanilla KD, we extend the learning of high-confidence samples in the earlier training phase and make the training network focus on low-confidence samples that are hard to fit in the later training phase. (2) **Progressive Size**. Non-target items are gradually added into the training phase according to their frequency, which constructs a series of distillation sub-tasks with increasing difficulty to enhance the learning of knowledge among these items. (3) **Progressive Selection**. The training network first learns from the cohort of its peers, and then gradually selects an expert that provides more informative knowledge by a difference evaluator. In summary, the aforementioned two principles (1) and (2) address how to learn between two peers, and the last principal (3) instructs whom to learn from among three or more peers.

Our work makes the following contributions:

- We propose a sequential prediction framework with decoupled progressive distillation (DePoD). This framework employs response difference among multiple peer prediction networks to model the dynamics of interaction intents.
- To enhance KD within volume candidate items, we devise a decoupled progressive distillation strategy, including target item distillation with progressive difficulty, non-target item distillation with progressive size and progressive peer selection.
- Results from extensive experiments on four public datasets, covering urban event and web recommendation, demonstrate that DePoD achieves superior performance over state-of-the-art methods in terms of accuracy-based Top-N metrics, and can flexibly integrate various sequential prediction methods as sequence encoders.

Summarizing the rest of this paper, we first introduce our task in Section 2. Then, we explain our motivation in Section 3, and present the details of our framework DePoD in Section 4. The experimental setup, results and related analyses are reported in Section 5. We review the related work in Section 6 and finally offer conclusions in Section 7.

2 TASK DEFINITION

Sequential prediction for next item is a basic task that can naturally extend to more complex sequence-to-sequence prediction after reorganizing the historical sequence and prediction results into a new input sequence.

Basically, sequential prediction contains objects $\mathcal{O}$ (e.g. users, urban regions) and items $\mathcal{I}$ (e.g. behaviors, events). For each object $o \in \mathcal{O}$, the items that object o interacted with are arranged in chronological order, and form a historical sequence $[\text{item}_1, \cdots, \text{item}_l, \cdots, \text{item}_{L_o}]$, where $\text{item}_l \in \mathcal{I}$ is the item interacted at time step l and L_o is the number of interactions. More formally, let $\mathcal{D} = (\mathcal{X}, \mathcal{Y}) = \{(\boldsymbol{x}_m, \boldsymbol{y}_m)\}_{m=1}^{M}$ be a training set with M samples, where $\boldsymbol{x}_m$ refers to the mth historical sequence with $L-1$ items and $\boldsymbol{y}_m$ denotes the corresponding next item at time step L. L is the maximum length of a sequence. In this work, sequential prediction aims to learn a deep model $\mathcal{F}_\theta : \mathcal{X} \to \mathcal{Y}$ that infers the probability distribution $\boldsymbol{p}$ of candidate items where the item with the maximum probability is the prediction result $\hat{\boldsymbol{y}}$The notations are summarized in Table 1.

Table 1. Notations and Descriptions. For convenience, we omit k in some discussions.

Notations	Description
$\mathcal{O}$, $\lvert\mathcal{O}\rvert$	$\mathcal{O}$ is the set of objects, and $\lvert\mathcal{O}\rvert$ is size of $\mathcal{O}$
$\mathcal{I}$, $\lvert\mathcal{I}\rvert$	$\mathcal{I}$ is the set of items, and $\lvert\mathcal{I}\rvert$ is size of $\mathcal{I}$
L	The maximum length of input historical sequence
K, k	K is the total number of peers and k is the index of peer
d	The dimension of embedding tables
$\boldsymbol{W_E}$	The item look-up table
$\boldsymbol{E} \in \mathbb{R}^{L\times d}$	The embedding matrix of input historical sequence
$\boldsymbol{h}^{(k)}$	The intent representation inferred by the kth peer encoder
$\boldsymbol{z}^{(k)}$	The logit of the kth encoder (model)
$\boldsymbol{p}^{(k)}$	The probability distribution of the kth encoder (model)
$\boldsymbol{p}_{\mathrm{TI}}^{(k)}$	The probability distribution with respect to target item
$\boldsymbol{q}_{\mathrm{NI}}^{(k)}$	The probability distribution with respect to non-target items
$\tilde{\boldsymbol{p}}^{(k)}$, $\tilde{\boldsymbol{p}}_{\mathrm{TI}}^{(k)}$, $\tilde{\boldsymbol{q}}_{\mathrm{NI}}^{(k)}$	The probability distribution of corresponding peer (teacher).
$*$	The index of target item (ground-truth)
π, Π	π is the current training epoch and Π is the total number of epochs
t	t is the progress of current training epoch and its value is between 0 and 1
α	The importance of earlier target item distillation
γ	The intensity of deliberate practice in later target item distillation
β	The importance of non-target item distillation

Due to the existence of real-world interaction dynamics, especially in volume candidate items, distinctive model responses will finally reflect in their output probability distributions. That is, given K different prediction networks, the probability distribution

$$\boldsymbol{p}^{(k)} = [p_1^{(k)}, \cdots, p_i^{(k)}, \cdots, p_{|\mathcal{I}|}^{(k)}] \in \mathbb{R}^{1\times|\mathcal{I}|}, \quad \text{where } p_i^{(k)} = \frac{\exp(z_i^{(k)})}{\sum_{j=1}^{|\mathcal{I}|}\exp(z_j^{(k)})} \tag{1}$$

from the kth network may be different from others. Here, $z_i^{(k)}$ is the corresponding logit of the ith item. As such, this work aims to better exploit the response difference among multiple peers and use them together to optimize their learning of intent representations.

3 REVISITING KNOWLEDGE DISTILLATION

KD adopts a mimicry strategy that employs cross-entropy (CE) or relative entropy to transfer knowledge between teacher and student networks [21, 74, 75], which is particularly suitable for modeling the dynamics of interaction intents. To enhance the distillation within volume items, we first revisit the mimicry strategy and explain some findings that motivate our proposed DePoD.

Decoupled Formulation of KD. Intuitively, for a training sample $(\boldsymbol{x}_m, \boldsymbol{y}_m) \in \mathcal{D}$, a comprehensive next intent representation with high posterior [74] should be close to the most likely next target item, and present different distances to non-target items according to the dynamics of interaction intents. Therefore, it is necessary to analyze the effect of KD on target and non-target items, respectively. However, their probability distribution $\boldsymbol{p}^{(k)}$ is coupled by the softmax of logits in Equation (1). To release non-target items, inspired by the independent model probabilities in [75], we further define the probability distribution $\boldsymbol{p}_{\mathrm{TI}}^{(k)} \in \mathbb{R}^{1\times 2}$ with respect to target item (TI) as:

$$\boldsymbol{p}_{\mathrm{TI}}^{(k)} = \left[p_*^{(k)},\ p_{\backslash *}^{(k)}\right] = \left[\frac{\exp(z_*^{(k)})}{\sum_{j=1}^{|\mathcal{I}|}\exp(z_j^{(k)})},\ \frac{\sum_{l=1,l\neq *}^{|\mathcal{I}|}\exp(z_l^{(k)})}{\sum_{j=1}^{|\mathcal{I}|}\exp(z_j^{(k)})}\right], \tag{2}$$

and the probability distribution related to non-target items (NI) as

$$\begin{aligned}
\boldsymbol{q}_{\mathrm{NI}}^{(k)} &= [q_1^{(k)}, \cdots, q_i^{(k)}, \cdots, q_{|\mathcal{I}|}^{(k)}] \in \mathbb{R}^{1\times(|\mathcal{I}|-1)}, \\
q_i^{(k)} &= \frac{\exp(z_i^{(k)})}{\sum_{j=1,j\neq *}^{|\mathcal{I}|}\exp(z_j^{(k)})}, \quad s.t. \sum_{i=1,i\neq *}^{|\mathcal{I}|} q_i^{(k)} = 1
\end{aligned} \tag{3}$$

where "$*$" is the index of target item (ground-truth) and $p_*^{(k)}$ denotes the confidence of samples. Then, we have $p_i^{(k)} = q_i^{(k)} \cdot p_{\backslash *}^{(k)}$, and use tilde (~) for the symbol of corresponding probability distribution from a teacher network. The cross-entropy of KD can be rewritten as:

$$\begin{aligned}
\mathcal{L}_{\mathrm{KD}}^{(k)} = \mathrm{CE}(\tilde{\boldsymbol{p}}^{(k)} \| \boldsymbol{p}^{(k)}) &= -\tilde{p}_*^{(k)}\log(p_*^{(k)}) - \sum_{i=1,i\neq *}^{|\mathcal{I}|} \tilde{p}_i^{(k)}\log(p_i^{(k)}) \\
&= -\tilde{p}_*^{(k)}\log(p_*^{(k)}) - \tilde{p}_{\backslash *}^{(k)} \sum_{i=1,i\neq *}^{|\mathcal{I}|} \tilde{q}_i^{(k)} \left[\log(q_i^{(k)}) + \log(p_{\backslash *}^{(k)})\right] \\
&= \underbrace{\underbrace{-\tilde{p}_*^{(k)}\log(p_*^{(k)})}_{\textcircled{1}} \underbrace{-\tilde{p}_{\backslash *}^{(k)}\log(p_{\backslash *}^{(k)})}_{\textcircled{2}}}_{\textbf{Target: } \mathrm{CE}(\tilde{\boldsymbol{p}}_{\mathrm{TI}}^{(k)} \| \boldsymbol{p}_{\mathrm{TI}}^{(k)})} - \underbrace{\tilde{p}_{\backslash *}^{(k)}}_{\textcircled{3}} \underbrace{\sum_{i=1,i\neq *}^{|\mathcal{I}|} \tilde{q}_i^{(k)}\log(q_i^{(k)})}_{\substack{\textcircled{4}\ \textbf{Non-target:} \\ \mathcal{L}_{\mathrm{NI}}^{(k)} = \mathrm{CE}(\tilde{\boldsymbol{q}}_{\mathrm{NI}}^{(k)} \| \boldsymbol{q}_{\mathrm{NI}}^{(k)})}}\,.
\end{aligned} \tag{4}$$

Our Analysis. The Equation (4) reformulates the vanilla KD loss into two parts, i.e. target item part (term ① and term ②), and non-target item part (term ④) with the weight $\tilde{p}_{\backslash *}^{(k)}$ (term ③). Drawing on the work of [75], the term ③ suppresses the term ④ on high-confidence samples supervised by the teacher network. The term ② also suppresses the learning of low-confidence samples. The decoupled formulation parses these rewarding parts (i.e., term ① and term ④) that exploit response difference to facilitate comprehensive intent representations. We conclude their effects as follows:

- *Why progressive difficulty?* In line with [13, 51], we observe the term ① in the target item part is related to the training difficulty of samples where the probability $\tilde{p}_*^{(k)}$ is the importance weight. This will increase the contribution of high-confidence samples that are well-predicted. However, inspired by curriculum learning [2], deep models tend to learn in a meaningful low-to-high difficulty scheme, and tend to focus on some potential samples with low-confidence in the later training phase. Especially in sequential prediction, low-confidence samples commonly exist due to the real-world interaction dynamics between sequences and volume candidate items. Such samples may also be informative and it is not advisable to neglect them in the whole training process.
- *Why progressive size?* The term ④ in the non-target item part transfers knowledge among volume non-target items (i.e., dark knowledge [13, 75]). Such knowledge reflects the probability differences that various non-target items corresponding to different intents present in the next interaction, which is crucial for modeling their dynamics. Due to the challenging volume

of items in sequential prediction, we suppose that it is difficult to directly make the training network discriminate all non-target items. Furthermore, curriculum learning [2] inspires the training of deep models by starting with a small subtask. We attempt to construct a small subset of non-target items, and then gradually increase the size of this subset according to the item frequency.

- *Why progressive selection?* Some works [45, 48, 83] observe that effective knowledge transfer between two networks is up to a certain response difference, not smaller or larger. Given multiple teachers, the training network will have more choices for finding an expert that shows adequate difference, ultimately obtaining better performance. This can be motivated by a progressive process where people (novice) first learns general knowledge from various individuals in the earlier phase. With increasing experience, they gradually select an expert who provides informative knowledge [1, 17].

Based on the above reformulation and conclusions, we propose a novel framework DePoD. In particular, we add training progress $t = \pi/\Pi$ to achieve decoupled progressive distillation, where π is the current training epoch and Π is the total epochs. The term ① from the kth prediction network is denoted as $\mathcal{L}_{\text{One}}^{(k)} = -\alpha\tilde{p}_*^{(k)}\log(p_*^{(k)})$ where α is a coefficient to adjust its importance. In terms of the rewarding term ① and term ④, our decoupled progressive distillation strategy can be defined as follows:

$$\mathcal{L}_{\text{KD}}^{(k)} \Rightarrow \mathcal{L}_{\text{DePoD}}^{(k)} = \underbrace{\Gamma(t, \mathcal{L}_{\text{One}}^{(k)}, \mathcal{L}_{\text{DP}}^{(k)})}_{\textbf{Target: } \mathcal{L}_{\text{TI}}^{(k)}} + \underbrace{\text{CE}(\tilde{\boldsymbol{q}}_{\text{NI}}^{(k)}(t) \| \boldsymbol{q}_{\text{NI}}^{(k)}(t))}_{\textbf{Non-target: } \mathcal{L}_{\text{NI}}^{(k)}} . \tag{5}$$

For the target item part, we extend the earlier training phase ($\mathcal{L}_{\text{One}}^{(k)}$) and devise another learning manner for the later training phase ($\mathcal{L}_{\text{DP}}^{(k)}$). $\Gamma(\cdot)$ is a function to switch the earlier and later training according to t. For the non-target item part, we employ the training progress t to mask non-target items in their probability distribution $\boldsymbol{q}_{\text{NI}}^{(k)}$. The effectiveness of different parts is discussed in the experimental results in Section 5.3.

4 PROPOSED FRAMEWORK: DePoD

In this paper, our key idea is to capture the dynamics of interaction intents by enhancing the distillation among target and volume non-target items according to the progressive nature of human cognition [2, 37]. To this end, we propose an encoder-agnostic sequential prediction framework (DePoD) that can integrate various sequential prediction models and make them learn from one another. It consists of multi-peer prediction networks and a decoupled progressive distillation strategy. The overall framework of DePoD is shown in Figure 2. In what follows, we start with **Multi-Peer Prediction Networks** to construct distinctive intent representations in Section 4.1, and employ **Target Item Distillation with Progressive Difficulty** and **Non-target Item Distillation with Progressive Size** to transfer knowledge between two peers in Section 4.2 and Section 4.3, respectively. Then, we present **Progressive Peer Selection** from a cohort of peers in Section 4.4 that is based on the foundation of target and non-target item distillation between two peers. Finally, the **Joint Optimization** and holistic training procedure are both introduced in Section 4.5.

4.1 Multi-Peer Prediction Networks

As shown in the left of Figure 2, the multi-peer prediction networks mainly consist of a sequence embedding module, an intent representation learning module with multiple peer encoders and a next item prediction module. Note that multiple distinctive intent representations can be obtained from each peer encoder. The embedding and prediction modules before and after encoders are

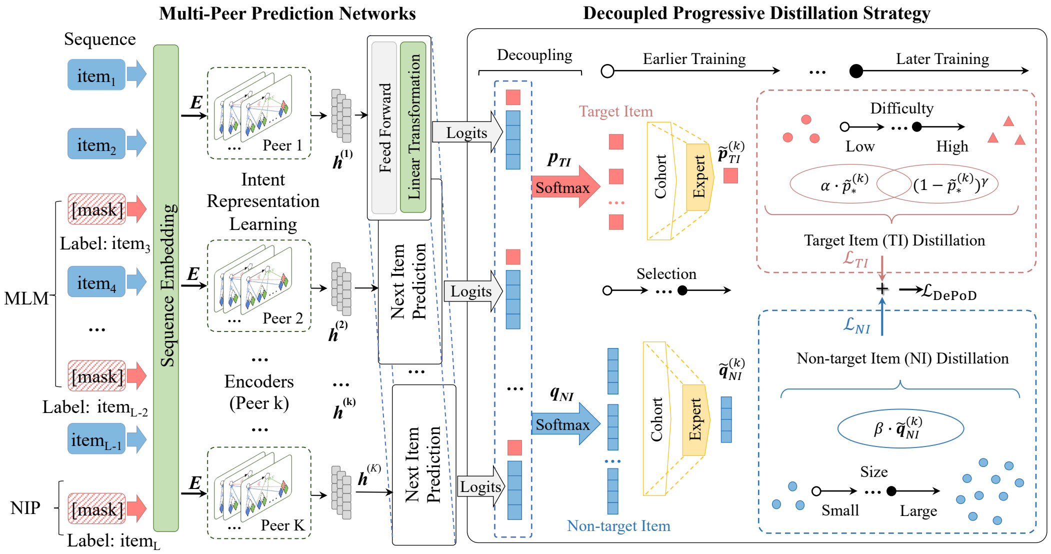


Fig. 2. The framework of our proposed DePoD. DePoD mainly consists of multi-peer prediction networks and a decoupled progressive distillation strategy. In particular, different encoders are first employed as peers to derive multiple intent representations $\boldsymbol{h}^{(k)}$ with distinctive knowledge. Then, our decoupled progressive distillation strategy utilizes their response difference and adaptively adjusts the learning priority of target and non-target items. This makes the prediction network progressively model the dynamics of interaction intents in the context of volume candidate items.

shared in the framework. This makes the encoders reach a common ground on the input and output of sequence embedding, and focus on the modeling of various intents.

4.1.1 Sequence Embedding Module. Basically, an interaction sequence contains a sequence of historical items $\boldsymbol{x}_m = [\text{item}_1, \text{item}_2, \cdots, \text{item}_{\text{L-1}}]$, the corresponding next item $\boldsymbol{y}_m = [\text{item}_\text{L}]$, and several "[mask]" tokens that replace the predicted items. To exploit the self-supervised information, both mask language modeling (MLM) and next item prediction (NIP) can be applied in the training phase, while only NIP is performed in the testing phase.

- Mask Language Modeling (MLM). It randomly masks a proportion of items in the historical sequence $\boldsymbol{x}_m$, and further makes the remaining items predict them. With the enhancement of MLM, more samples can be generated to train our progressive distillation framework.
- Next Item Prediction (NIP). Next item is our final goal that is performed in both the training and testing phase in sequential prediction. To this end, a "[mask]" token is added at the end of historical sequence $\boldsymbol{x}_m$ at time step L. Together, they form an input sequence for our DePoD.

With the above masked items, each of them refers to a training sample for our progressive distillation framework. Then, the input sequence is embedded into a d-dimensional representation matrix $\boldsymbol{E} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_L] \in \mathbb{R}^{L\times d}$ through an item look-up table $\boldsymbol{W}_E \in \mathbb{R}^{|\mathcal{I}|\times d}$. $|\mathcal{I}|$ is the number of candidate items. If the follow-up encoder (such as Transformer) cannot capture position information, we should further introduce another trainable matrix $\boldsymbol{S} = [\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_L] \in \mathbb{R}^{L\times d}$ by following [55], i.e., $\boldsymbol{E} = \boldsymbol{E} + \boldsymbol{S}$. Finally, the embedding matrix $\boldsymbol{E}$ is fed into different encoder branches simultaneously.

4.1.2 Intent Representation Learning Module. The intent representation learning module aims to model item transitions in the input embedding matrix $\boldsymbol{E}$, and further infers the intent representation vector $\boldsymbol{h}^{(k)} = f^{(k)}(\boldsymbol{E}) \in \mathbb{R}^{1\times d}$ corresponding to the "[mask]" token where $f^{(k)}(\cdot)$ denotes the kth encoder. Then, the intent representation vector $\boldsymbol{h}^{(k)}$ is fed into the next item prediction module, which generates the probability distribution over the item set. In practice, most deep neural methods devise a specific sequence structure (e.g., GRU, CNN, Transformer) to meet the requirement of various scenarios. To this end, our DePoD is an encoder-agnostic learning framework that is capable of integrating various sequential prediction methods (e.g., BERT4Rec [55], GRU4Rec [20] and Caser [57]). In this paper, we mainly adopt the popular Transformer-based BERT4Rec as encoders [61], since it captures intents without respect to distance and shows promising performance in a majority of tasks. Additionally, we conduct related experiments to investigate the effectiveness of DePoD when integrating other sequential prediction methods with different sequence structures in Section 5.4.1.

How to work with the follow-up decoupled progressive distillation. The dynamics of interaction intents will ultimately result in distinctive model responses. And, the aim of our decoupled progressive distillation is to utilize their response difference in the forms of probability distributions and facilitate the learning of comprehensive intent representations. Here, multiple different sequence encoders are an indispensable part to enable the discovery of interaction dynamics. These encoders are regarded as peers for each other with distinctive knowledge, and respectively infer intent representations $\boldsymbol{h}^{(k)}$. Interestingly, some works [24, 38, 74] have shown that several identical network structures with random initialization are sufficient to induce diversity and generate response difference. Based on the findings of these studies, we adopt multiple identical Transformer encoders [61] with different initialization as peers to capture the dynamics of interaction intents. The combinations of different sequence structures are also discussed in Section 5.4.1.

4.1.3 Next Item Prediction Module. Multiple encoders generate different intent representations $\boldsymbol{h}^{(k)}$ to infer the masked item and all of them are fed into the shared next item prediction module one-by-one. In this module, to enhance the alignment between different representations, we first devise an additional feed forward layer that is identically applied to the output of each peer encoder. Inspired by the work of [61], the item look-up table $\boldsymbol{W_E}$ is reused as the weight of the pre-softmax linear transformation. Formally, the logit $\boldsymbol{z}^{(k)} \in \mathbb{R}^{1\times|\mathcal{I}|}$ of the kth peer encoder is defined as follows:

$$\boldsymbol{z}^{(k)} = \mathrm{GeLu}(\boldsymbol{h}^{(k)}\boldsymbol{W}_O + \boldsymbol{b}_O)\boldsymbol{W}_E^T + \boldsymbol{b}_E, \tag{6}$$

where $\boldsymbol{W_O} \in \mathbb{R}^{d\times d}$ is the trainable parameter matrix, and $\boldsymbol{b_O} \in \mathbb{R}^{1\times d}$ and $\boldsymbol{b_E} \in \mathbb{R}^{1\times|\mathcal{I}|}$ are the biases.

Based on the logit $\boldsymbol{z}^{(k)}$, probability distributions $\boldsymbol{p}^{(k)}$, $\boldsymbol{p}_{\mathrm{TI}}^{(k)}$, $\boldsymbol{q}_{\mathrm{NI}}^{(k)}$ can be derived from Equations (1), (2) and (3), respectively. These distinctive responses from peers provide additional information to model the dynamics of interaction intents and will be exploited in our follow-up decoupled progressive distillation in Equation (5). In the context of volume candidate next items, we further introduce how to utilize the response difference in target item distillation (Section 4.2) and non-target item distillation (Section 4.3) between two peers. They are the foundation of distillation among multi-peer prediction networks (Section 4.4).

4.2 Target Item Distillation with Progressive Difficulty

As discussed in Section 3, the probability $\tilde{p}_*^{(k)}$ in the target item part has a strong relationship with training difficulty of the input sequence sample. As a result, vanilla KD will make each prediction network prone to the samples with high-confidence supervised by its peers, and the samples with low-confidence will reduce their contributions in the overall training signal. However, starting from samples that are easy to learn, i.e., high-confidence samples, is just an early phase in human cognition [37]. Following curriculum learning [2], deep learning models tend to benefit from potential low-confidence samples in the later training phase. Such samples may also be informative yet hard to fit, and it is not advisable to ignore them in the whole training process. Furthermore, deliberate practice [1, 17] provides a complementary training manner with mimicry, where teachers usually instruct individuals to obtain higher performance by refining less accomplished behaviors. To this end, we propose to distill target items by gradually enhancing low-confidence samples.

Deliberate Practice. Inspired by several cognitive theories [1, 2, 17], we first devise a novel target item distillation loss that can compensate the weakness of term ① and focus on the low-confidence samples in the later training phase, namely deliberate practice. Initially, the novel target item distillation loss $\mathcal{L}_{\text{DP}}^{(k)}$ can be defined as follows:

$$\mathcal{L}_{\text{DP}}^{(k)} = -\psi_{DP}(\tilde{p}_*^{(k)}) \cdot log(p_*^{(k)}), \tag{7}$$

where $\tilde{p}_*^{(k)}$ is the prediction probability of ground-truth from its peer and $\psi_{DP}(\cdot)$ is a function of $\tilde{p}_*^{(k)}$ that refers to the intensity of deliberate practice. In terms of the response difference between the kth training network and its peer, the deliberate practice intensity $\psi_{DP}(\tilde{p}_*^{(k)})$ is expected to adjust the learning of low-confidence samples via the following aspects:

(1) Excessively low confidence samples could also possibly be noise. Over-fitting such noisy samples will hurt the generalization performance [66]. As such, the kth network should not highlight it in the later deliberate practice phase.
(2) For the sample with consistently correct (or consistently wrong) responses, its probabilities $\tilde{p}_*^{(k)}$ and $p_*^{(k)}$ are relatively larger (or smaller) in overall samples. In this case, the deliberate practice intensity should be relatively smaller (or larger) to reduce (or amplify) the loss.
(3) For the sample with inconsistent responses, the value of deliberate practice intensity need to be between the two situations mentioned in aspect (2) above, which enables us to distinguish samples better. In this way, the correctly predicted network will enhance the learning of this sample to avoid being wrong or result in catastrophic forgetting [59] in the following training, while the wrongly predicted network will also consider the correct response to produce appropriate gradients.

In this work, we devise a simple yet effective $\psi_{DP}(\tilde{p}_*^{(k)}) = (1 - \tilde{p}_*^{(k)})^\gamma$ that satisfies the above aspects, where γ is a coefficient to adjust the deliberate practice intensity. In particular, to avoid over-fitting excessively low confidence samples, we assume that a sample is highly possible to be noise if it cannot be well-predicted by all networks ($\forall k$ network). Hence, the probability from its peer can be truncated as follows:

$$\tilde{p}_*^{(k)} = \begin{cases} 1, & \text{rank}(p_*^{(\forall k)}) < \varepsilon \cdot |\mathcal{B}| \\ \tilde{p}_*^{(k)}, & \text{otherwise} \end{cases} \tag{8}$$

where ε is the truncation proportion, $|\mathcal{B}|$ is the batch size of training samples and rank$(\cdot)$ denotes the probability rank among all training samples in ascending order. Finally, for the kth network, the loss can be formally defined as:

$$\mathcal{L}_{\text{DP}}^{(k)} = -(1 - \tilde{p}_*^{(k)})^\gamma \cdot log(p_*^{(k)}). \tag{9}$$

Progressive Target Item Loss. The conventional term ① in Equation (4) concerns the high-confidence samples that works in the earlier training phase, and our proposed deliberate practice in Equation (9) aims to further exploit low-confidence samples that have the potential for boosting performance in the later training phase. To combine the training manners, a progressive training process can be devised as follows:

$$\mathcal{L}^{(k)}_{\text{TI}}(\tilde{\boldsymbol{p}}^{(k)}_{\text{TI}}\|\boldsymbol{p}^{(k)}_{\text{TI}}) = \begin{cases} \mathcal{L}^{(k)}_{\text{One}}, & \text{if } t < \tau_0, \\ \tau_r\mathcal{L}^{(k)}_{\text{One}} + (1-\tau_r)\mathcal{L}^{(k)}_{\text{DP}}, & \text{if } \tau_0 \le t \le \tau_1 \\ \mathcal{L}^{(k)}_{\text{DP}}, & \text{if } t > \tau_1 \end{cases} \tag{10}$$

where $t = \pi/\Pi$ denotes the progress of the current training epoch. Here, τ_0 and τ_1 ($\tau_0 < \tau_1$) are the predefined parameters to ensure sufficient training of different phases. For the intermediate training phase, we introduce a uniform distribution to generate a random number $\text{rand}(0,1)$ between 0 and 1 in each iteration to switch the training manners. And τ_r is equal to 1 when $t < \text{rand}(0,1)$, otherwise it is 0.

4.3 Non-target Item Distillation with Progressive Size

Non-target item distillation in the term ④ of Equation (4) transfers the knowledge between peers. Such knowledge reflects the probability differences in non-target items present over the next interaction, which is beneficial for modeling the dynamics of interaction intents. However, volume non-target items increases the complexity of the knowledge, resulting in the challenge of enabling prediction networks to better discriminate their probability differences.

Progressive Non-target Item Loss. Inspired by curriculum learning [2], the introduction of such knowledge can start from a sub-task with small non-target items, and then gradually increase its size. To this end, instead of directly exposing all non-target items, they can be added into the training process gradually by constructing an increasing subset. Specifically, a masking vector $\boldsymbol{c} = [\cdots, 0, \cdots, 1, \cdots] \in \mathbb{R}^{1\times|\mathcal{I}|}$ is introduced to present the non-target items that the training network needs to discriminate in the next iteration. The probability distribution of non-target items can be rewritten as:

$$\begin{gathered} \boldsymbol{q}^{(k)}_{\text{NI}}(t) = \left[q^{(k)}_1, q^{(k)}_2, \cdots, q^{(k)}_{|\mathcal{I}|}\right] = \text{softmax}(\boldsymbol{z}^{(k)} - 1000 \cdot \boldsymbol{c}), \\ \text{s. t.} \quad c_* = 1 \quad \text{and} \quad \sum_{i=1, i\neq *}^{|\mathcal{I}|} c_i = \begin{cases} (1-t)\cdot|\mathcal{I}|, & \text{if } t < \tau_1 \\ 0, & \text{otherwise} \end{cases} \end{gathered} \tag{11}$$

where c_i is an element in the masking vector $\boldsymbol{c}$, $t = \pi/\Pi$ denotes the training progress and τ_1 ($\tau_0 < \tau_1$) is a predefined parameter to ensure the sufficient training of all non-target items. Here, the value of masked non-target item and target item in $\boldsymbol{c}$ is set to 1, otherwise 0. For numerical stability, we use a relatively large constant of 1,000 to remove the contributions of masked items. Thus, the progressive non-target item loss for the kth peer can be defined as follows:

$$\mathcal{L}^{(k)}_{\text{NI}}\left(\tilde{\boldsymbol{q}}^{(k)}_{\text{NI}}(t)\|\boldsymbol{q}^{(k)}_{\text{NI}}(t)\right) = -\beta \cdot \sum_{i=1}^{|\mathcal{I}|} \tilde{q}^{(k)}_i \log(q^{(k)}_i), \tag{12}$$

where β is a coefficient that adjusts the importance of non-target item distillation and $\tilde{q}^{(k)}_i$ is the corresponding probability in the probability distribution $\tilde{\boldsymbol{q}}^{(k)}_{\text{NI}}(t)$ of its peer.

How to Generate Masking Vectors. For non-target item distillation, the masking vector $\boldsymbol{c}$ is our major contribution. With the advance of training process t, the masked non-target items will gradually decrease and finally degrade to a one-hot vector of label, which makes the training network continuously receive knowledge related to the newly added non-target items. As such, how to sample non-target items and generate the masking vector $\boldsymbol{c}$ is an inevitable problem, which impacts the learning order of non-target items. Since item frequency is an important factor that affects the prediction performance in many sequential prediction tasks [25, 39], we consider

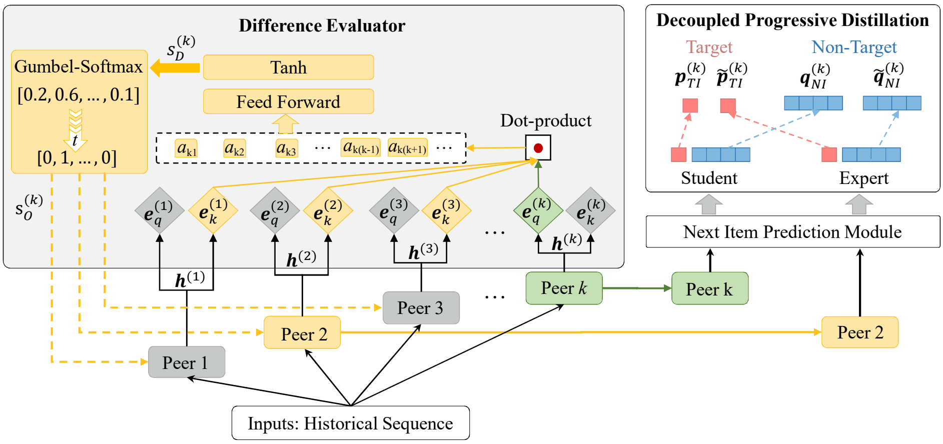

Difference Evaluator
$s_D^{(k)}$
Gumbel-Softmax
[0.2, 0.6, ..., 0.1]
t
[0, 1, ..., 0]
Tanh
Feed Forward
a_{k1} a_{k2} a_{k3} ⋯ $a_{k(k-1)}$ $a_{k(k+1)}$ ⋯
Dot-product
$s_O^{(k)}$
$e_q^{(1)}$ $e_k^{(1)}$ $e_q^{(2)}$ $e_k^{(2)}$ $e_q^{(3)}$ $e_k^{(3)}$ … $e_q^{(k)}$ $e_k^{(k)}$
$h^{(1)}$ $h^{(2)}$ $h^{(3)}$ $h^{(k)}$
Peer 1
Peer 2
Peer 3
Peer k
Inputs: Historical Sequence
Decoupled Progressive Distillation
Target
Non-Target
$p_{TI}^{(k)}$ $\tilde{p}_{TI}^{(k)}$ $q_{NI}^{(k)}$ $\tilde{q}_{NI}^{(k)}$
Student
Expert
Next Item Prediction Module
Peer k
Peer 2

Fig. 3. Process of progressive peer selection via a trainable difference evaluator. In the earlier phase, the kth encoder (Peer k) learns from the cohort of its peers, and with the increment of training progress t, Peer k gradually turns to the sample-wise learning from an expert.

sampling masked non-target items based on a frequency-based distribution. In the frequency-based proposal, according to our empirical findings in Section 5.4.2, we find that the effect of item frequency varies with the real-world scenarios.

4.4 Progressive Peer Selection

The Range of Response Difference. The aforementioned target and non-target distillation mainly utilize the response difference between two peers to model the dynamics of interaction intents. These are the foundation of distillation among our multi-peer prediction networks (three or more peers), enabling any two of them to learn from each other and further boost the performance. Intuitively, we may hope the difference between the model $\boldsymbol{p}^{(k)}$ and its peer $\tilde{\boldsymbol{p}}^{(k)}$ are as large as possible to push the distillation. However, the range of their response difference G can be inferred as follows:

$$G = \|\boldsymbol{p}^{(k)} - \tilde{\boldsymbol{p}}^{(k)}\|_1 = \|(\boldsymbol{p}^{(k)} - \boldsymbol{y}) - (\tilde{\boldsymbol{p}}^{(k)} - \boldsymbol{y})\|_1 \geq \|\boldsymbol{p}^{(k)} - \boldsymbol{y}\|_1 - \|\tilde{\boldsymbol{p}}^{(k)} - \boldsymbol{y}\|_1, \qquad (13)$$

where $\boldsymbol{y}$ is the one-hot embedding of the ground-truth, and

$$G = \|\boldsymbol{p}^{(k)} - \tilde{\boldsymbol{p}}^{(k)}\|_1 < \|\boldsymbol{p}^{(k)} - \boldsymbol{y}\|_1 \qquad (14)$$

when its peer shows superior performance. As such, their response difference ought to be neither too large nor too small. Some works [45, 48, 83] have observed performance degradation in overly large differences in various scenarios. The empirical evidence regarding sequential predictions are also present in Section 5.3.5. In this work, to adaptively and dynamically find a peer with adequate difference, we develop sample-wise progressive peer selection via a trainable difference evaluator.

Difference Evaluator. Specifically, the evaluator calculates the difference correlations between any two peers, and then transforms their correlations into difference scores. The peer with adequate difference can be selected progressively according to the difference scores. First, as shown in Figure 3, multiple encoders are peers for each other. Given the intent representation $\boldsymbol{h}^{(k)}$ from the kth encoder (Peer k), the difference evaluator dynamically calculates the difference scores vi aa dot-product operator between $\boldsymbol{h}^{(k)}$ and the representations $\boldsymbol{H}_p^{(k)} = \boldsymbol{h}^{(1}[, \cdots, \boldsymbol{h}^{(j)}, \cdots, \boldsymbol{h}^{(K)} \in \quad]$

$\mathbb{R}^{(K-1)\times d}(j \neq k)$ from its peers. Inspired by the query-key matching manner in [61], the evaluation process can be formulated as follows:

$$\begin{aligned} \boldsymbol{e}_q^{(k)} &= \boldsymbol{h}^{(k)}\boldsymbol{W}_q + \boldsymbol{b}_q, \\ \boldsymbol{E}_k^{(k)} &= \boldsymbol{H}_p^{(k)}\boldsymbol{W}_k + \boldsymbol{b}_k = [\boldsymbol{e}_k^{(1)}, \cdots, \boldsymbol{e}_k^{(j)}, \cdots, \boldsymbol{e}_k^{(K)}], \quad \text{where} \quad j \neq k \\ \boldsymbol{a}_D^{(k)} &= \boldsymbol{e}_q^{(k)}(\boldsymbol{E}_k^{(k)})^{\mathrm{T}}, \end{aligned} \tag{15}$$

where $\boldsymbol{W}_q \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}_q \in \mathbb{R}^{1\times d}$ are the trainable parameter matrix and bias of query vector $\boldsymbol{e}_q^{(k)} \in \mathbb{R}^{1\times d}$, and $\boldsymbol{W}_k \in \mathbb{R}^{d\times d}$ and $\boldsymbol{b}_k \in \mathbb{R}^{1\times d}$ are the trainable parameter matrix and bias of key matrix $\boldsymbol{E}_k^{(k)} \in \mathbb{R}^{(K-1)\times d}$. Here, the correlation between peer k and other peers can be established through calculating their query-key vector $\boldsymbol{a}_D^{(k)} \in \mathbb{R}^{1\times(K-1)}$. Their difference scores can be derived by a nonlinear transformation as:

$$\boldsymbol{s}_D^{(k)} = \mathrm{Tanh}(\boldsymbol{a}_D^{(k)}\boldsymbol{W}_D + \boldsymbol{b}_D), \tag{16}$$

where $\boldsymbol{W}_D \in \mathbb{R}^{(K-1)\times(K-1)}$ and $\boldsymbol{b}_D \in \mathbb{R}^{1\times(K-1)}$ are the trainable parameter matrix and bias, respectively. Here, $\boldsymbol{s}_D^{(k)} \in \mathbb{R}^{1\times(K-1)}$ contains the difference scores between peer k and other peers.

Progressive Selection. The idea of progressive peer selection follows that people (novices) are prone to learning various general knowledge from a cohort in the earlier learning phase, then gradually turn to an expert teacher who provides more informative knowledge in the later phase [1, 17]. To this end, we then introduce the training progress $t = \pi/\Pi$ and a Gumbel-Softmax function [30] to gradually derive one-hot weight vector of peers. This process can be written as:

$$s_{O,i}^{(k)} = \frac{\exp\left((s_{D,i}^{(k)} + g_i)/(1-t)\right)}{\sum_{j=1}^{K-1}\exp\left((s_{D,j}^{(k)} + g_j)/(1-t)\right)}, \quad \text{for} \quad i = 1, \cdots, K-1 \tag{17}$$

where $\boldsymbol{s}_O^{(k)} = \left[s_{O,1}^{(k)}, \cdots, s_{O,i}^{(k)}, \cdots, s_{O,K-1}^{(k)}\right] \in \mathbb{R}^{1\times(K-1)}$ is the weight vector, and g_i is drawn from the Gumbel distribution. Finally, the probability distribution of the kth encoder's teacher can be computed as:

$$\begin{aligned} \tilde{\boldsymbol{p}}_{\mathrm{TI}}^{(k)} &= \boldsymbol{s}_O^{(k)} \cdot [\boldsymbol{p}_{\mathrm{TI}}^{(1)}, \cdots, \boldsymbol{p}_{\mathrm{TI}}^{(j)}, \cdots, \boldsymbol{p}_{\mathrm{TI}}^{(K)}], \quad \text{for} \quad j = 1, \cdots, K \quad \text{and} \quad j \neq k, \\ \tilde{\boldsymbol{q}}_{\mathrm{NI}}^{(k)} &= \boldsymbol{s}_O^{(k)} \cdot [\boldsymbol{q}_{\mathrm{NI}}^{(1)}, \cdots, \boldsymbol{q}_{\mathrm{NI}}^{(j)}, \cdots, \boldsymbol{q}_{\mathrm{NI}}^{(K)}], \quad \text{for} \quad j = 1, \cdots, K \quad \text{and} \quad j \neq k. \end{aligned} \tag{18}$$

The above $\tilde{\boldsymbol{p}}_{\mathrm{TI}}^{(k)}$ and $\tilde{\boldsymbol{q}}_{\mathrm{NI}}^{(k)}$ correspond to the probability distributions in distillation loss in Equations (10) and (12), respectively. And, these distillation losses and the cross-entropy loss of ground-truth together join during the optimization in Section 4.5. The training of the difference evaluator is under the supervision of the final prediction performance.

Why Gumbel-Softmax? Compared to the conventional softmax function, Gumbel-Softmax [30] introduce a Gumbel distribution that enables random exploration among peers in the earlier training phase. This can help the training network obtain various knowledge from peers. With the increment of the training progress t, the softmax temperature $(1-t)$ is annealed and gradually select an expert peer with adequate difference as its teacher in the later training phase. Note that depicting the optimal response difference is still an open-ended problem. We believe that there exist other potential methods over ensemble multiple peer prediction networks. These do not contradict our motivation of decoupled progressive distillation within volume items, and may work together with our target and non-target item distillation.

Algorithm 1 Training Procedure of DePoD.

```
Input: the training set $(\boldsymbol{x}_m, \boldsymbol{y}_m) \in \mathcal{D}$, the peer number $K$, the current training epoch $\pi$ and the
    total epoch $\Pi$, the coefficients $\alpha$, $\gamma$ and $\beta$ for adjusting progressive distillation
Output: the parameters $\Theta$ of DePoD
1:  Randomly initialize all parameters $\Theta$ ;
2:  for each training batch $(\boldsymbol{x}_m, \boldsymbol{y}_m)$ do
3:      Update training progress $t = \pi/\Pi$
4:      Obtain $\boldsymbol{h}^{(k)}$ from encoders and compute $\boldsymbol{p}^{(k)}$, $\boldsymbol{p}_{\text{TI}}^{(k)}$, $\boldsymbol{q}_{\text{NI}}^{(k)}$ in Equations (1), (2) and (11)
5:      for each peer encoder $k$ do
6:          Select its teacher from peers according to Equation (18)
7:          Compute $\mathcal{L}_{\text{TI}}^{(k)}$ and $\mathcal{L}_{\text{NI}}^{(k)}$ according to Equations (10) and (12)
8:          Compute cross-entropy $\mathcal{L}_{\text{CE}}^{(k)}$ according to the ground-truth
9:      end for
10:     $\mathcal{L} \leftarrow \sum_{k=1}^{K}(\mathcal{L}_{\text{CE}}^{(k)} + \mathcal{L}_{\text{TI}}^{(k)} + \mathcal{L}_{\text{NI}}^{(k)})$
11:     Update all parameters to minimize $\mathcal{L}$;
12: end for
13: return $\Theta$
```

4.5 Joint Optimization

We jointly optimize the conventional cross-entropy loss of ground-truth and the progressive distillation loss as a holistic decoupled progressive distillation framework:

$$\min_{\Theta} \mathcal{L} = \sum_{k=1}^{K}(\mathcal{L}_{\text{CE}}^{(k)} + \mathcal{L}_{\text{DePoD}}^{(k)}) = \sum_{k=1}^{K}(\mathcal{L}_{\text{CE}}^{(k)} + \mathcal{L}_{\text{TI}}^{(k)} + \mathcal{L}_{\text{NI}}^{(k)}), \tag{19}$$

where K denotes the number of peer encoders and Θ is the parameters of the multi-peer prediction network and the difference evaluator. $\mathcal{L}_{\text{CE}}^{(k)} = \text{CE}(\boldsymbol{y} \| \boldsymbol{p}^{(k)})$ is the cross-entropy between the ground-truth and the prediction probability distribution of the kth peer.

The overall training procedure of DePoD is presented in Algorithm 1. The parameters are initialized in Line 1. For each epoch, the training progress will be updated in Line 3. The distinctive intent representations are inferred in Line 4. For each peer encoder, we first select the teacher in Line 6, and then conduct distillation in Line 7. The cross-entropy of ground-truth (Line 8) is jointly learned with distillation loss in Line 10. Finally, we update the network parameters in Line 11 and repeat the above steps until the last epoch.

Time Complexity. The trainable parameters of different modules are summarized in Table 2. To learn these parameters, it will take multi-round training for each training sample, including the forward propagation in multi-peer prediction networks and the computation of progressive distillation loss. Their time complexity is related to different hyper-parameters, including the length of sequence L, the dimension of intent representation d, the number of items $|\mathcal{I}|$, the number of peers K, the batch size $|\mathcal{B}|$ and the training progress t.

- **Multi-peer Prediction Networks**. Intent representation learning is the most time-consuming module that employs multi-layer sequential encoding to infer intent representation vectors. Since our DePoD is an encoder-agnostic learning framework, the time complexity depends on the specific sequence structures. For example, in terms of the representative Transformer [61], the time complexity is $O(L^2 \cdot d)$.

Table 2. The number of trainable parameters in different modules.

Module	Parameter number	Description
Sequence Embedding Module	$\|\mathcal{I}\| * d + L * d$	The embedding of items and their relative position information.
Intent Representation Learning Module	$\propto (K * d)$	The parameter number depends on a specific sequence encoder and it is proportional to peer number K and intent representation dimension d.
Next Item Prediction Module	$d * (d + 1) + \|\mathcal{I}\|$	The weights and bias of alignment and pre-softmax transformation.
Difference Evaluator	$d * (d + 1) * 2 + K * (K + 1)$	The parameters of computing query and key vectors and the nonlinear transformation of difference scores.

- **Target Item Distillation**. Target item distillation first requires calculating the probability distribution with respect to target item, where time complexity is $O(1)$. Then, the time complexity of finding excessively low confidence samples is $O(|\mathcal{B}|\log(|\mathcal{B}|))$. Finally, the computation of progressive target item loss is $O(1)$. Therefore, considering the above steps, the overall time complexity of target item distillation is approximately $O(|\mathcal{B}|\log(|\mathcal{B}|))$.
- **Non-target Item Distillation**. The time complexity of calculating the probability distribution with respect to non-target items is $O(|\mathcal{I}| - 1)$. Due to the construction of an increasing subset, the time complexity of sampling a non-target item is $O((1 - t) \cdot |\mathcal{I}|)$ that is related to the training progress. The computation of progressive non-target item loss is $O(|\mathcal{I}| - 1)$. Therefore, the overall time complexity of non-target item distillation is $O(|\mathcal{I}|)$.
- **Progressive Peer Selection**. The progressive peer selection employs the query-key matching to evaluate peers where the time complexity is $O(K^2 \cdot d)$. And, the time complexity of Gumbel-Softmax is $O(K)$. Therefore, the overall time complexity of progressive peer selection is $O(K^2 \cdot d)$.

Based on the above analysis, we can conclude that the time complexity in our progressive distillation is $O(|\mathcal{I}|)$, since sequential prediction task generally contains a large candidate item set. It's worth noting that this is a rough approximation, and the actual implementation details and optimizations can affect the practical running time.

5 EXPERIMENT

The proposed DePoD utilizes the decoupled progressive distillation among target and non-target items to enhance the modeling of interaction dynamics, which will optimize the learning of intent representations. To validate its effectiveness, we conduct extensive experiments to answer the following research questions:

- **RQ1:** What is the performance of our DePoD as compared to state-of-the-art sequential prediction methods?
- **RQ2:** Do different parts in our DePoD framework contribute to the prediction performance?
- **RQ3:** What is the effect of coefficients (i.e., α, γ, β) in DePoD when varying their values?
- **RQ4:** Does our framework DePoD facilitate the representation learning of next intent?

Table 3. Dataset statistics.

	Urban Event		Web Recommendation	
	NYC16	CHI18	Beauty	Toys
# Objects (regions/users)	3,229	2,692	22,363	19,412
# Items (events/behaviors)	440	246	12,101	11,924
# Interactions	473,887	264,314	198,502	167,597
Avg. interactions per object	146.76	98.18	8.88	8.63

5.1 Experimental Setup

5.1.1 Dataset. Considering the interaction dynamics widely exist in long or short sequences (avg. interactions per object), our proposed DePoD is evaluated on four real-world datasets covering urban and web spaces. The statistics across the datasets are shown in Table 3. The urban event datasets with relatively long sequence are prone to presenting multifarious intents (Sequence 1 in Figure 1), while the web recommendation datasets with less interactions may lack discriminative information (Sequence 2). And the real-world environment may impose noise interference on both of them (Sequence 3).

For the urban event datasets, NYC16[1] and CHI18[2] record the crime events of New York in 2016 and Chicago in 2018, respectively. These records mainly contain the fields of crime regions, time occurred, and event types. Following the work of [24], to deal with fine-grained prediction, we divide time into different slots at every 3 hours. And a simple event model [60] is employed to describe regions by using geographical information and events by using time slots and event types. The benchmark Amazon review datasets Beauty and Toys are from the work of [79], which is known for high sparsity.

For data preprocessing, we follow the common strategy in [10, 24, 55], and remove the inactive objects with fewer than five items. The last item of each object is used for testing, and the item before the last is a validation set. We set the maximum length of a sequence as 200 for urban event datasets (following [24]) and 100 for web recommendation datasets (following [10]). For the sequences beyond the maximum, we split them from right to left. And if the length is less than the maximum length, we add extra "padding" tokens to the right.

5.1.2 Evaluation Metrics. Following [55, 79], we evaluate the ranking performance by top-N Hit Ratio (HR@N), Top-N Normalized Discounted Cumulative Gain (NDCG@N), and Mean Reciprocal Rank (MRR). N is set as $\{1, 5, 10\}$ and HR@1 is equal to NDCG@1 when $N = 1$. All metrics are derived from the principle of the higher, the better. To speed up evaluation, we follow a common strategy in [55, 62, 64] and pair each ground-truth with 100 randomly sampled negative items that the object has not interacted with according to their popularity. The metrics for the process are defined as:

$$\begin{aligned} \text{NDCG@}N &= \frac{1}{|O|}\sum_{o\in O}\frac{\mathbf{1}(r_o \le N)}{\log_2(r_o+1)}, \\ \text{HR@}N &= \frac{1}{|O|}\sum_{o\in O}\mathbf{1}(r_o \le N), \\ \text{MRR} &= \frac{1}{|O|}\sum_{o\in O}\frac{\mathbf{1}(r_o \le N)}{r_o}. \end{aligned} \tag{20}$$

[1] https://data.cityofnewyork.us/browse?q=Arrest

[2] https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2

where $\mathbf{1}(\cdot)$ is an indicator function that returns 1 when the condition is satisfied, otherwise it is 0. And r_o is the rank of the ground-truth of object $o \in O$ based on its prediction probabilities.

5.1.3 Implementation Details. Our experimental environment mainly consists of NVIDIA TITAN XP GPU with 12G memory, Python 3.8 and Pytorch 1.10.0. The proposed method[3] is trained from scratch by using Adam optimizer with linear decay and a batch size of 256 for 100 training epochs.

For sequence encoders, we integrate various sequential prediction methods with different sequence structures. The Transformer-based BERT4Rec [55] shows the best performance. Therefore, we mainly tune the hyper-parameters and report the results under the integration of BERT4Rec. Specifically, we mask 15% items in the historical sequence and perform both MLM and NIP in the training phase by following BERT4Rec. We set the number of multi-head layers as 2, the number of heads as 2, the dimension of embedding as 64 and the dimension of the intermediate layer as 256. The threshold of learning progress τ_0 and τ_1 are set as 0.2 and 0.7, respectively to ensure sufficient training time for different phases, and the truncation proportion ε is set as 0.01. For the coefficients in our DePoD, α, γ, and β are tuned through grid search on the validation set from $\{1, 3, 5, 7, 9\}$, $\{0, 0.5, 1.0, 1.5, 2.0\}$ and $\{1, 3, 5, 7, 9\}$, respectively. After the grid search, the final coefficient groups (α, γ, β) are $(5, 1, 1)$, $(3, 1, 3)$, $(5, 1.5, 7)$ and $(3, 0.5, 5)$ for NYC16, CHI18, Beauty and Toys, respectively.

5.1.4 Baselines. To evaluate the effectiveness of our DePoD, we compare it with 12 representative works related to interaction dynamics, including novel urban event prediction methods [22, 24, 78], sequential recommendation methods [10, 55, 64, 66, 79], deep ensemble [81] and distillation methods [24, 31, 74, 75]. We reproduce R-CE [66], Bagging [81], DML [74] and DKD [75] based on BERT4Rec, and implement other methods by utilizing the codes provided by the authors. The results of all methods are reported under the optimal hyper-parameter settings in our experiments.

Urban Event Prediction Methods:

- **DuroNet** [22]: A noise-robust urban event model for predicting crime counts. To adapt it to our task, we replace its objectives with cross-entropy to achieve event item classification.
- **Informer** [78]: A long sequence time-series forecasting method that solves dynamics by exploiting long-term information. Similar to DuroNet, we also reformulate its loss function to adapt for the classification problem.
- **HAIL** [24]: A novel distillation based sequential prediction framework that employs mutual exclusivity knowledge from peers to address implicitly hard interactions caused by dynamics.

Sequential Recommendation Methods:

- **BERT4Rec** [55]: BERT4Rec exploits bi-directional information for users' behavior prediction. If the number of peer encoders is set to 1, our proposed DePoD will degrade to BERT4Rec.
- **R-CE** [66]: A denoising implicit feedback strategy for recommendation. We apply this strategy for BERT4Rec as a baseline of weakening the contribution of dynamic interaction samples.
- **S^3-Rec** [79]: A mutual information maximization-based sequential recommendation method that aims to amplify the intrinsic data correlation. Since extra attribute information is not employed in our work, we just adopt its MIP and SP object functions for fair comparison.
- **HyperRec** [64] HyperRec employs multiple sequential hypergraphs to model dynamic preferences of users.
- **STOSA** [10] A novel distribution based sequential recommendation method that incorporates dynamic uncertainty into the modeling of item transitions.

[3] https://github.com/hukx-issac/DePoD

Deep Ensemble and Distillation Methods:

- **Bagging** [81]: A simple bagging based method. We apply this method for five BERT4Rec models where each model is trained with 70% randomly sampled training data. The output probability distribution is the average of all models.
- **DML** [74]: An online mutual distillation method that directly learns the probability distributions between two peers. We integrate this distillation manner with BERT4Rec.
- **BiCAT** [31]: A novel self-knowledge distillation based sequential recommendation method that employs augmented and original sequences to enhance intent representations.
- **DKD** [75]: A novel decoupled knowledge distillation method. Our DePoD will degrade to it after removing the decoupled progressive distillation strategy.

5.2 Overall Performance Comparison (RQ1)

We report the performances of two kinds of DePoD according to the number of peer encoders:

- **DePoD**: This is the primary version of DePoD that only adopts two peer encoders (K=2), which makes a fair comparison against baselines in limited GPU memory. We present the metrics of both peers for a comparison between their performance difference. With the model setting described in Section 5.1.3, the number of trainable parameters is about 0.2 million in the intent representation learning module.
- **DePoD(multi)**: This is the full version of DePoD where the number of peer encoders is five (K=5), which incurs more computation resource. We set a small batch size 64 to save more network parameters in limited GPU memory, which is different from our default setting 256. For conciseness, we report the average of all peers in each metric. The number of trainable parameters is about 0.5 million in the intent representation learning module while the number in the progressive selection module is about 8.35 thousands.

The performance and running efficiency of different methods on urban event and web recommendation datasets are presented in Table 4 and Table 5, respectively. Note that although we adopt multiple peer encoders in the training phase, we can just keep any one peer in the testing phase, since KD makes them produce similar performances. In conclusion, we have the following findings:

5.2.1 ***The Advantage of DePoD Over All Baselines.*** From Table 4 and Table 5, we can observe that with few exceptions, HAIL and STOSA are the strongest baselines in urban event and web recommendation datasets, respectively. Despite their success in specific scenarios, our proposed DePoD consistently outperforms all baselines. And, the full version DePoD(multi) can further amplify the performance advantage in most metrics. Especially in the largest Beauty dataset, DePoD(multi) achieves 9.83%, 12.05% and 12.14% improvements in terms of NDCG@5, HR@5 and MRR, respectively. These observations demonstrate that our DePoD can effectively model different types of dynamics of interaction intents by progressively exploiting the distinctive model responses over volume candidate items.

Compared to Informer, BERT4Rec and S^3-Rec, DePoD employs the distinctive model responses between multiple prediction networks to optimize the learning of intent representations. Compared to DuroNet and R-CE, the main difference of DePoD lies in the distillation of non-target items which provide knowledge among non-target items. Compared to STOSA, the superiority of DePoD indicates distribution-based representation is less competitive than multiple vector-based representation, especially for long sequences, which is in line with findings in [12, 18, 53]. The Bagging method focuses on result fusion while incurring more computation resource in both training and testing phases. Compared to Bagging, distillation-based methods (e.g., HAIL, DML, BiCAT and our DePoD) perform knowledge fusion and achieve comparable or even better performance with higher iteration speed and less running memory. Compared to HAIL, HyperRec, DML and BiCAT,

Table 4. The performance comparison on the urban event datasets. The numbers in **Bold** and Underline denote the best and sub-optimal results, respectively. The row “Improv.” means the relative improvement of our best result over the best baseline result. The row “p-value” refers to the significance level in t-test by comparing the five results of DePoD(multi) with the average of DePoD. The columns “speed” and “memory” indicate the iteration speed and GPU memory cost in the training (or testing) phase.

Method	*NYC16*							
	NDCG@1	NDCG@5	NDCG@10	HR@5	HR@10	MRR	speed(iter/s)	memory(GB)
DuroNet [22]	0.0622	0.1777	0.2270	0.2926	0.4460	0.1833	~9.3(12.4)	~3.3(1.6)
Informer [78]	0.1059	0.2108	0.2540	0.3165	0.4500	0.2160	~1.1(12.8)	~9.6(1.8)
HAIL [24]	0.3140	0.4131	0.4461	0.5070	0.6141	0.4113	~6.2(12.3)	~3.3(0.7)
BERT4Rec [55]	0.2870	0.3927	0.4267	0.4908	0.5955	0.3901	~14.5(12.3)	~2.4(0.8)
R-CE [66]	0.2471	0.3301	0.3599	0.3986	0.4921	0.3375	~5.3(12.3)	~2.3(0.8)
S^3-Rec [79]	0.2533	0.3315	0.3681	0.4103	0.5240	0.3391	~3.8(6.0)	~6.2(2.8)
HyperRec [64]	0.0790	0.1266	0.1418	0.1682	0.2159	0.1424	~4.5(12.9)	~2.9(0.9)
STOSA [10]	0.2725	0.3713	0.4143	0.4642	0.5974	0.3727	~4.9(6.2)	~2.8(1.7)
Bagging [81]	0.2982	0.4022	0.4346	0.5050	0.6050	0.3983	~5.1(1.8)	~11.2(1.2)
DML [74]	0.2927	0.3997	0.4317	0.5005	0.5999	0.3949	~6.2(12.1)	~3.3(1.1)
BiCAT [31]	0.2942	0.4120	0.4554	0.5109	0.6107	0.4083	~5.3(12.0)	~8.5(0.7)
DKD [75]	0.3032	0.4083	0.4419	0.5042	0.6079	0.4059	~6.1(12.1)	~3.4(1.2)
DePoD (Peer 1)	0.3311	0.4335	0.4642	0.5290	0.6240	0.4297	~6.4(12.3)	~3.4(1.2)
DePoD (Peer 2)	0.3311	0.4337	0.4637	0.5302	0.6228	0.4295		
Avg. DePoD	0.3311	0.4336	0.4640	0.5296	0.6234	0.4296		
Avg. DePoD(multi)	**0.3357**	**0.4418**	**0.4729**	**0.5389**	**0.6352**	**0.4370**	~7.8(12.3)	~2.5(1.2)
p-value	6.17e-04	1.18e-04	4.45e-06	1.22e-02	6.46e-05	2.89e-05	-	-
Improv.	+6.91%	+6.95%	+3.84%	+5.48%	+3.44%	+6.25%	-	-
Method	*CHI18*							
	NDCG@1	NDCG@5	NDCG@10	HR@5	HR@10	MRR	speed(iter/s)	memory(GB)
DuroNet [22]	0.0676	0.1757	0.2236	0.2923	0.4421	0.1802	~9.8(10.6)	~3.3(1.5)
Informer [78]	0.1174	0.2200	0.2660	0.3228	0.4650	0.2259	~1.9(12.4)	~9.6(1.8)
HAIL [24]	0.3584	0.4684	0.5026	0.5683	0.6742	0.4631	~6.2(14.3)	~3.3(0.7)
BERT4Rec [55]	0.3491	0.4619	0.4965	0.5638	0.6712	0.4562	~13.7(14.3)	~2.3(0.7)
R-CE [66]	0.2426	0.3127	0.3450	0.3822	0.4832	0.3240	~7.8(14.3)	~2.3(0.7)
S^3-Rec [79]	0.3132	0.3923	0.4293	0.4736	0.5888	0.3978	~3.5(6.5)	~6.2(2.8)
HyperRec [64]	0.0353	0.0776	0.1025	0.1319	0.2058	0.0893	~4.8(14.1)	~2.4(0.9)
STOSA [10]	0.2975	0.4090	0.4505	0.5126	0.6415	0.4050	~5.3(7.15)	~2.8(1.7)
Bagging [81]	0.3184	0.4367	0.4746	0.5431	0.6605	0.4312	~5.0(1.2)	~11.0(1.2)
DML [74]	0.3510	0.4645	0.4992	0.5654	0.6731	0.4588	~6.3(14.5)	~3.2(1.2)
BiCAT [31]	0.3247	0.4192	0.4610	0.5108	0.6400	0.4218	~5.4(12.2)	~8.8(0.7)
DKD [75]	0.3536	0.4683	0.5004	0.5717	0.6716	0.4610	~6.2(14.5)	~3.3(1.2)
DePoD (Peer 1)	0.3915	0.5046	0.5363	**0.6059**	0.7036	0.4969	~6.3(14.3)	~3.3(1.2)
DePoD (Peer 2)	0.3908	0.5039	0.5361	0.6048	0.7039	0.4965		
Avg. DePoD	0.3912	0.5043	0.5362	0.6054	0.7038	0.4967		
Avg. DePoD(multi)	**0.3974**	**0.5061**	**0.5381**	0.6056	**0.7042**	**0.4990**	~7.7(14.3)	~2.4(1.2)
p-value	1.09e-06	8.10e-04	1.28e-03	0.50	0.15	2.21e-05		
Improv.	+10.88%	+8.05%	+7.06%	+5.98%	+4.45%	+7.75%	-	-

DePoD delves into different effects of target and non-target item distillation. And, similar with the augmented sequences in BiCAT, the MLM training adopted in our DePoD can also employ more samples to enhance the learning of intent representations. Compared to DKD, we further develop a decoupled progressive distillation strategy to help DePoD achieve better performance.

Table 5. The performance comparison on the web recommendation datasets. The numbers in **Bold** and Underline denote the best and sub-optimal results, respectively. The row "Improv." means the relative improvement of our best result over the best baseline result. The row "p-value" refers to the significance level in a t-test by comparing the five results of DePoD(multi) with the average of DePoD. The columns "speed" and "memory" indicates the iteration speed and GPU memory cost in the training (or testing) phase.

Method	*Beauty*							
	NDCG@1	NDCG@5	NDCG@10	HR@5	HR@10	MRR	speed(iter/s)	memory(GB)
DuroNet [22]	0.0678	0.1291	0.1544	0.1881	0.2665	0.1398	~14.4(29.3)	~1.6(0.9)
Informer [78]	0.0140	0.0393	0.0564	0.0650	0.1188	0.0607	~1.4(23.1)	~5.2(1.1)
HAIL [24]	0.1031	0.1779	0.2070	0.2488	0.3396	0.1859	~4.7(22.3)	~6.2(2.2)
BERT4Rec [55]	0.0953	0.1599	0.1862	0.2207	0.3025	0.1701	~7.4(22.3)	~3.9(2.3)
R-CE [66]	0.0138	0.0351	0.0496	0.0573	0.1025	0.0553	~15.7(22.3)	~3.9(2.3)
S^3-Rec [79]	0.0705	0.1534	0.1929	0.2344	0.3589	0.1636	~7.5(20.7)	~2.7(1.5)
HyperRec [64]	0.0440	0.0957	0.1231	0.1470	0.2326	0.1122	~14.3(34.9)	~5.5(4.9)
STOSA [10]	0.1172	0.1892	0.2173	0.2573	0.3444	0.1919	~16.1(31.0)	~1.6(1.2)
Bagging [81]	0.0964	0.1827	0.2183	0.2548	0.3649	0.1913	~3.6(1.5)	~14.0(3.7)
DML [74]	0.0963	0.1659	0.2025	0.2469	0.3607	0.1758	~4.1(21.6)	~7.5(1.7)
BiCAT [31]	0.0825	0.1714	0.2075	0.2571	0.3690	0.1777	~16.9(23.1)	~2.7(0.7)
DKD [75]	0.1013	0.1878	0.2129	0.2495	0.3651	0.1808	~3.9(21.5)	~7.6(1.7)
DePoD (Peer 1)	0.1194	0.2008	0.2314	0.2773	0.3720	0.2078	~4.0(22.3)	~8.1(1.7)
DePoD (Peer 2)	0.1183	0.2011	0.2310	0.2794	0.3723	0.2072		
Avg. DePoD	0.1189	0.2010	0.2312	0.2784	0.3722	0.2075		
Avg. DePoD(multi)	**0.1232**	**0.2078**	**0.2420**	**0.2883**	**0.3943**	**0.2152**	~3.9(22.3)	~8.4(1.7)
p-value	2.85e-02	1.37e-04	1.59e-05	2.72e-04	3.09e-06	3.27e-04	-	-
Improv.	+5.12%	+9.83%	+11.37%	+12.05%	+6.86%	+12.14%	-	-

Method	*Toys*							
	NDCG@1	NDCG@5	NDCG@10	HR@5	HR@10	MRR	speed(iter/s)	memory(GB)
DuroNet [22]	0.0298	0.0688	0.0958	0.1087	0.1931	0.0894	~13.6(25.3)	~1.6(0.9)
Informer [78]	0.0128	0.0358	0.0535	0.0593	0.1148	0.0585	~1.4(23.0)	~5.2(1.1)
HAIL [24]	0.1158	0.1894	0.2181	0.2583	0.3476	0.1984	~4.5(20.2)	~6.1(2.2)
BERT4Rec [55]	0.0935	0.1783	0.2134	0.2594	0.3685	0.1874	~8.7(20.2)	~3.9(2.3)
R-CE [66]	0.0157	0.0292	0.0399	0.0429	0.0763	0.0494	~13.4(20.2)	~3.9(2.3)
S^3-Rec [79]	0.0610	0.1402	0.1800	0.2190	0.3424	0.1534	~7.4(22.4)	~2.8(1.5)
HyperRec [64]	0.0813	0.1464	0.1740	0.2090	0.2949	0.1584	~14.5(31.8)	~5.5(5.0)
STOSA [10]	0.1299	0.2026	0.2295	0.2710	0.3544	0.2052	~17.3(29.5)	~1.6(1.2)
Bagging [81]	0.0973	0.1817	0.2159	0.2616	0.3672	0.1904	~ 3.5(1.6)	~14.5(3.6)
DML [74]	0.0993	0.1832	0.2169	0.2618	0.3691	0.1919	~4.1(20.0)	~7.5(1.7)
BiCAT [31]	0.0803	0.1618	0.1983	0.2412	0.3543	0.1714	~16.7(22.8)	~2.7(0.7)
DKD [75]	0.1007	0.1821	0.2170	0.2622	0.3705	0.1916	~4.0(19.8)	~7.5(1.7)
DePoD (Peer 1)	0.1303	0.2190	0.2518	0.3032	0.3999	0.2248	~4.1(20.2)	~8.0(1.7)
DePoD (Peer 2)	0.1299	0.2167	0.2498	0.2992	0.4048	0.2234		
Avg. DePoD	0.1301	0.2179	0.2508	0.3012	**0.4024**	0.2241		
Avg. DePoD(multi)	**0.1402**	**0.2239**	**0.2551**	**0.3024**	0.3995	**0.2303**	~3.9(20.2)	~8.3(1.7)
p-value	8.19e-06	4.73e-05	1.38e-05	3.57e-02	1.78e-03	1.72e-05		
Improv.	+7.91%	+10.51%	+11.15%	+11.88%	+9.26%	+12.23%	-	-

5.2.2 ***Small Performance Difference Between Peers.*** DePoD consists of multiple peer encoders and each of them will output a prediction result. It is a problem to decide which one will be employed in the testing stage. Fortunately, we find that the performance difference is marginal between the Peer 1 and Peer 2, since KD makes them learn from each other. This means the selection of encoders can be random and take less effect in deployment. More observations about performance difference can be found in Section 5.4.1 and Section 5.6.2.

5.2.3 ***Higher Prediction Accuracy in More Peers.*** Compared to the primary DePoD, we find that with few exceptions, DePoD(multi) with a small batch size achieves better performance in most metrics. And, the p-value is less than 0.05 or even 0.01, indicting significance in performance improvement. In particular, the values of NDCG@1 between DePoD and STOSA are close in Toys and Beauty, while DePoD(multi) shows better performance in NDCG@1. This indicates that employing the diversity of more peers can further boost prediction performance, while their computation cost increases accordingly.

5.2.4 ***Effect of Item Number in Urban and Web Datasets.*** Compared with DePoD, DePoD(multi) exhibited better improvements in web recommendation datasets than that in urban event datasets. And, in terms of significance level, we can also obverse some exceptions in the HR@{5,10} in CHI18. This is because the item number in urban event datasets is less than those in web recommendation datasets, which reduces the difficulty of modeling dynamics of interaction intents. And, hit ratio is a relatively simple metric that only considers the number of correct responses without their ranking positions. It demonstrates a relationship between the number of peers and items whereby more items will increase interaction dynamics and require more peers to capture diverse intents.

5.2.5 ***Comparable Efficiency in Deployment***. The last two columns of Table 4 and Table 5 shows the iteration speed and memory of different methods, respectively. Due to the distillation among multiple peers, our proposed DePoD needs to compute more parameters and gradients. As such, DePoD results in relatively low iteration speed and high memory occupation compared to most baselines in the training phase. Nevertheless, the iteration speed of DePoD still outperforms several vector-based methods (e.g., HAIL, DKD, HyperRec) and the distribution-based STOSA method in urban event datasets with long sequences. What is more, if we keep only one peer encoder in the testing phase, the efficiency of DePoD can be boosted higher, and achieve comparable speed and memory with the baseline methods. In real-work applications, it is generally acceptable to obtain better performance by employing more computation resources in the training phase. Our proposed DePoD does not bring much burden in the testing phase or deployment.

5.3 Ablation Study (RQ2)

To investigate the plausibility of our proposed target and non-target item distillation, we set four variants of DePoD with two peer encoders:

- $\neg$TI^{+}: This variant removes the deliberate practice of target item in the later training stage, i.e., $\mathcal{L}_{\mathrm{DP}}^{(k)}$ in Equation (10), and retains $\mathcal{L}_{\mathrm{One}}^{(k)}$ during the whole training process.
- $\neg$TI: This variant removes the target item distillation and only adopts non-target item distillation with progressive size, i.e., $\mathcal{L} = \sum_{k=1}^{K}(\mathcal{L}_{\mathrm{CE}}^{(k)} + \mathcal{L}_{\mathrm{NI}}^{(k)})$
- $\neg$NI^{+}: This variant removes the progressive size setting in Equation (11) and just adopts the vanilla non-target item distillation, i.e., term ④, in Equation (4).
- $\neg$NI: This variant removes the non-target item distillation and only adopts target item distillation with progressive difficulty, i.e., $\mathcal{L} = \sum_{k=1}^{K}(\mathcal{L}_{\mathrm{CE}}^{(k)} + \mathcal{L}_{\mathrm{TI}}^{(k)})$.
- $\neg$PE: This variant removes the trainable matrix $\boldsymbol{S}$ that encodes the relative position information of sequential items.

The average results of different peers in terms of NDCG@5 are reported in Figure 4. Moreover, we compare the proposed progressive peer selection with several representative peer ensemble

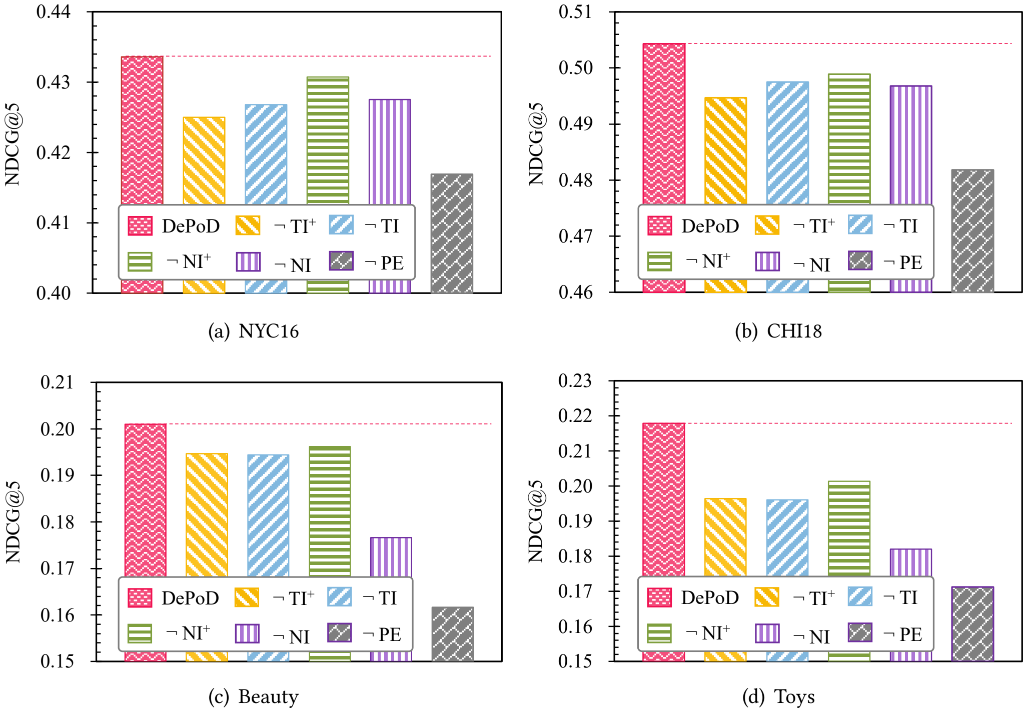


Fig. 4. Ablation study of our DePoD on four datasets (NDCG@5). "¬" indicates that the corresponding objective is removed in the training phase, while the remaining objectives are kept. "+" denotes our decoupled progressive distillation.

strategies (e.g., attention, direct selection) in Table 6 and validate its effectiveness with a different number of peers in Figure 5. In summary, we can make the following observations:

5.3.1 ***Effectiveness of Target Item Distillation with Progressive Difficulty***. From the variant ¬TI$^+$ (yellow bar) of Figure 4, we can observe a clear performance degradation after removing deliberate practice in the later training phase. And this degradation is more significant in web recommendation datasets with relatively large candidate items (about 3.24% in Beauty and 10.94% in Toys). Moreover, by comparing ¬TI$^+$ and ¬TI (blue bar), we can see that their performance difference is marginal in the web recommendation datasets, while $\mathcal{L}_{\text{One}}^{(k)}$ in the earlier training phase shows negative effect on urban event datasets. The above observations demonstrate that the contribution of vanilla target item distillation $\mathcal{L}_{\text{One}}^{(k)}$ is limited in most cases, which is in line with the previous work [75]. Importantly, our proposed deliberate practice in target item distillation can effectively improve performance by focusing on the potential low-confidence samples.

5.3.2 ***Effectiveness of Non-target Item Distillation with Progressive Size***. By comparing the variant ¬NI$^+$ (green bar), we find that DePoD consistently outperforms ¬NI$^+$ on all datasets. This observation clearly indicates the effectiveness of our proposed decoupled progressive distillation strategy within non-target items. Moreover, when further removing the whole non-target item distillation, the performance of ¬NI (purple bar) degrades, especially in web recommendation datasets. The above observations reveal that the current approach is not the optimal way to directly

Table 6. Average Performance of different peer ensemble strategy with five peer encoders. "↓" refers to performance degradation when comparing the means of two peer encoders.

Dataset		Progressive Peer Selection			Attention [82]			Selection w/ Large Difference			Selection w/ Small Difference		
		NDCG5	HR@5	MRR	NDCG5	HR@5	MRR	NDCG5	HR@5	MRR	NDCG5	HR@5	MRR
Urban Event	NYC16	**0.4418**	**0.5389**	**0.4370**	0.4354	0.5348	0.4304	0.4349	0.5323	0.4304	0.4343	0.5317	0.4297
	CHI18	**0.5061**	**0.6056**	**0.4990**	0.4712↓	0.5768↓	0.4638↓	0.4728↓	0.5813↓	0.4640↓	0.4728↓	0.5779↓	0.4656↓
Web Recommendation	Beauty	**0.2078**	**0.2883**	**0.2152**	0.1982↓	0.2783	0.2061↓	0.2043	0.2846	0.2114	0.2030	0.2831	0.2105
	Toys	**0.2239**	**0.3024**	**0.2303**	0.2147↓	0.2930↓	0.2213↓	0.2191	0.2955↓	0.2262	0.2207	0.2988↓	0.2273

expose the global knowledge among non-target items, and our decoupled progressive distillation strategy can effectively ease this problem.

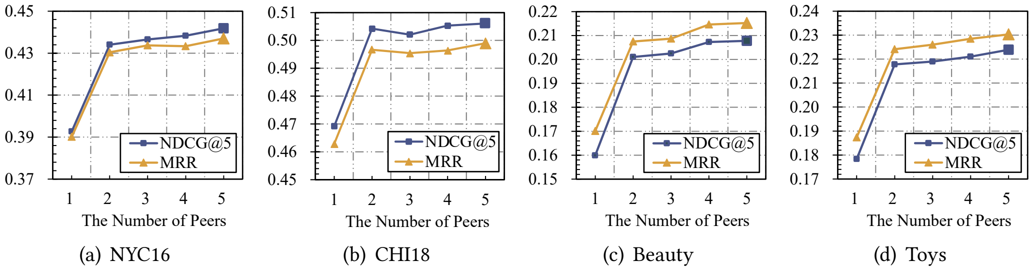
5.3.3 ***Varying Importance between Target and Non-target Item Distillation***. To further investigate the importance of target and non-target item distillation, we compare the performance among variants ¬TI, ¬NI and the original framework DePoD (red bar). We find that both target and non-target item distillation take positive effect on prediction performance. In particular, non-target item distillation is more important than target item in web recommendation and shows comparable contribution with target item distillation in urban event prediction. This is because the volume of items vary with different scenarios.

5.3.4 ***Effectiveness of Relative Position Information***. After removing the relative position information, we find that the prediction performance of the variant ¬PE (gray bar) shows significant degradation. And, the value of its NDCG@5 metrics is also lower than other variants. This observation indicates that relative position information is the foundation of our sequential prediction task. It models the order of items within a historical sequence and further affects the inference of next intent. Therefore, to better capture the dynamics of interaction intents, relative position information is crucial for our decoupled progressive distillation.

5.3.5 ***Effectiveness of Progressive Peer Selection***. The proposed progressive peer selection aims to gradually find a teacher with adequate difference, transferring informative knowledge. To validate its effectiveness, we investigate it from the following three aspects:

Selection vs. Attention. The proposed progressive peer selection can degrade to an attention-based ensemble manner [82] after removing the training progress t in Equation (17). From Table 6, we find that the improvement of attention-based ensemble manner is limited and even degrades the prediction performance. Notably, our proposed progressive peer selection consistently outperforms the attention-based ensemble manner and boosts the prediction performance. In line with the work of [74], it indicates that the attention-based ensemble manner can produce a powerful teacher with high posterior probabilities at the ground-truth item, which reduces the diversity of knowledge among items and contradicts the objective of online distillation. Our progressive peer selection can avoid weighting average and exploit informative knowledge by gradually selecting one adequate peer in the later training phase.

Progressive Selection vs. Direct Selection. Besides progressive peer selection, the teacher is also directly selected according to the cosine distances between their intent representations $\boldsymbol{h}^{(k)}$. In this setting, a large cosine distance refers to a large difference between the training network and the corresponding peer, and vice versa. From Table 6, we can observe that the proposed DePoD with progressive peer selection consistently outperforms the selection with a large difference and a small difference. This demonstrate our decoupled progressive distillation requires an adequate difference, not larger or smaller.



(a) NYC16 (b) CHI18 (c) Beauty (d) Toys

Fig. 5. Average Performance trend of DePoD with the growing number of peers.

Effect of the Number of Peers. From Figure 5, we observe that the performance of our proposed DePoD is improved by increasing the numbers of peers. In particular, we notice the margin between two peers and one peer is larger than other situations. And the difference between four peers and five peers is relatively slight, indicating an upper limit for peers. This observation is helpful to select a good trade-off between the prediction performance and training costs according to the practical computation conditions.

5.4 Further Probing

5.4.1 ***Integration with Different Sequence Encoders.*** The proposed DePoD is a general framework that can integrate various sequential prediction methods as encoders. To validate the flexibility of our DePoD, we attempt the following representative sequential prediction methods with different sequence structures:

- Caser [57]: This is a CNN-based sequential prediction method that models a sequence as an "image" in the time. It embeds the whole sequence into an intent presentation vector, which cannot further be enhanced by MLM.
- GRU4Rec [20]: It is a conventional GRU-based sequential prediction method that just performs NIP training in the original setting.
- GRU4Rec*: Based on GRU4Rec, we further apply both MLM and NIP in the process of model training.
- BERT4Rec [55]: This is a popular Transformer-based sequential method. It adopts both MLM and NIP in the training phase, which is our default setting.

Table 7 shows the results of different combinations of the above methods. We have the following findings: (1) Comparing with a solo method, DePoD shows significant performance improvement when combining identical methods into a same framework. In particular, GRU4Rec*+GRU4Rec* even shows superior performance than a solo BERT4Rec in the NYC16 dataset. (2) MLM shows great benefits for our DePoD, since it generates more samples to train the prediction framework with decoupled progressive distillation. For example, GRU4Rec+GRU4Rec only shows 6.32% improvement over GRU4Rec in Beauty in terms of NDCG@5, while the relative improvement is 23.87% by comparing GRU4Rec*+GRU4Rec* and GRU4Rec*. (3) When combining different sequence structures, DePoD reduces their performance difference. We further compare the results of different structures with the average of combining identical structures, and find they almost fail to obtain better improvement (this observation is in line with [74]). In most situations, the stronger method shows performance degradation (↓), while the weaker method obtains performance gains (↑). Some obvious exceptions are mainly in the large web recommendation datasets without MLM. This indicates the importance of MLM for our DePoD, and large diversity within different structures may not bring

Table 7. Performance comparison of our proposed DePoD in terms of two peer encoders when combining different sequential prediction methods. "↑" and "↓" denote the performance variation when combining different methods into a same framework, compared to the average of combining identical methods.

Task	Encoder	NYC16			CHI18			Beauty			Toys		
		NDCG5	HR@5	MRR	NDCG5	HR@5	MRR	NDCG5	HR@5	MRR	NDCG5	HR@5	MRR
NIP	Caser	0.3093	0.4007	0.3170	0.2965	0.3770	0.3093	0.1049	0.1572	0.1200	0.0677	0.1061	0.0859
	Caser	0.3265	0.4162	0.3292	0.3178	0.4094	0.3244	0.1143	0.1715	0.1278	0.0738	0.1150	0.0916
	+Caser	0.3270	0.4168	0.3295	0.3181	0.4108	0.3243	0.1143	0.1678	0.1286	0.0730	0.1132	0.0913
	Avg.	0.3278	0.4165	0.3294	0.3180	0.4101	0.3244	0.1143	0.1697	0.1282	0.0734	0.1141	0.0915
	GRU4Rec	0.2614	0.3465	0.2730	0.2932	0.3856	0.2989	0.0917	0.1423	0.1060	0.0453	0.0768	0.0674
	GRU4Rec	0.2881	0.3669	0.2964	0.2998	0.3881	0.3066	0.0969	0.1467	0.1123	0.0701	0.1119	0.0873
	+GRU4Rec	0.2885	0.3676	0.2966	0.2998	0.3881	0.3066	0.0980	0.1481	0.1131	0.0698	0.1106	0.0873
	Avg.	0.2883	0.3673	0.2965	0.2998	0.3881	0.3066	0.0975	0.1474	0.1127	0.0700	0.1113	0.0873
	Caser	0.2921↓	0.3719↓	0.2995↓	0.3026↓	0.3908↓	0.3098↓	0.0642↓	0.1044↓	0.0840↓	0.0563↓	0.0939↓	0.0764↓
	+GRU4Rec	0.2919↑	0.3716↑	0.2995↑	0.3025↑	0.3930↑	0.3077↑	0.0647↓	0.1048↓	0.0842↓	0.0557↓	0.0922↓	0.0761↓
MLM+NIP	GRU4Rec*	0.3549	0.4528	0.3530	0.4467	0.5475	0.4422	0.1282	0.1848	0.1413	0.1154	0.1777	0.1293
	GRU4Rec*	0.4043	0.5014	0.4011	0.4606	0.5613	0.4552	0.1589	0.2236	0.1696	0.1621	0.2286	0.1733
	+GRU4Rec*	0.4037	0.5005	0.4009	0.4608	0.5617	0.4552	0.1587	0.2243	0.1690	0.1608	0.2266	0.1725
	Avg.	0.4040	0.5009	0.4010	0.4607	0.5615	0.4552	0.1588	0.2240	0.1693	0.1615	0.2276	0.1729
	BERT4Rec	0.3927	0.4908	0.3901	0.4619	0.5638	0.4562	0.1599	0.2207	0.1701	0.1783	0.2594	0.1874
	BERT4Rec	0.4335	0.5290	0.4297	0.5046	0.6059	0.4969	0.2008	0.2773	0.2078	0.2190	0.3032	0.2248
	+BERT4Rec	0.4337	0.5302	0.4295	0.5039	0.6048	0.4965	0.2011	0.2794	0.2072	0.2167	0.2992	0.2234
	Avg.	**0.4336**	**0.5296**	**0.4296**	**0.5043**	**0.6054**	**0.4967**	**0.2010**	**0.2784**	**0.2075**	**0.2179**	**0.3012**	**0.2241**
	GRU4Rec*	0.4033↓	0.4992↓	0.4008↓	0.4720↑	0.5758↑	0.4643↑	0.1942↑	0.2690↑	0.2009↑	0.2060↑	0.2797↑	0.2135↑
	+BERT4Rec	0.4259↓	0.5240↓	0.4215↓	0.4779↓	0.5836↓	0.4693↓	0.1946↓	0.2687↓	0.2020↓	0.2074↓	0.2811↓	0.2151↓

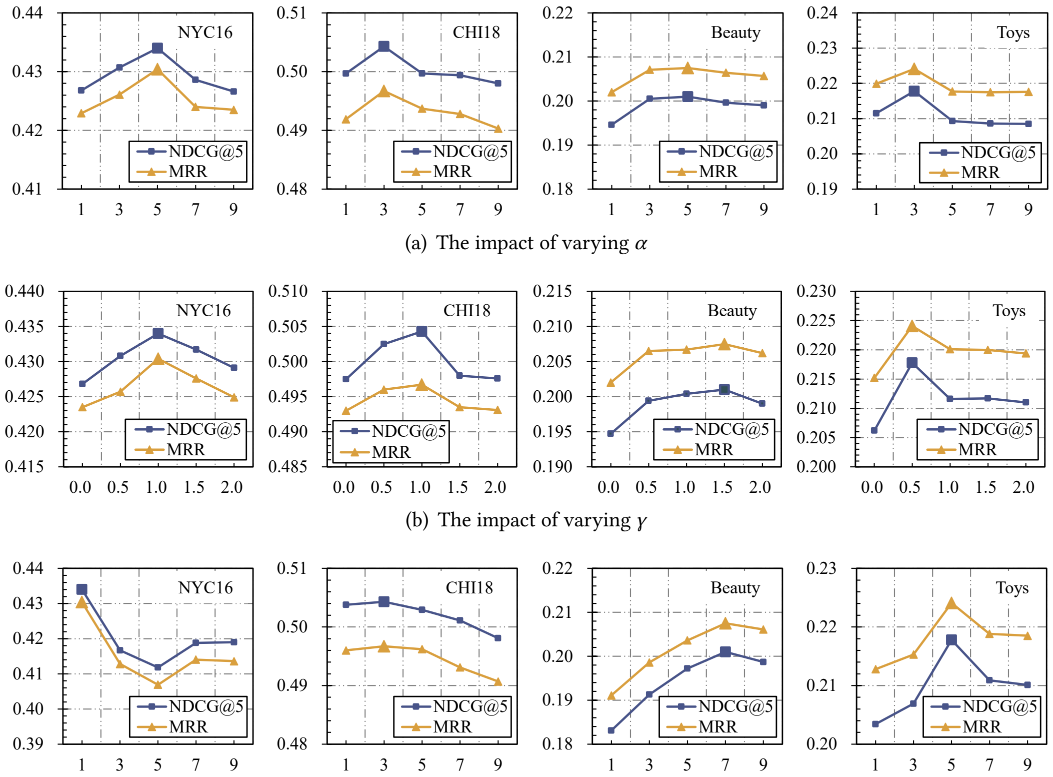
Table 8. Average Performance of different masking manners in our non-target item distillation with progressive size. "Uniform" and "Frequency" refer to sample non-target items according to the uniform distribution or the frequency-based distribution, respectively. "Masking" (or "Non-masking") denotes that the sampled non-target items are set to 1 (or 0) in the masking vector $\boldsymbol{c}$, while the other values are set as 0 (or 1).

Dataset		Frequency, Non-masking			Uniform, Masking			Frequency, Masking		
		NDCG5	HR@5	MRR	NDCG5	HR@5	MRR	NDCG5	HR@5	MRR
Urban Event	NYC16	**0.4336**	0.5296	**0.4296**	0.4296	**0.5301**	0.4238	0.4299	0.5290	0.4247
	CHI18	0.5043	0.6054	0.4967	**0.5053**	**0.6064**	**0.4973**	0.4993	0.6003	0.4924
Web Recommendation	Beauty	0.1972	0.2744	0.2039	0.1981	0.2755	0.2050	**0.2010**	**0.2784**	**0.2075**
	Toys	0.2028	0.2773	0.2111	0.2004	0.2752	0.2091	**0.2179**	**0.3012**	**0.2241**

adequate response difference. (4) BERT4Rec+BERT4Rec consistently achieves the best performance, and the GRU4Rec*+BERT4Rec is the sub-optimal model. The above observations demonstrate the effectiveness and flexibility of our DePoD. Moreover, in line with [45, 74], a large response difference caused by different sequence structures (e.g., Caser+GRU4Rec, GRU4Rec*+BERT4Rec) may not beneficial for training models to converge to a more robust minima.

5.4.2 *How to Mask Non-target Items?*

In Section 4.3, non-target item distillation with progressive size gradually introduces the knowledge among non-target items, which needs to sample non-target items and mask them in the vector $\boldsymbol{c}$. To further investigate the effect of item frequency, we consider sampling masked non-target items based on the uniform distribution, or a distribution that is in direct proportion to item frequency. Table 8 presents the average results of different masking manners. We observe that different scenarios show distinctive preference for high frequency items. Specifically, web recommendation datasets tend to learn the items with high frequency in the later training phase, while urban event datasets prefer the items with low frequency. This



(a) The impact of varying α

(b) The impact of varying γ

(c) The impact of varying β

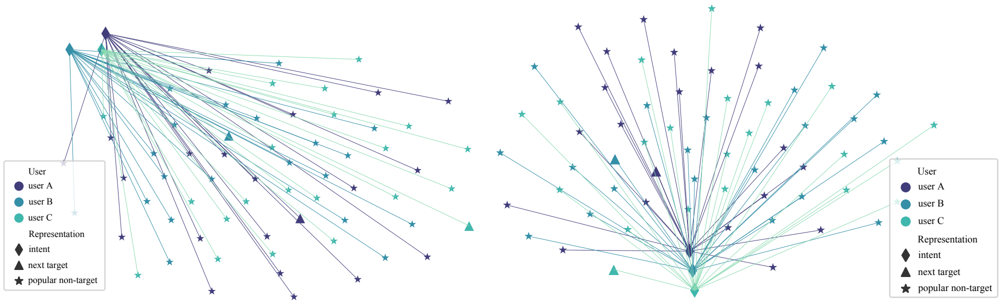
Fig. 6. Performance trend of DePoD by tuning the coefficients. All other coefficients are kept unchanged.

observation is in line with our experience where popular items in web datasets should be suppressed in personalized recommendations [25]. On the contrary, the high-frequency crime events need to be prevented with priority in urban scenarios.

5.5 Parameter Sensitivity (RQ3)

To investigate the impacts of major coefficients, we tune their values in a vanilla target item distillation, deliberate practice and non-target item distillation under the optimal settings reported in Section 5.1.3, and present the average performance on the testing datasets in Figure 6.

5.5.1 ***Coefficient α of Earlier Target Item Distillation.*** Figure 6(a) presents the impact of varying α in the range $\{1, 3, 5, 7, 9\}$. We make the following three observations: (1) As the coefficient α increases, the performance becomes better at first. This is because the target item distillation in the earlier phase mainly focuses on the high-confidence samples that can reduce the effect of noise. (2) When the α surpasses a certain threshold, the performance begins to drop against α further increasing. The reason is that overemphasizing high-confidence samples cannot make use of the diversity of training samples and may neglect some informative samples. (3) By comparing the results, three or five seems like a reasonable setting for the value of α.



(a) BERT4Rec (one encoder without decoupled progressive distillation)

(b) DePoD (decoupled progressive distillation between two BERT4Rec-based peers)

Fig. 7. Visualization of t-SNE transformed intent representations inferred from three similar user interaction sequences and their relevant item representations from the item look-up table $\boldsymbol{W}_E$ on the Beauty testing set. The color is used to distinguish the representations from a different user interaction sequence. The diamond "♦" represents the inferred next intent, the triangle "▲" represents the next target item and the star"★" represents the selected popular non-target item that the user has not interacted with.

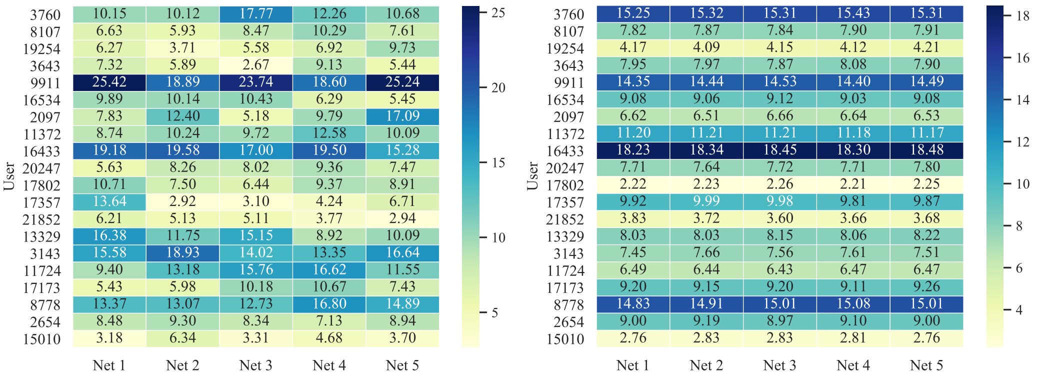
5.5.2 ***Coefficient γ of Deliberate Practice***. Figure 6(b) presents the impact of varying γ in the range $\{0.0, 0.5, 1.0, 1.5, 2.0\}$. We can observe that, with few exceptions, the performance of DePoD first rises, and then falls after a certain threshold when increasing the coefficient γ. The reason is that γ is related to the sensitivity of peer variation, thus we should tune this coefficient to find an adequate value.

5.5.3 ***Coefficient β of Non-target Item Distillation***. Figure 6(c) presents the impact of varying β in the range $\{1, 3, 5, 7, 9\}$. We have the following two observations: (1) We observe that the performance first rises before falling in most datasets, except for NYC16. This is because the non-target item distillation provides knowledge among non-target items at the beginning. However, if the value β is beyond a certain threshold, overemphasizing non-target item will suppress the contribution of the target item. For the NYC16, its threshold may be relatively low and thus show overall drop and fluctuations with β increasing. (2) Different types of datasets reflect distinct demand for knowledge among non-target items. Specifically, the coefficient β is relatively small in urban event datasets, while large in web recommendation datasets. The reason lies in the volume of candidate items, in which large sets of items can amplify the dynamics of interaction intents.

5.6 Case Study (RQ4)

5.6.1 Analysis of inferred intent representations. To investigate how the proposed decoupled progressive distillation strategy facilitates the comprehensive intent representations learning, we follow a pipeline proposed in [68] to perform the t-SNE transformation. Considering the size and clarity of the figure, we randomly select three similar user sequences paired with 21 related items (including next target item and 20 popular non-target items) that the user has not interacted with in the testing Beauty dataset. The intent representations inferred by BERT4Rec and our DePoD are plotted in Figure 7(a) and Figure 7(b), respectively. There are three observations:

- **Close to the target items from other similar sequences.** By observing the intent representations from three similar sequences, we find that compared to BERT4Rec, the intent representation inferred by our DePoD is not only close to its own target item, but also close to the target items from other similar sequences. This indicates our DePoD can facilitate the



(a) Five single BERT4Rec trained independently.

(b) DePoD with five peer encoders.

Fig. 8. Heatmap of prediction results in the Beauty test set. The number in each cell refers to the negative log-likelihood, i.e., $-\log(p_*)$, that is related to whether the user interacts with the ground-truth item.

learning of comprehensive intent representations by employing distinctive knowledge from multiple peers.

- **Close to the most likely next target item.** By comparing the representation distance between an intent (diamond) and its next target item (triangle), we find that DePoD learns a closer intent representation to the next item than BERT4Rec, especially for "user A" and "user C". This demonstrates that our target item distillation with progressive difficulty can facilitate a better intent representation corresponding to the next target item.
- **Separable representations of non-target items.** We further observe that the non-target items (star) are distributed in the lower right corner of users (diamond) in BERT4Rec and the angle between the user and two farthest non-target items is below 90 degrees. For DePoD, we find the non-target items are located on the left and right side of users, and the angle between the user and two farthest non-target items is close to 180 degrees. The above observations demonstrate that the representations of non-target items in DePoD are more separable than those in BERT4Rec. In other words, DePoD can model the interaction dynamics effectively, and the inferred comprehensive intent representation presents different distances to non-target items.

5.6.2 Visualization of inferred next target item probabilities. Our work mainly employs the response difference between multi-peer prediction networks to model the dynamics of interaction intents. As such, to investigate how our proposed DePoD exploits the difference between networks, the negative log-likelihood of inferred next target item probabilities from 20 randomly selected users are shown in Figure 8. Comparing Figure 8(a) and Figure 8(b), we have the following observations:

- **Consistent color and close number in Figure 8(b).** The color and number in each cell of Figure 8(b) are more consistent than those of Figure 8(a). This indicates that our DePoD effectively transfers knowledge between different networks and reduce their response difference, thus modeling the dynamics of interactions intents.
- **Relatively small value in the colorbar of Figure 8(b).** We find the maximum value in the colorbar of Figure 8(a) is 25, while that of Figure 8(b) is 18. This demonstrates that our DePoD does not overemphasize the intent corresponding to target item whilst considering the other intents covered in volume non-target items.

6 RELATED WORK

In this section, we briefly review the related works and present their main differences to our proposed DePoD. In particular, we focus on three areas: (i) sequential prediction, (ii) knowledge distillation and (iii) progressive learning.

6.1 Sequential Prediction

Sequential prediction methods have been applied in many domains, such as sequential recommendation [10, 55, 64, 66, 79], click-through rate estimation [46, 71] and urban computing [22, 24, 78]. For handling of interaction dynamics, they mainly focus on the optimization of intent representations and can be grouped into two categories: vector-based methods, and distribution-based methods.

Vector-based methods are in the spotlight of the research community, and can be further divided into three subcategories. Firstly, most methods commonly introduce extra side information to cope with the lack of discriminative information, e.g., user profiling [15], item attributes [79], cross-domain knowledge [69, 77], geographical information [39, 58] and environmental situations [27]. As the side information is not always available in the setting of sequential prediction tasks, some methods try to maximize the exploitation of observed historical sequences such as long-term information [8, 78], or temporal context [55, 73]. Secondly, without introducing side information, denoising-based methods [14, 66] focus on the exploration of high-quality samples and take the samples that cannot learn well as noise. Despite their effectiveness on small scenarios containing abundant training samples, these methods simply that regard interaction dynamics as noise will inevitablly to waste precious data resources. Thirdly, a line of works [4, 22, 24, 31, 64] are proposed to model interaction dynamics in sequential prediction, which is one of the best practices for making use of limited training data. DuroNet [22] devises a noise-robust structure to reduce the effect of local outliers in crime counts. HyperRec [64] tries to construct multiple sequential hyper-graphs to capture user preference and utilize attention to obtain aggregative representations. MTD [26] integrates intra- and inter-session transition dynamics by developing a position-aware attentive mechanism. More recent works [4, 24, 31] make multiple prediction models to learn from each other and employ the output probability distribution from the other to capture dynamic intents. SoftRec [4] focuses on well-designed soft targets to model the ambiguity of unobserved feedback. BiCAT [31] generates pseudo-prior items to address the cold-start problem. HAIL [24] collects the mutual exclusivity knowledge to mine implicitly hard interactions.

Along with the modeling of interaction dynamics, distribution-based methods [9, 10, 32] present a similar motivation to multiple vector-based methods [24, 64]. DT4SR [9] and GeRec [32] adopt Gaussian embedding to expand the representation of users and items. And, STOSA [10] further introduces uncertainty into pattern learning and develops stochastic attention via Wasserstein distance[4]. Despite their success and impressive theoretical explanation, the limitation mainly lies in the prior distribution, since the analytical solutions of most distributions are still unsolved in mathematics. More importantly, compared with multiple vector-based methods, some empirical studies [12, 18, 53] show prior distributions may impose restrictions on search spaces and lead to a sub-optimal prediction model.

Most of the above work optimizes the sequential pattern representations through network structures or representation manners. Based on their works, our DePoD further focuses on the effect of response difference on model training. In particular, we employ knowledge distillation to explicitly model the interaction dynamics. To enhance distillation among volume items, we devise a decoupled progressive distillation strategy to schedule the learning of different items.

[4] A distance function between probability distributions motivated by the idea of optimal transport. At present, we can only find its analytical solution in a few distributions, such as one-dimensional distributions and Gaussian distributions.

6.2 Knowledge Distillation

Knowledge distillation (KD) can be traced from model compression [21], where an efficient student model tries to mimic the responses from a cumbersome teacher model. Recently, the distillation scheme has been evolving from offline [21] to online [74, 76] training where multiple models reciprocally learn from each other and thus improve themselves together. In the scheme of online distillation, our work is mainly related to the logit knowledge, which presents straightforward and high-level semantic information in the output probability distributions, yet still needs to be fully exploited. Furthermore, on the basis of distillation between two sequential networks, we also investigate response difference among three or more networks.

Most logit distillation works mainly impose regularization and strictness [65] on softened labels and have achieved great success in CV and NLP. Cho et al. [5] point out that the logits from early stopped teachers can reduce the mismatch capacity between teachers and students. Mirzadeh et al. [45] further introduce an intermediate-sized network as teacher assistant to fill in the capacity gap. Inspired by the practice in CV and NLP, several works [4, 31, 70, 82] also apply KD in sequential prediction. Zhu et al. [82] attempt different KD schemes and propose an ensemble CTR estimation method. Xia et al. [70] focus on device recommendation systems and develop a self-supervised KD framework to obtain a compressed model. In contrast, some works [4, 24, 31] aim to obtain better prediction performance via distillation. BiCAT [31] and SoftRec [4] take advantage of augmented sequences and well-designed soft targets through self-distillation, respectively. HAIL [24] proposes mutual exclusivity distillation to acquire hints from the unlikelihood of teachers' correct responses. Furthermore, to make use of the heterogeneity knowledge from multiple teachers, a line of works focus on the ensemble of logits, including averaged or attentive aggregation [3, 74, 82], iterative learning [13, 54] and adaptive knowledge amalgamation [43].

The above works distill knowledge in a unified way, which does not consider the different effects of target and non-target item parts. More recently, DKD [75] initially reveals the suppression of non-target term by target term and provides a flexible distillation formulation, offering more unexplored spaces to improve target and non-target learning. Our work differs from them by devising a decoupled progressive distillation strategy to adapt from volume candidate items.

6.3 Progressive Learning

Progressive learning can be tracked in human cognitive progress where people gradually learn knowledge as they grow up. Various practices occur within this concept, such as curriculum learning [2, 40, 67], deliberate practice [1, 17] and continual learning [11]. Our work is mainly related to curriculum learning and deliberate practice in which they both follow a predefined easy-to-hard scheme. For the widely used curriculum learning, most existing works [34, 72, 80] focus on the difficulty of training samples. For example, Jin et al. [34] employs the optimization trajectory of a teacher model to construct an easy-to-hard sequence of learning target. Zhou et al. [80] utilize cross-entropy and variance to present the difficulty and uncertainty of training data. Zhang et al. [72] train multiple teacher models and distinguish easy and hard instances via a crossed manner. In this paper, we combine the knowledge distillation and curriculum learning to extend the current level. Besides the sample difficulty in target item distillation, the difficulty of sub-tasks and teachers are also defined in non-target item distillation and progressive peer selection.

7 CONCLUSION AND FUTURE WORK

In this work, we highlight the dynamics of interaction intents in sequential prediction, especially for those with volume candidate items. To this end, we propose a sequential prediction framework with decoupled progressive distillation (DePoD) that is inspired by the progressive nature of human cognition to enhance distillation among multi-peer prediction networks, thus modeling the interaction dynamics. Basically, multi-peer prediction networks present distinctive model responses in the impact of interaction dynamics. To enhance distillation among volume candidate items, we reveal the effects of target and non-target items according our theoretical analysis, and further develop a decoupled progressive distillation strategy. In particular, we address two issues: (1) (How to learn?) The target item distillation with progressive difficulty and non-target item distillation, with progressive size starting from an easy and small point and gradually developing toward a hard training scheme, which encourages comprehensive intent representations; (2) (Whom to learn from?) The progressive peer selection employs a trainable difference evaluator to gradually select an expert with adequate response difference, which further enhances the target and non-target distillation. Extensive experiments on four public datasets show that DePoD achieves superiority over state-of-the-art methods on a set of accuracy-based metrics.

Notwithstanding the impressive problem and promising performance, our DePoD framework still has some limitations. On the one hand, our DePoD still needs to be extended to more complex sequential prediction tasks. These tasks may cover more side information or prediction demands, e.g., multi-modal sequential prediction and sequence-to-sequence prediction. On the other hand, DePoD follows the online distillation scheme, which incurs relatively large computation resources in the training phase. In the future, other distillation schemes can be investigated to alleviate this problem, such as self-distillation.

REFERENCES

[1] K Anders Ericsson. 2008. Deliberate practice and acquisition of expert performance: a general overview. *Academic emergency medicine* 15, 11 (2008), 988–994.

[2] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, Vol. 382. ACM, Montreal, Quebec, Canada, 41–48.

[3] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen. 2020. Online Knowledge Distillation with Diverse Peers. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. AAAI Press, New York, NY, USA, 3430–3437.

[4] Mingyue Cheng, Fajie Yuan, Qi Liu, Shenyang Ge, Zhi Li, Runlong Yu, Defu Lian, Senchao Yuan, and Enhong Chen. 2021. Learning Recommender Systems with Implicit Feedback via Soft Target Enhancement. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Virtual Event, Canada, 575–584.

[5] Jang Hyun Cho and Bharath Hariharan. 2019. On the Efficacy of Knowledge Distillation. In *2019 IEEE/CVF International Conference on Computer Vision*. IEEE, Seoul, Korea (South), 4793–4801.

[6] Didan Deng, Liang Wu, and Bertram E. Shi. 2021. Iterative Distillation for Better Uncertainty Estimates in Multitask Emotion Recognition. In *Proceedings of the 18th IEEE International Conference on Computer Vision*. IEEE, Montreal, BC, Canada, 3550–3559.

[7] Armen Der Kiureghian and Ove Ditlevsen. 2009. Aleatory or epistemic? Does it matter? *Structural safety* 31, 2 (2009), 105–112.

[8] Jing Du, Zesheng Ye, Bin Guo, Zhiwen Yu, and Lina Yao. 2023. IDNP: Interest Dynamics Modeling Using Generative Neural Processes for Sequential Recommendation. In *Proceedings of the 16th ACM International Conference on Web Search and Data Mining*. ACM, Singapore, 481–489.

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

[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 *Proceedings of the 31st Web Conference*. ACM, Virtual Event, Lyon, France, 2036–2047.

[11] Haytham M. Fayek, Lawrence Cavedon, and Hong Ren Wu. 2020. Progressive learning: A deep learning framework for continual learning. *Neural Networks* 128 (2020), 345–357.

[12] Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. 2019. Deep Ensembles: A Loss Landscape Perspective. *CoRR* abs/1912.02757 (2019). http://arxiv.org/abs/1912.02757

[13] Tommaso Furlanello, Zachary Chase Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. 2018. Born-Again Neural Networks. In *Proceedings of the 35th International Conference on Machine Learning*, Vol. 80. PMLR,

Stockholm, Sweden, 1602–1611.

[14] Yunjun Gao, Yuntao Du, Yujia Hu, Lu Chen, Xinjun Zhu, Ziquan Fang, and Baihua Zheng. 2022. Self-Guided Learning to Denoise for Robust Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Madrid, Spain, 1412–1422.

[15] Yulong Gu, Zhuoye Ding, Shuaiqiang Wang, and Dawei Yin. 2020. Hierarchical User Profiling for E-commerce Recommender Systems. In *Proceedings of the 13th ACM International Conference on Web Search and Data Mining*. ACM, Houston, TX, USA, 223–231.

[16] Maya R. Gupta, Samy Bengio, and Jason Weston. 2014. Training highly multiclass classifiers. *J. Mach. Learn. Res.* 15, 1 (2014), 1461–1492.

[17] David Z Hambrick, Frederick L Oswald, Erik M Altmann, Elizabeth J Meinz, Fernand Gobet, and Guillermo Campitelli. 2014. Deliberate practice: Is that all it takes to become an expert? *Intelligence* 45 (2014), 34–45.

[18] Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. 2020. Bayesian Deep Ensembles via the Neural Tangent Kernel. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems*. Vancouver, Canada.

[19] Ruining He and Julian J. McAuley. 2016. Fusing Similarity Models with Markov Chains for Sparse Sequential Recommendation. In *Proceedings of the 16th IEEE International Conference on Data Mining*. IEEE, Barcelona, Spain, 191–200.

[20] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *Proceedings of the 4th International Conference on Learning Representations*. OpenReview.net, San Juan, Puerto Rico.

[21] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2014. Distilling the Knowledge in a Neural Network. In *Proceedings of the the 28th Conference on Neural Information Processing Systems*. Montreal, Quebec, Canada.

[22] Kaixi Hu, Lin Li, Jianquan Liu, and Daniel Sun. 2021. DuroNet: A Dual-robust Enhanced Spatial-temporal Learning Network for Urban Crime Prediction. *ACM Trans. Internet Techn.* 21, 1 (2021), 24:1–24:24.

[23] Kaixi Hu, Lin Li, Xiaohui Tao, Juan D. Velásquez, and Patrick J. Delaney. 2023. Information fusion in crime event analysis: A decade survey on data, features and models. *Inf. Fusion* 100 (2023), 101904–101918.

[24] Kaixi Hu, Lin Li, Qing Xie, Jianquan Liu, and Xiaohui Tao. 2021. What is Next when Sequential Prediction Meets Implicitly Hard Interaction?. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management*. ACM, Virtual Event, Queensland, Australia, 710–719.

[25] Liang Hu, Longbing Cao, Shoujin Wang, Guandong Xu, Jian Cao, and Zhiping Gu. 2017. Diversifying Personalized Recommendation with User-session Context. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. ijcai.org, Melbourne, Australia, 1858–1864.

[26] Chao Huang, Jiahui Chen, Lianghao Xia, Yong Xu, Peng Dai, Yanqing Chen, Liefeng Bo, Jiashu Zhao, and Jimmy Xiangji Huang. 2021. Graph-Enhanced Multi-Task Learning of Multi-Level Transition Dynamics for Session-based Recommendation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*. AAAI Press, Virtual Event, 4123–4130.

[27] Chao Huang, Chuxu Zhang, Jiashu Zhao, Xian Wu, Nitesh V. Chawla, and Dawei Yin. 2019. MiST: A Multiview and Multimodal Spatial-Temporal Learning Framework for Citywide Abnormal Event Forecasting. In *Proceedings of the 29th Web Conference*. ACM, San Francisco, CA, USA, 717–728.

[28] Ji Huang, Minbo Ma, Yongsheng Dai, Jie Hu, and Shengdong Du. 2023. DBAFormer: A Double-Branch Attention Transformer for Long-Term Time Series Forecasting. *Hum. Centric Intell. Syst.* 3, 3 (2023), 263–274.

[29] Eyke Hüllermeier and Willem Waegeman. 2021. Aleatoric and epistemic uncertainty in machine learning: an introduction to concepts and methods. *Mach. Learn.* 110, 3 (2021), 457–506.

[30] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *Proceedings of the 5th International Conference on Learning Representations*. OpenReview.net, Toulon, France.

[31] Juyong Jiang, Yingtao Luo, Jae Boum Kim, Kai Zhang, and Sunghun Kim. 2021. Sequential Recommendation with Bidirectional Chronological Augmentation of Transformer. *CoRR* abs/2112.06460 (2021). https://arxiv.org/abs/2112.06460

[32] Junyang Jiang, Deqing Yang, Yanghua Xiao, and Chenlu Shen. 2019. Convolutional Gaussian Embeddings for Personalized Recommendation with Uncertainty. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. ijcai.org, Macao, China, 2642–2648.

[33] Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for Natural Language Understanding. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. ACL, Virtual Event, 4163–4174.

[34] Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. 2019. Knowledge Distillation via Route Constrained Optimization. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision*. IEEE, Seoul, Korea (South), 1345–1354.

[35] SeongKu Kang, Junyoung Hwang, Wonbin Kweon, and Hwanjo Yu. 2020. DE-RRD: A Knowledge Distillation Framework for Recommender System. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. ACM, Virtual Event, Ireland, 605–614.

[36] Bartosz Krawczyk, Mikel Galar, Michal Wozniak, Humberto Bustince, and Francisco Herrera. 2018. Dynamic ensemble selection for multi-class classification with one-class classifiers. *Pattern Recognit.* 83 (2018), 34–51.

[37] Kai A Krueger and Peter Dayan. 2009. Flexible shaping: How learning in small steps helps. *Cognition* 110, 3 (2009), 380–394.

[38] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Proceedings of the the 31st Conference on Neural Information Processing Systems*. Long Beach, CA, USA, 6402–6413.

[39] Defu Lian, Yongji Wu, Yong Ge, Xing Xie, and Enhong Chen. 2020. Geography-Aware Sequential Location Recommendation. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, Virtual Event, CA, USA, 2009–2019.

[40] Jingren Liu, Yi Chen, Huajun Liu, Haofeng Zhang, and Yudong Zhang. 2022. From Less to More: Progressive Generalized Zero-Shot Detection With Curriculum Learning. *IEEE Trans. Intell. Transp. Syst.* (2022), 1–14.

[41] Kangzheng Liu, Feng Zhao, Hongxu Chen, Yicong Li, Guandong Xu, and Hai Jin. 2022. DA-Net: Distributed Attention Network for Temporal Knowledge Graph Reasoning. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. ACM, Atlanta, GA, USA, 1289–1298.

[42] Kangzheng Liu, Feng Zhao, Guandong Xu, Xianzhi Wang, and Hai Jin. 2022. Temporal Knowledge Graph Reasoning via Time-Distributed Representation Learning. In *Proceedings of the IEEE International Conference on Data Mining*. IEEE, Orlando, FL, USA, 279–288.

[43] Sihui Luo, Wenwen Pan, Xinchao Wang, Dazhou Wang, Haihong Tang, and Mingli Song. 2020. Collaboration by Competition: Self-coordinated Knowledge Amalgamation for Multi-talent Student Learning. In *Proceedings of the 16th European Conference on Computer Vision*, Vol. 12351. Springer, Glasgow, UK, 631–646.

[44] Yingtao Luo, Qiang Liu, and Zhaocheng Liu. 2021. STAN: Spatio-Temporal Attention Network for Next Location Recommendation. In *Proceedings of the 30th Web Conference*. ACM, Virtual Event / Ljubljana, Slovenia, 2177–2185.

[45] Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. 2020. Improved Knowledge Distillation via Teacher Assistant. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. AAAI Press, New York, NY, USA, 5191–5198.

[46] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Practice on Long Sequential User Behavior Modeling for Click-Through Rate Prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, Anchorage, AK, USA, 2671–2679.

[47] Sayan Putatunda and Arnab Kumar Laha. 2023. Travel Time Prediction in Real time for GPS Taxi Data Streams and its Applications to Travel Safety. *Hum. Centric Intell. Syst.* 3, 3 (2023), 381–401.

[48] Biao Qian, Yang Wang, Hongzhi Yin, Richang Hong, and Meng Wang. 2022. Switchable Online Knowledge Distillation. In *Proceedings of the 17th European Conference*, Vol. 13671. Springer, Tel Aviv, Israel, 449–466.

[49] Steven Reich, David Mueller, and Nicholas Andrews. 2020. Ensemble Distillation for Structured Prediction: Calibrated, Accurate, Fast - Choose Three. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. ACL, Virtual Event, 5583–5595.

[50] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*. ACM, Raleigh, North Carolina, USA, 811–820.

[51] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. 2015. FitNets: Hints for Thin Deep Nets. In *Proceedings of the 3rd International Conference on Learning Representations*. OpenReview.net, San Diego, CA, USA.

[52] Hui Shi, Yupeng Gu, Yitong Zhou, Bo Zhao, Sicun Gao, and Jishen Zhao. 2023. Everyone's Preference Changes Differently: A Weighted Multi-Interest Model For Retrieval. In *Proceedings of the 40th International Conference on Machine Learning*. PMLR, Honolulu, Hawaii, USA, 31228–31242.

[53] Jasper Snoek, Yaniv Ovadia, Emily Fertig, Balaji Lakshminarayanan, Sebastian Nowozin, D. Sculley, Joshua V. Dillon, Jie Ren, and Zachary Nado. 2019. Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift. In *Proceedings of the 33rd Annual Conference on Neural Information Processing Systems*. Vancouver, Canada, 13969–13980.

[54] Wonchul Son, Jaemin Na, Junyong Choi, and Wonjun Hwang. 2021. Densely Guided Knowledge Distillation using Multiple Teacher Assistants. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision*. IEEE, Montreal, QC, Canada, 9375–9384.

[55] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *Proceedings of the 28th ACM International*

Conference on Information and Knowledge Management. ACM, Beijing, China, 1441–1450.

[56] Qiaoyu Tan, Jianwei Zhang, Ninghao Liu, Xiao Huang, Hongxia Yang, Jingren Zhou, and Xia Hu. 2021. Dynamic Memory based Attention Network for Sequential Recommendation. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*. AAAI Press, Virtual Event, 4384–4392.

[57] Jiaxi Tang and Ke Wang. 2018. Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding. In *Proceedings of the 11th ACM International Conference on Web Search and Data Mining*. ACM, Marina Del Rey, CA, USA, 565–573.

[58] Wanjie Tao, Yu Li, Liangyue Li, Zulong Chen, Hong Wen, Peilin Chen, Tingting Liang, and Quan Lu. 2022. SMINet: State-Aware Multi-Aspect Interests Representation Network for Cold-Start Users Recommendation. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence*. AAAI Press, Virtual Event, 8476–8484.

[59] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. 2019. An Empirical Study of Example Forgetting during Deep Neural Network Learning. In *Proceedings of the 7th International Conference on Learning Representations*. OpenReview.net, New Orleans, LA, USA.

[60] Willem Robert van Hage, Véronique Malaisé, Roxane Segers, Laura Hollink, and Guus Schreiber. 2011. Design and use of the Simple Event Model (SEM). *J. Web Semant.* 9, 2 (2011), 128–136.

[61] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the the 31st Conference on Neural Information Processing Systems*. Long Beach, CA, USA, 5998–6008.

[62] Chenyang Wang, Weizhi Ma, Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2023. Sequential Recommendation with Multiple Contrast Signals. *ACM Trans. Inf. Syst.* 41, 1 (2023), 11:1–11:27.

[63] Chenyang Wang, Weizhi Ma, Min Zhang, Chong Chen, Yiqun Liu, and Shaoping Ma. 2021. Toward Dynamic User Intention: Temporal Evolutionary Effects of Item Relations in Sequential Recommendation. *ACM Trans. Inf. Syst. (TOIS)* 39, 2 (2021), 16:1–16:33.

[64] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item Recommendation with Sequential Hypergraphs. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. ACM, Virtual Event, China, 1101–1110.

[65] Lin Wang and Kuk-Jin Yoon. 2022. Knowledge Distillation and Student-Teacher Learning for Visual Intelligence: A Review and New Outlooks. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 6 (2022), 3048–3068.

[66] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising Implicit Feedback for Recommendation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. ACM, Virtual Event, Israel, 373–381.

[67] Xin Wang, Yudong Chen, and Wenwu Zhu. 2022. A Survey on Curriculum Learning. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 9 (2022), 4555–4576.

[68] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural Graph Collaborative Filtering. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Paris, France, 165–174.

[69] Junda Wu, Zhihui Xie, Tong Yu, Handong Zhao, Ruiyi Zhang, and Shuai Li. 2022. Dynamics-Aware Adaptation for Reinforcement Learning Based Cross-Domain Interactive Recommendation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Madrid, Spain, 290–300.

[70] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Guandong Xu, and Quoc Viet Hung Nguyen. 2022. On-Device Next-Item Recommendation with Self-Supervised Knowledge Distillation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. ACM, Madrid, Spain, 546–555.

[71] Weinan Xu, Hengxu He, Minshi Tan, Yunming Li, Jun Lang, and Dongbai Guo. 2020. Deep Interest with Hierarchical Attention Network for Click-Through Rate Prediction. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. ACM, Virtual Event, China, 1905–1908.

[72] Licheng Zhang, Zhendong Mao, Benfeng Xu, Quan Wang, and Yongdong Zhang. 2021. Review and Arrange: Curriculum Learning for Natural Language Understanding. *IEEE ACM Trans. Audio Speech Lang. Process.* 29 (2021), 3307–3320.

[73] Qi Zhang, Longbing Cao, Chongyang Shi, and Zhendong Niu. 2022. Neural Time-Aware Sequential Recommendation by Jointly Modeling Preference Dynamics and Explicit Feature Couplings. *IEEE Trans. Neural Networks Learn. Syst.* 33, 10 (2022), 5125–5137.

[74] Ying Zhang, Tao Xiang, Timothy M. Hospedales, and Huchuan Lu. 2018. Deep Mutual Learning. In *Proceedings of the 2018 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, UT, USA, 4320–4328.

[75] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. 2022. Decoupled Knowledge Distillation. In *Proceedings of the 35th IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, New Orleans, Louisiana, 11953–11962.

[76] Haojie Zhao, Gang Yang, Dong Wang, and Huchuan Lu. 2021. Deep mutual learning for visual object tracking. *Pattern Recognit.* 112 (2021), 107796–107808.

[77] Yuyue Zhao, Xiang Wang, Jiawei Chen, Yashen Wang, Wei Tang, Xiangnan He, and Haiyong Xie. 2023. Time-aware Path Reasoning on Knowledge Graph for Recommendation. *ACM Trans. Inf. Syst.* 41, 2 (2023), 26:1–26:26.

[78] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*. AAAI Press, Virtual Event, 11106–11115.

[79] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. ACM, Virtual Event, Ireland, 1893–1902.

[80] Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-Aware Curriculum Learning for Neural Machine Translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. ACL, Virtual Event, 6934–6944.

[81] Bing Zhu, Cheng Qian, Seppe vanden Broucke, Jin Xiao, and Yuanyuan Li. 2023. A bagging-based selective ensemble model for churn prediction on imbalanced data. *Expert Syst. Appl.* 227 (2023), 120223 – 120233.

[82] Jieming Zhu, Jinyang Liu, Weiqi Li, Jincai Lai, Xiuqiang He, Liang Chen, and Zibin Zheng. 2020. Ensembled CTR Prediction via Knowledge Distillation. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management*. ACM, Virtual Event, Ireland, 2941–2958.

[83] Yichen Zhu and Yi Wang. 2021. Student Customized Knowledge Distillation: Bridging the Gap Between Student and Teacher. In *Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision*. IEEE, Montreal, QC, Canada, 5037–5046.